高密度実装(LSI)
FPGA活動報告 2024年度

テーマ：新たな技術を習得する

目標：実基板上で動作検証を行う

2024年4月～9月　開発部

・FPGAとは？

プログラムを書き込んで処理内容が変更できる部品です。

メリット

①部品を実装した基板でも処理内容の変更が可能です。
→　基板の再設計が必要ない。

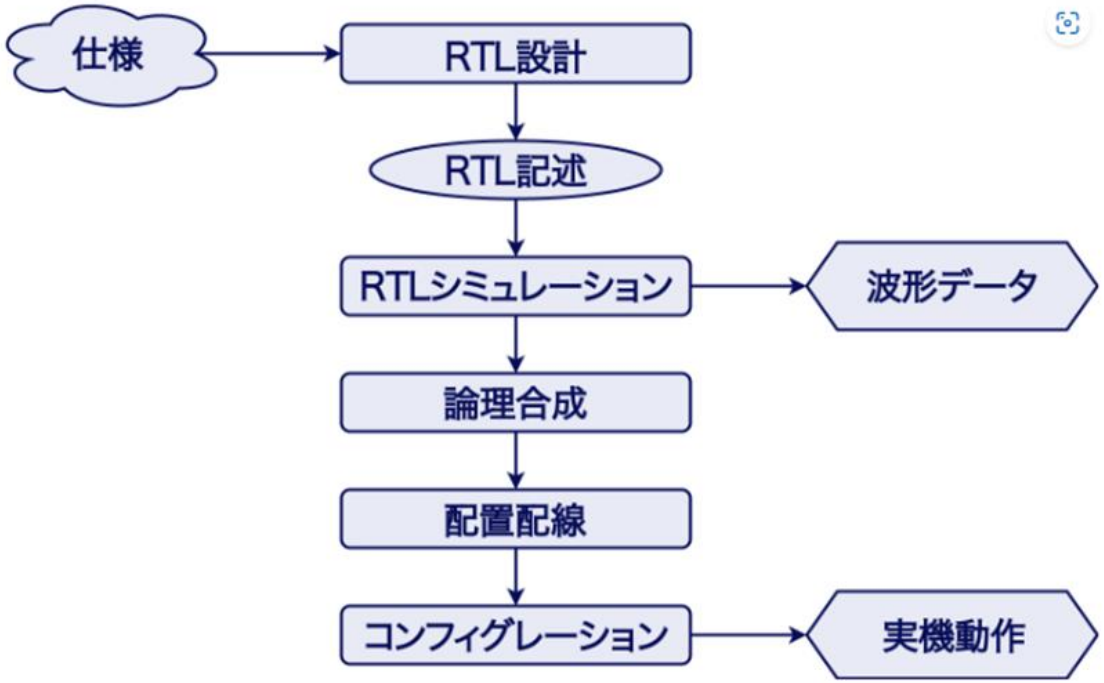
②実装面積でも省スペース化が可能です。
→　基板の小型化も実現できます。

その他　消電力化、部品代の削減　（回路規模による）

・FPGAの開発手順は？

開発手順

FPGAで回路を作るには、以下の手順を踏みます。



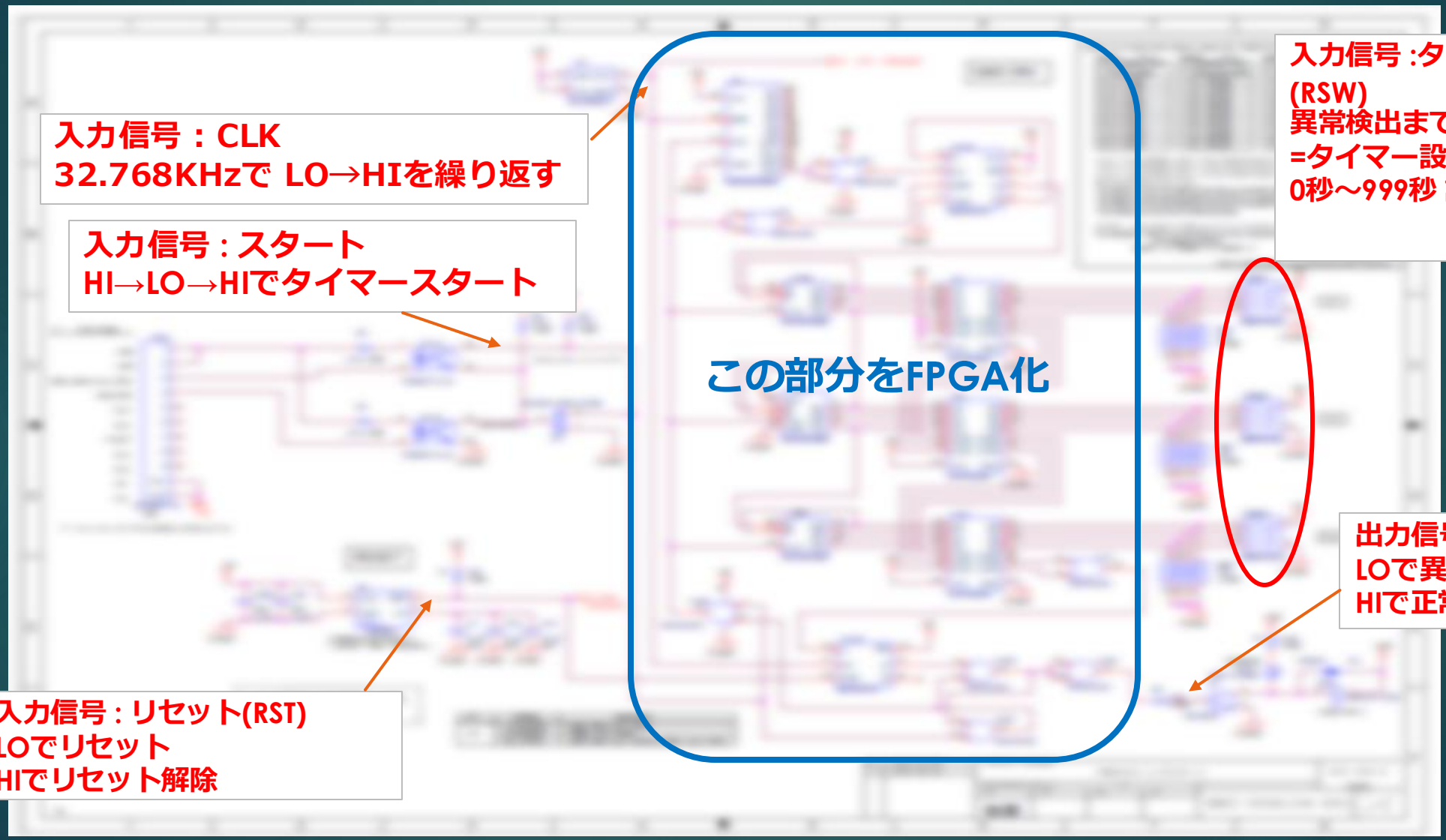
まず仕様を確認し、設計を考えます。 その後は、以下の手順に沿って開発を進めます。

開発手順	説明
RTL設計	仕様を確認し、どのようなハードウェアにするか設計を考えます。
RTL記述	設計を元に、ソースファイル(Verilogで記述された回路記述)を作成します。
RTLシミュレーション (Simulation)	RTL設計で作成した回路記述によって仕様通りの結果が得られるか、PC上でシミュレーションをします。
論理合成 (Synthesys)	RTLを元に、論理回路・ネットリストを生成します。
配置配線 (Implementation)	論理回路を元に、実際のFPGA上への配置・配線方法を決定します。この時に、FPGAチップ外部のスイッチやボタン・LEDに対する配線も決めます。レイアウトとも言います。
コンフィグレーション (プログラム)	配置配線データを元にFPGAを変更する「コンフィグレーションデータ」を生成し、FPGAに転送することで、実際の回路をFPGA上に構築します。

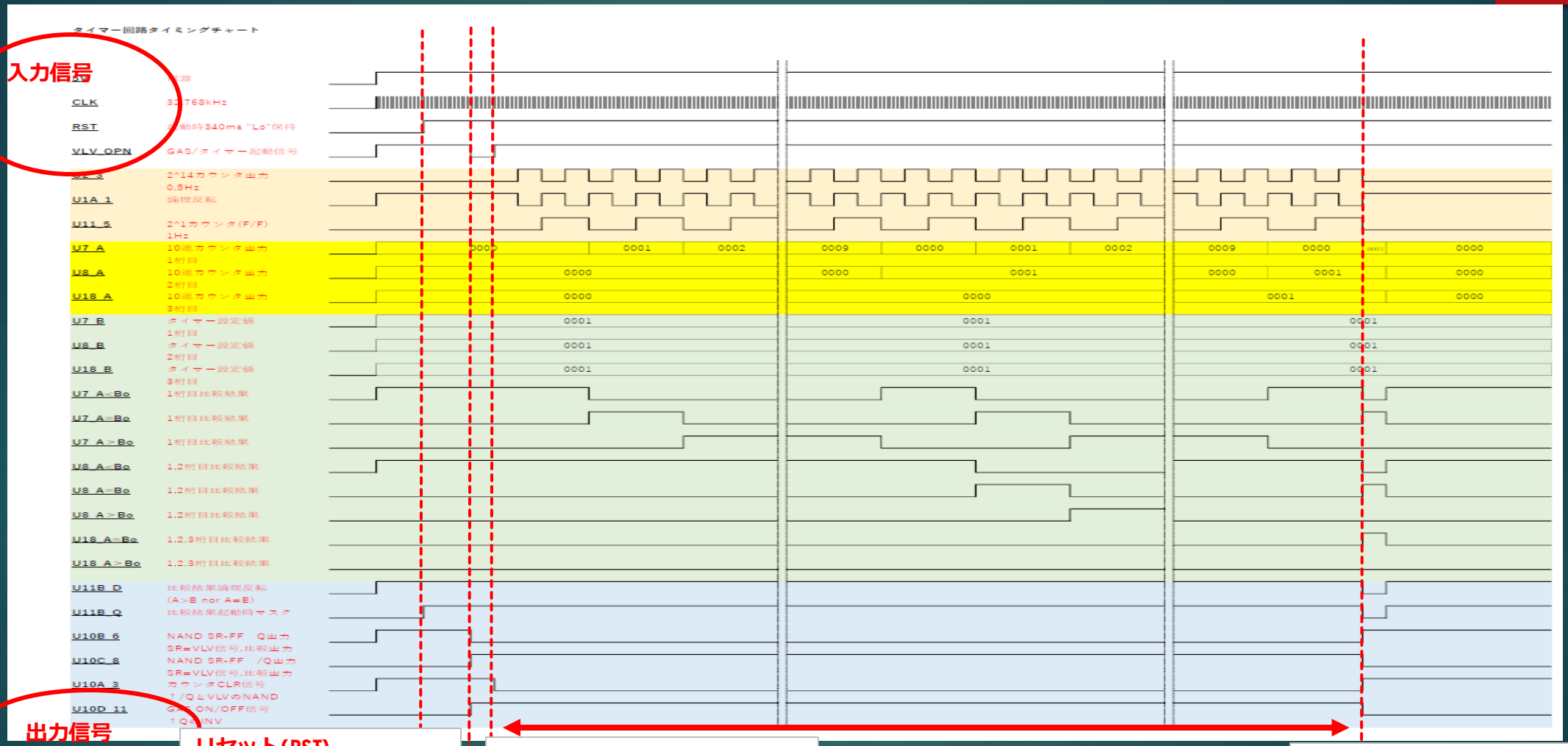
また、シミュレーションによって波形データが出力でき、コンフィグレーションを行うことでFPGAの実機動作が行えます。これらを用いてデバックを行い、仕様通りの回路を設計していくことになります。

・今回取り組んだ回路、仕様

部品数:9個



・仕様、タイムチャート


タイマー回路タイミングチャート
入力信号
CLK
RST
VLV_OPN
U7_A
U8_A
U18_A
U7_B
U8_B
U18_B
U7_A=Bo
U7_A=Bo
U7_A>Bo
U8_A=Bo
U8_A=Bo
U8_A>Bo
U18_A=Bo
U18_A>Bo
U11B_D
U11B_Q
U10B_6
U10C_8
U10A_3
U10D_11
出力信号
リセット(RST)
リセット中 → 解除
スタート LO→HI
→ タイマースタート
タイマー終了
→ 出力 HI→LO
スタート HI→LO
→ 出力 LO→HI

・設計環境、設計仕様

使用FPGA： xc7a35ticsg324-1l(xilinx)

設計言語：　Verilog HDL (ベリログ)

設計ソフト：Vivado (ビィバド) (AMD)

評価ボード：シミュレーションボード　ARTY A7(DIGILENT)、実機確認用基板

入出力仕様：

入力信号：クロック(シミュレーションボード：100MHz,実機基板：32.768KHz)、リセット、スタート、タイマー設定
出力信号：結果出力

タイマー設定時間がSWで変更可能　シミュレーションボード 4bit：0～15秒、 実機確認用基板 12Bit：0～999秒

実機基板で動作確認を行う際は、シミュレーションボードの外部ポートを使用して動作確認を行う。

ただし、シミュレーション基板と実機基板の信号のレベルが違うので変換が必要となる。

変換IC(74LVC8T245)を使用してレベル変換を行う。

シミュレーションボード側(FPGA)：+3.3V　↔　実機基板側：+5.0V

・環境の説明

設計言語：
　FPGAを設計する為には ソースファイル（プログラム）を作成する必要があります。
　使用した言語はVerilog HDLという言語で、今回初めて使用しました。

壁①：今まで何種類かの言語をコーディングしてきたが、すごく癖のある書き方、考え方でした。

設計ソフト：
　使用するシミュレーションボード搭載のFPGAがXILINX(AMD)の為、
　ソフトも同社のVivadoで開発する必要がありました。
　　FPGAのメーカーが変われば設計ソフトも、そのメーカーの物が必要となります。

壁②：ソフト自体が英語仕様なので全て英語表記で理解するのが大変でした。特にエラー表示など。

シミュレーションボード:
　FPGA及び周辺回路、外部ポート等が全て搭載されているボード。FPGA学習用の市販品。

設計方法や操作方法は、全てインターネットで調べて少しずつ理解していきます。

・動作検証１：シミュレーション

プログラムを作成したので動作検証を行ってみた。

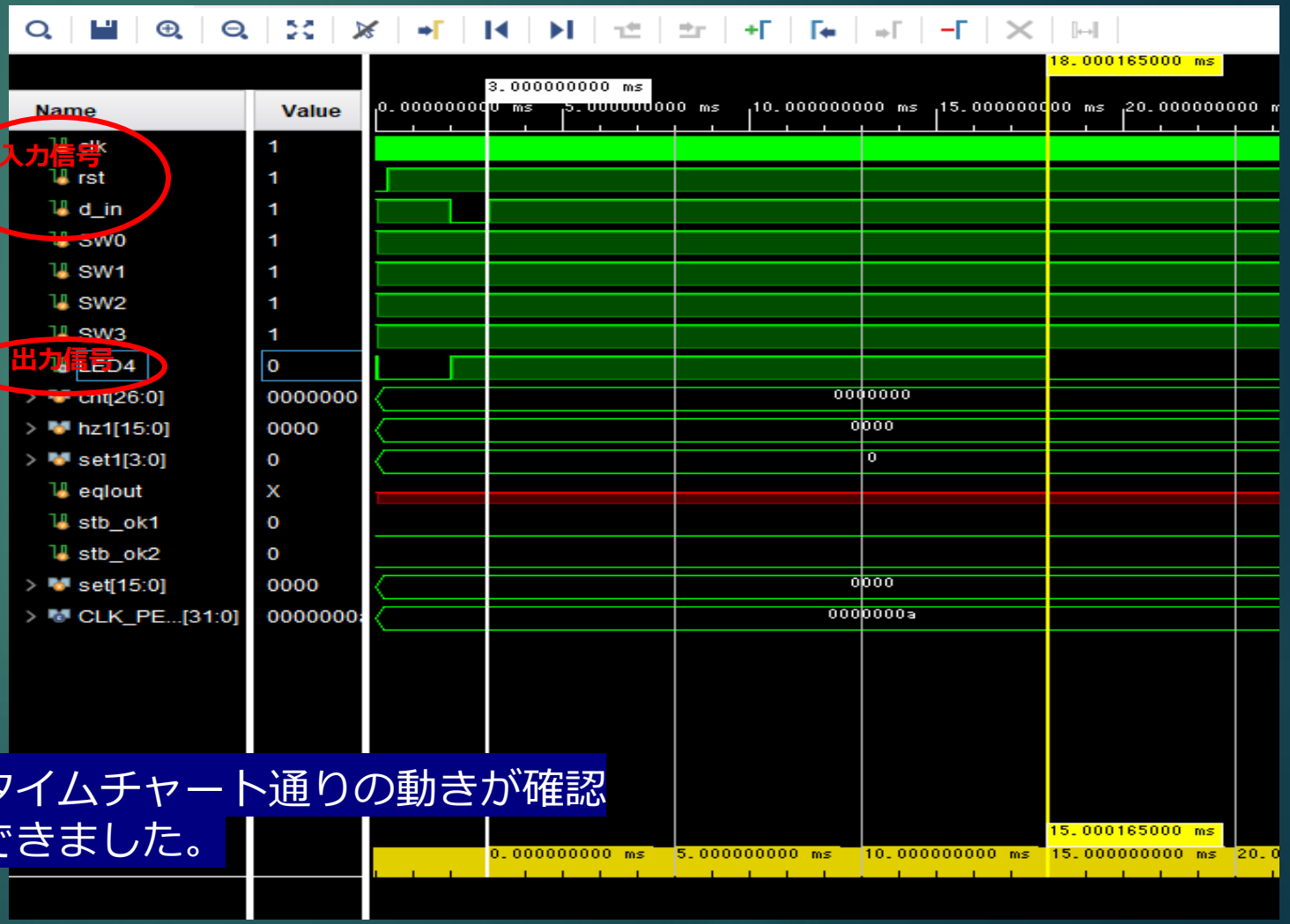
Vivadoには検証する為のシミュレータがありこれを使ってみました。

まず、シミュレーションを行うには
テストベンチが必要となります。

テストベンチとは、シミュレーションを行う際の入力信号を時系列で与える為のプログラムです。

色々試してシミュレーションが完成。

sw0～3 タイマー設定時間 (4bit)　設定時間 15s

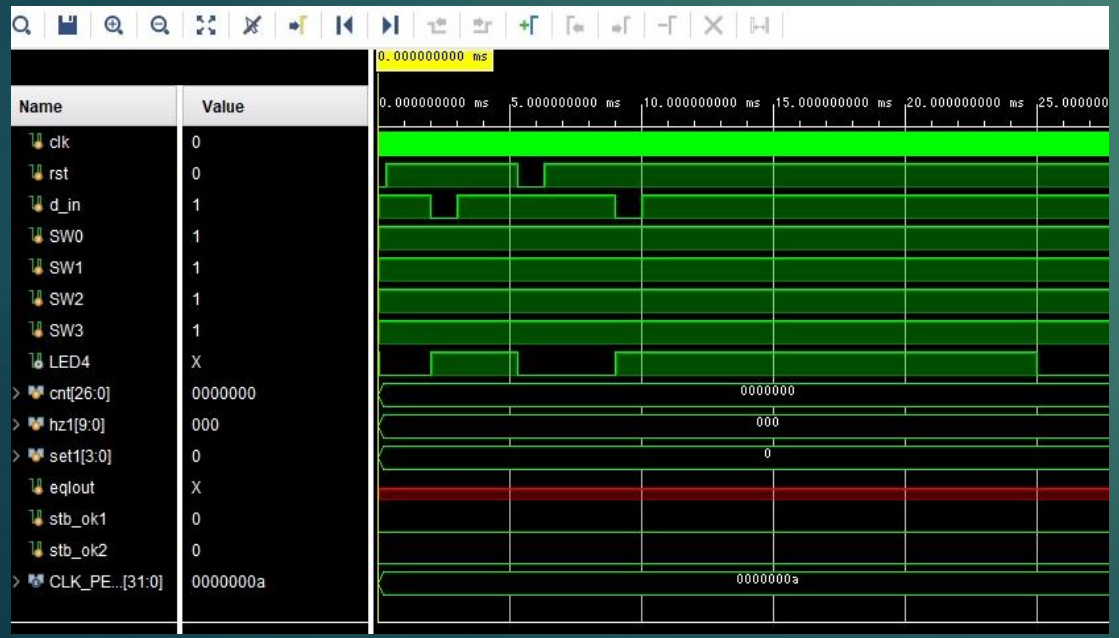


タイムチャート通りの動きが確認できました。

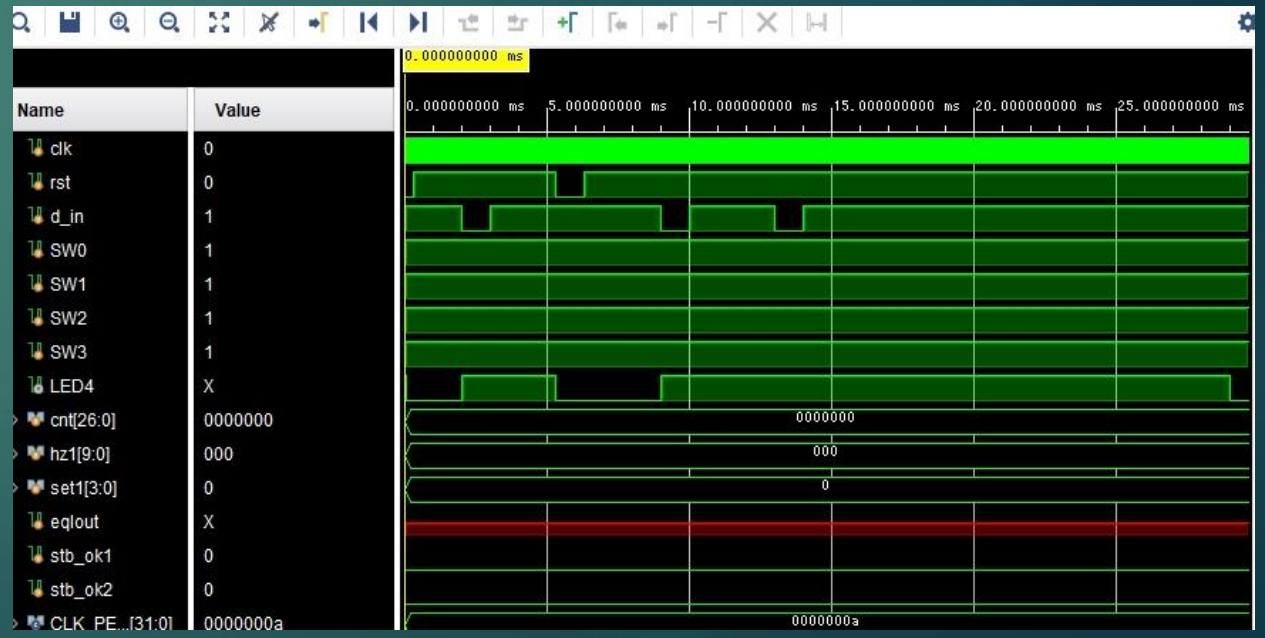
・シミュレーション　その２

さらに入力条件に変化を加えても問題が起きないようにコーディングを再度行いました。

カウント中にリセット信号(rst)が再度入力される。
その後 スタート信号(d_in)を再度入力する。



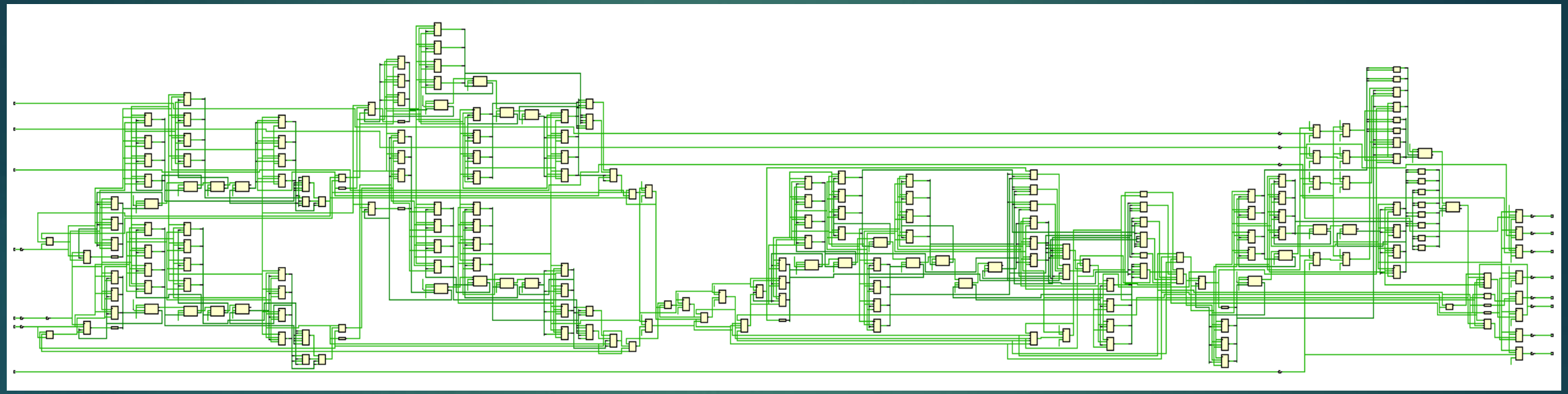
左の動作に加え
カウント中に再度スタート信号(d_in)を再度入力する。



問題なく動作できた

・論理合成、配置配線

論理合成を行い、作成したプログラムから下記の回路が生成されました。



配置配線を行う。

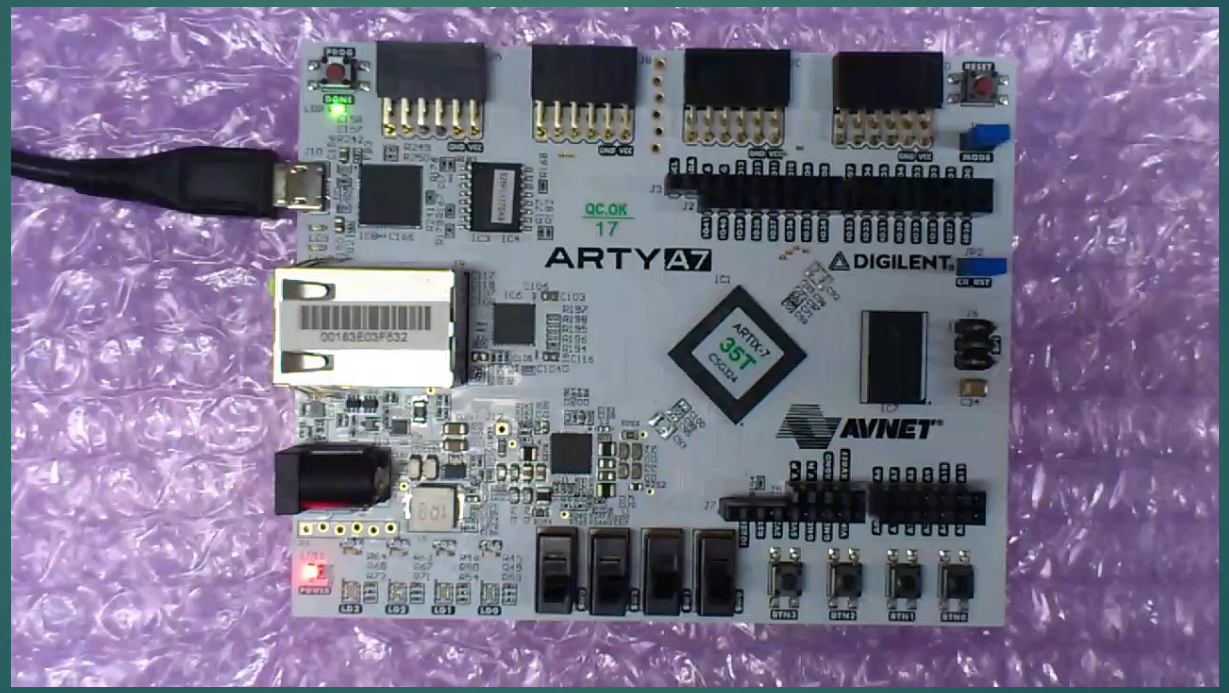
配置配線を行うには制約ファイル(.xdc)を作る必要があります。

制約ファイルとは、プログラムで使用している入出力ピンを実際のFPGAのピン番号、信号の種類等に紐づける情報を記載するデータになります。

・コンフィグレーション、動作確認

Vivadoで書き込み用のデータ（ビットストリーム .bit）を作成し、シミュレーションボードに書き込んで動作を確認しました。

イメージ通りの動きが確認できました。(8秒設定)



動画はこちらをクリックいただき、ご確認をいただけます。

・タイマー12Bit化

本来の回路はタイマー設定が12Bitだったが、シミュレーションボードは4Bitしかなかったので最初は4Bitで設計を行ってきました。

改めて12Bitのタイマーを作成することにしました。

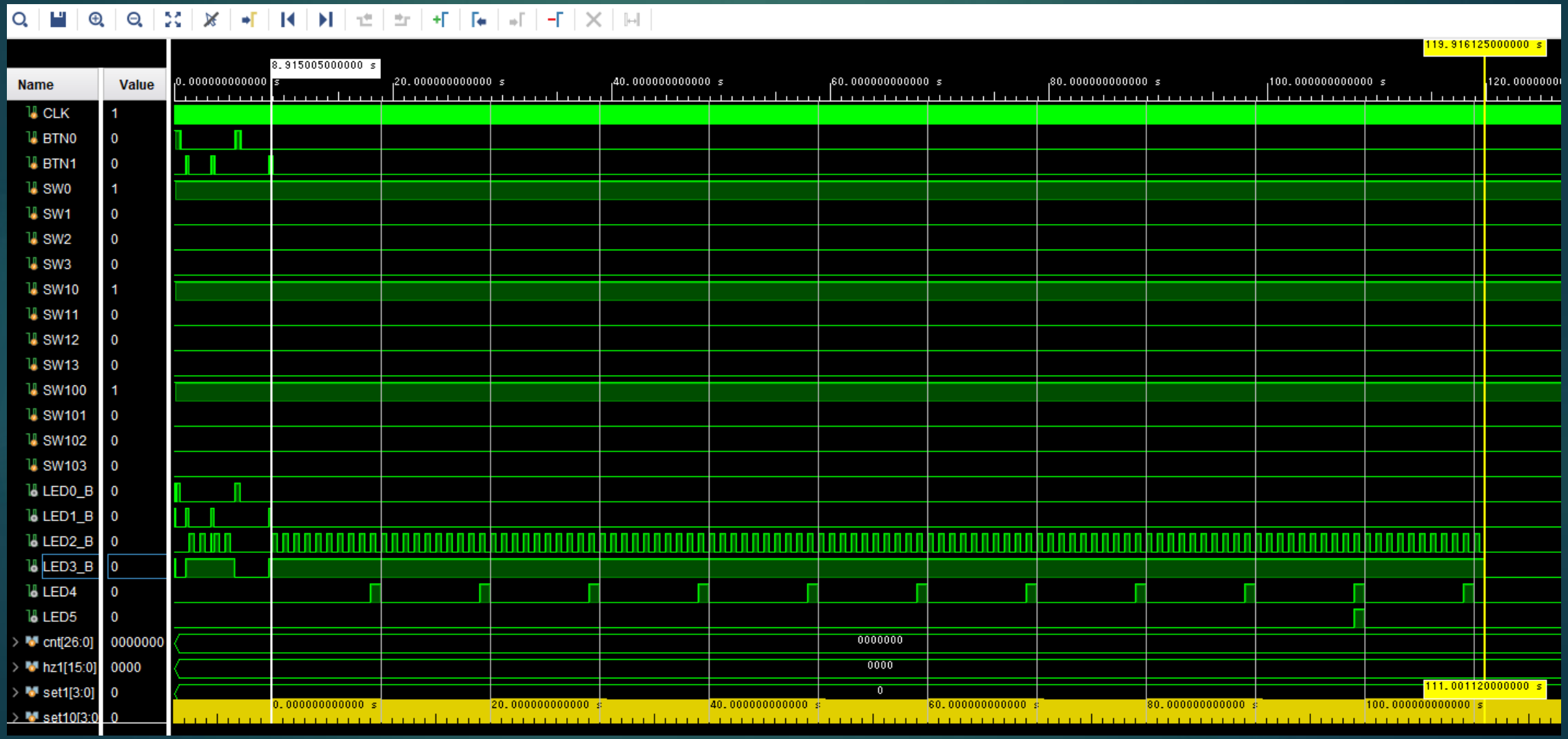
12bit　0～999秒の設定が可能

下位4Bit が 一の位:0～9秒、中間4Bitで 十の位:0x～9x秒、上位4Bitで 百の位:0xx～9xx秒

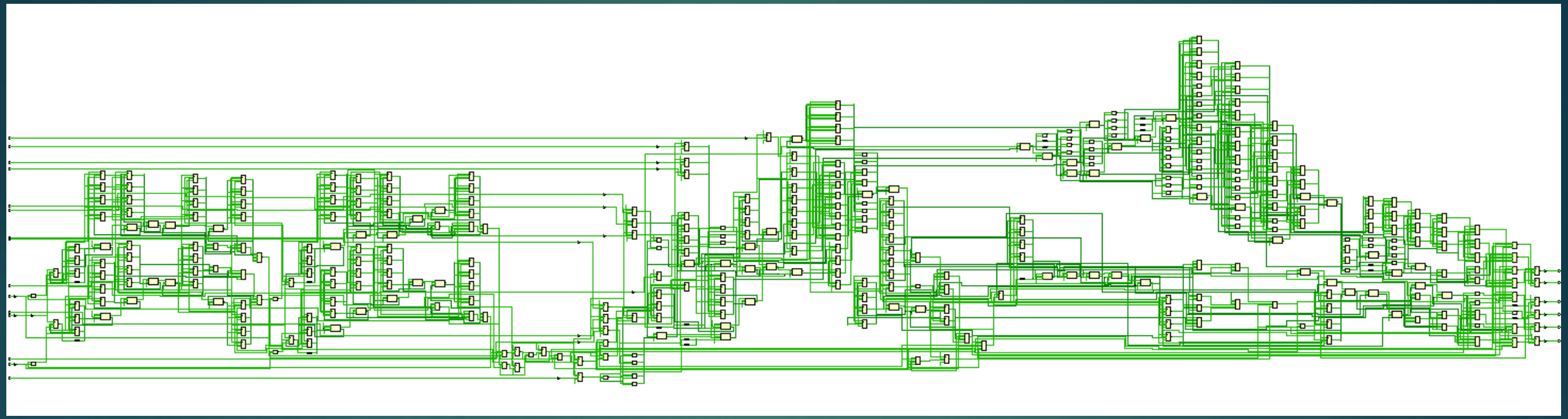
シミュレーションボードで確認する際は、外部ポートに新たに12Bit分のタイマー設定回路を追加する必要があります。

・タイマー12Bit化　シミュレーション

タイマーを12Bit化したときのシミュレーション結果。
タイマー設定 111s　　デバッグ用にLED2_B:1秒、LED4:10秒、LED5:100秒のパルス追加



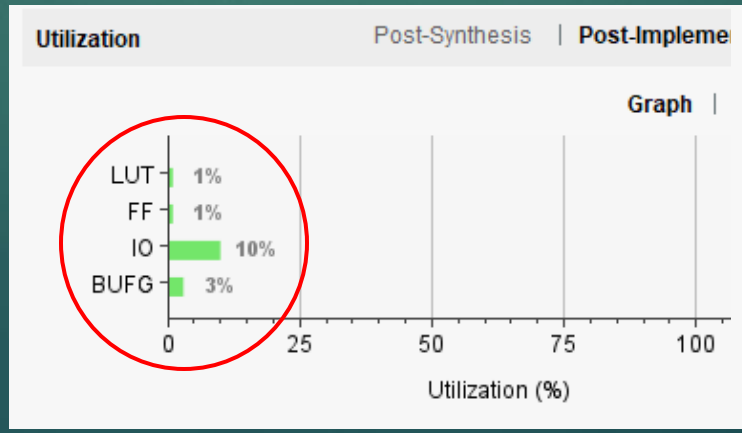
・論理合成　12Bit



今回使用しているFPGAで

この規模の回路では部品全体での

回路の使用率は右記の程度でした。



Utilization　Post-Synthesis | Post-Implementation

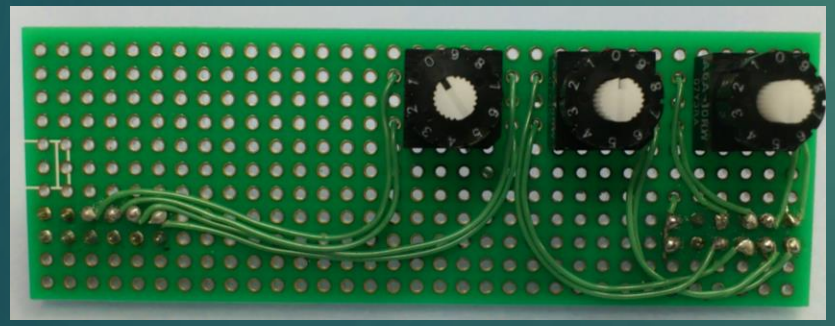
Graph | Table

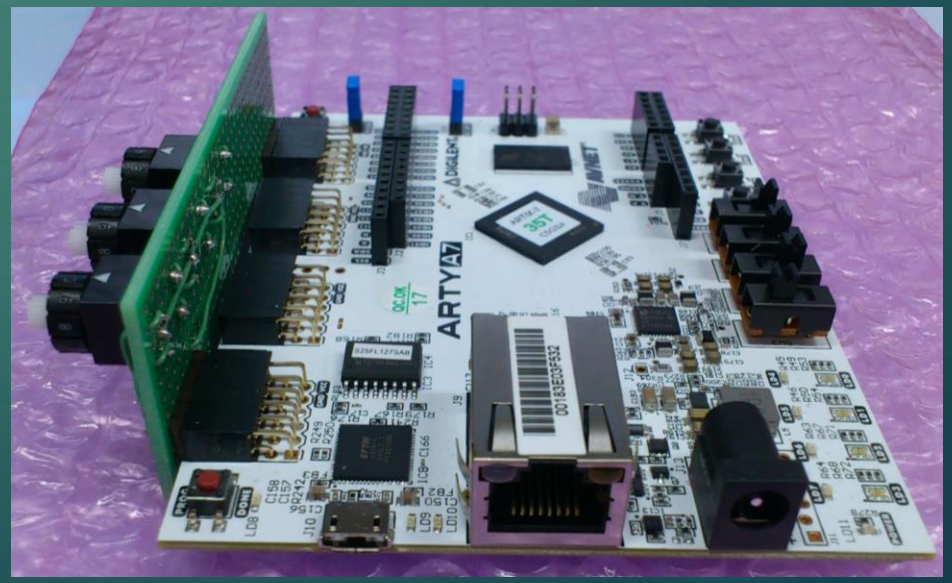
Resource	Utilization	Available	Utilization %
LUT	173	20800	0.83
FF	158	41600	0.38
IO	21	210	10.00
BUFG	1	32	3.13

・配置配線　12Bit化

シミュレーションボードに12Bitのスイッチを追加する為、
制約ファイル(.xdc)を変更しました。

12Bit タイマー設定回路を作成しました。

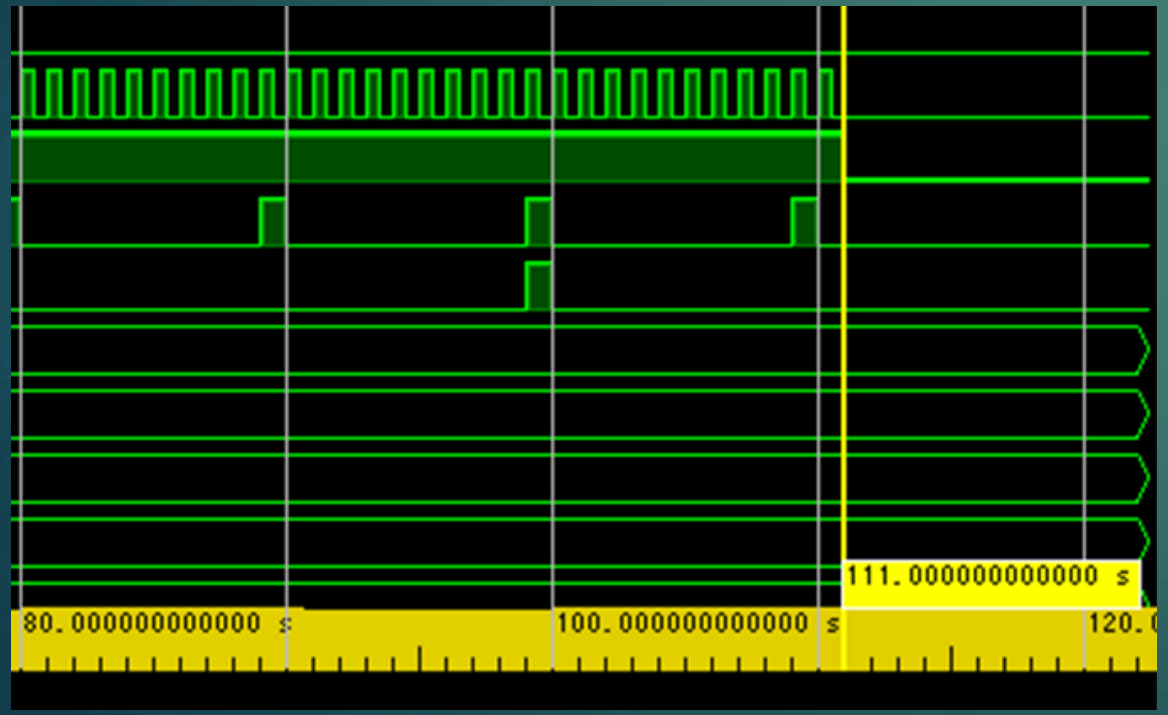


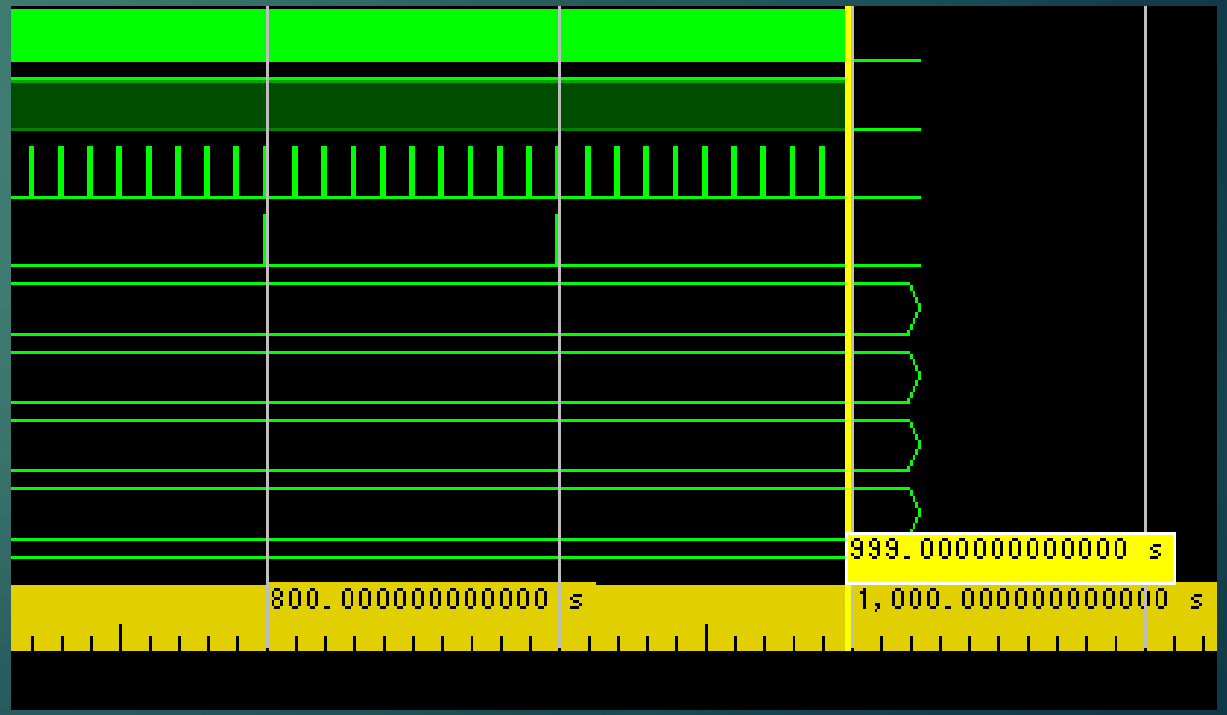


・12Bit　動作確認

ここまでにも細かい修正を行い、最終的なシミュレーション結果が下の図です。

シミュレーション結果、　111s, 999s　共に設定時間通り動きました。

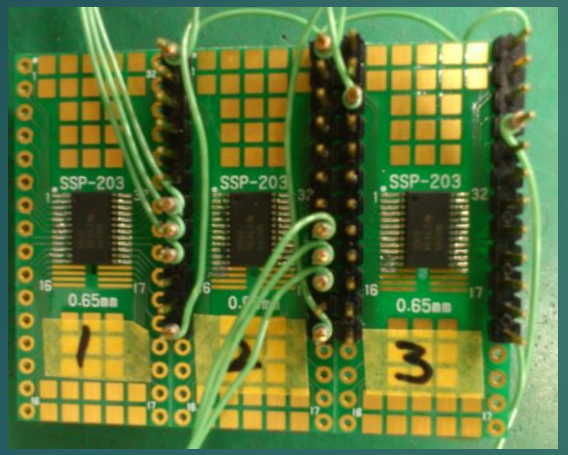


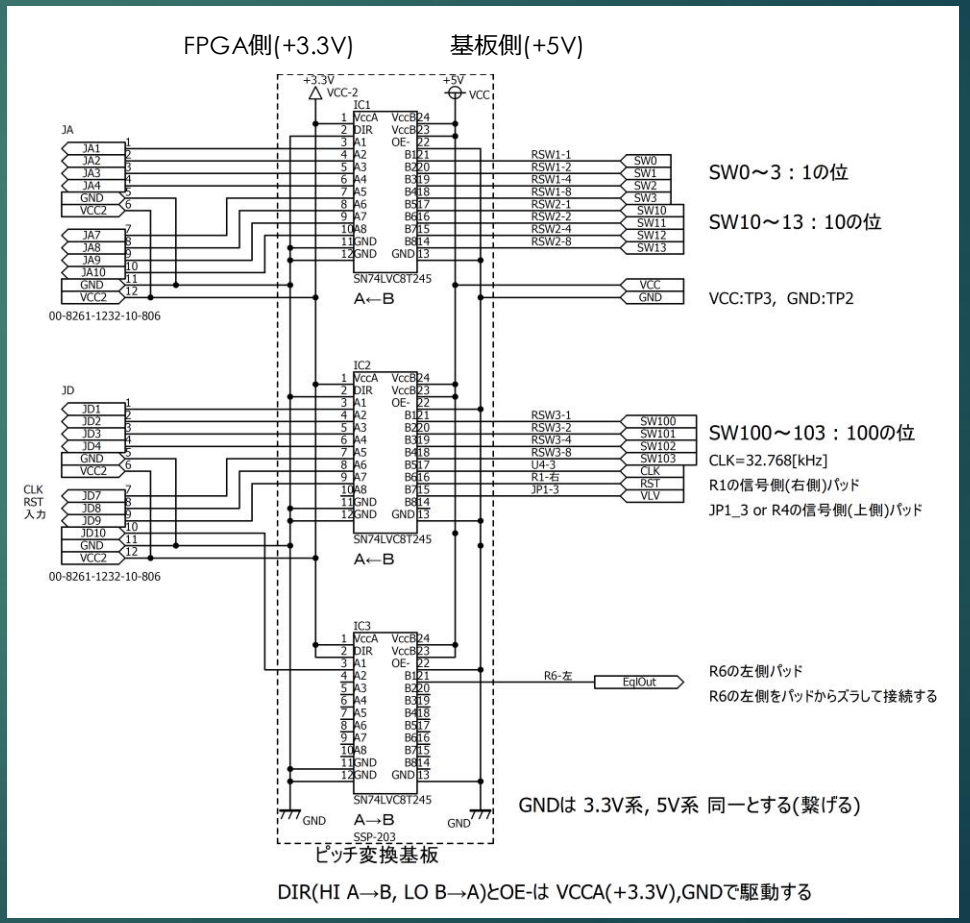


・実機確認

実機に合わせて制約ファイル(.xdc)の記述を変更しました。

レベル変換基板も作成しました。

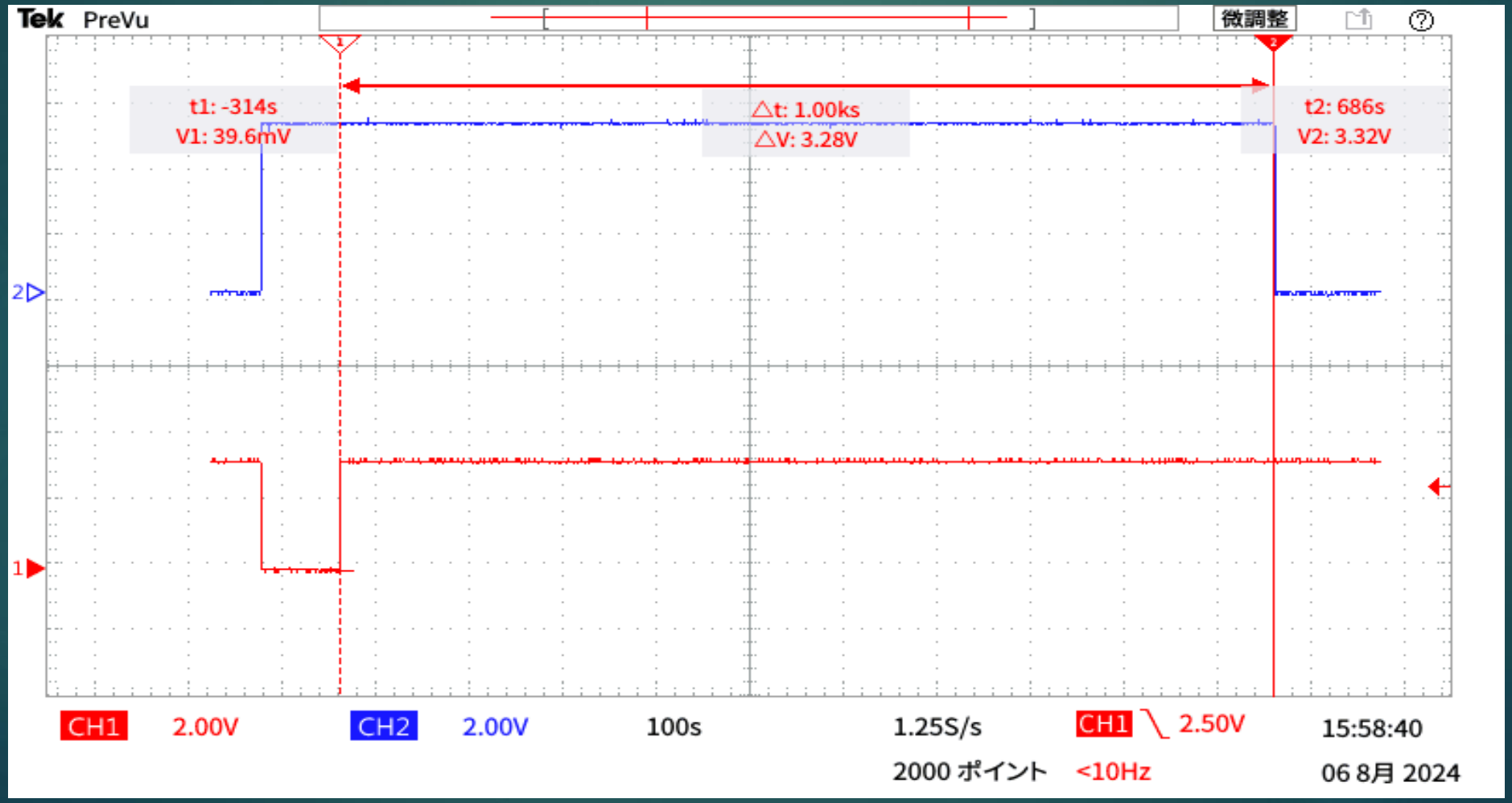




実際の動きを実演します。

・実機確認

オシロスコープで実機基板の信号を確認しました。
　設定時間: 999s　△t: 1.00ksとなっているが測定カーソルの微調整の問題あり。



・実機確認完了、所感

今期の目標であった実機確認は想定通りの動作を実現することが出来ました。

しかし、FPGAを動かすまでの一連の作業の習得であって、初めの一歩を踏み入れた程度の内容でした。

動かすことにのみに注力したが調べるほど奥が深い事がわかりました。

実際にFPGAを使用して回路を構築するにはまだまだ未知の部分も多く、この先も更なる技術習得が必要だと実感しました。

さらなる未知の部分がどれだけあるか、これを知るだけでも相当大変そうです。

以　上