

Tone Burst Converter

取り扱い説明書

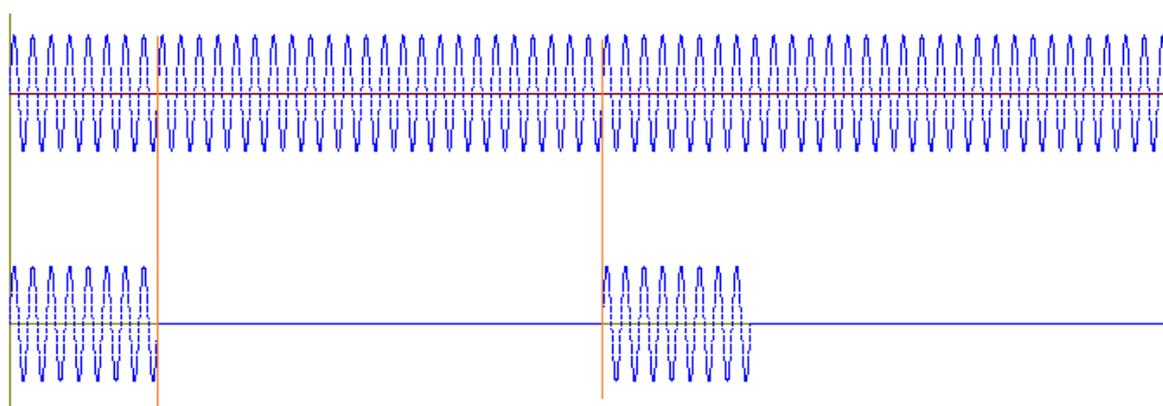
[1] この Tone Burst Converter の特徴	-----	1
[2] 動作の説明	-----	1
[3] 実動作の波形	-----	2
[4] 実装基板写真	-----	4
[5] 回路図	-----	5
[6] プログラミング	-----	6~12
[7] Windows 10 に ISE14.7 をインストール	-----	12

【 1 】 この Tone Burst Converter の特徴

- ・ 基板の寸法が 50 x 90mm で 10Hz~100KHz の TONE をバースト変換できます。
- ・ バースト(通過)する信号のダイナミックレンジは -5V ~ +5V と広いです。
- ・ JTAG 端子をコネクタで出しておりますので、書込器があれば、別途提供いたしますソースコード(VHDL)を変更して、バースト周期数と無信号周期数を変更することができます。

【 2 】 動作の説明

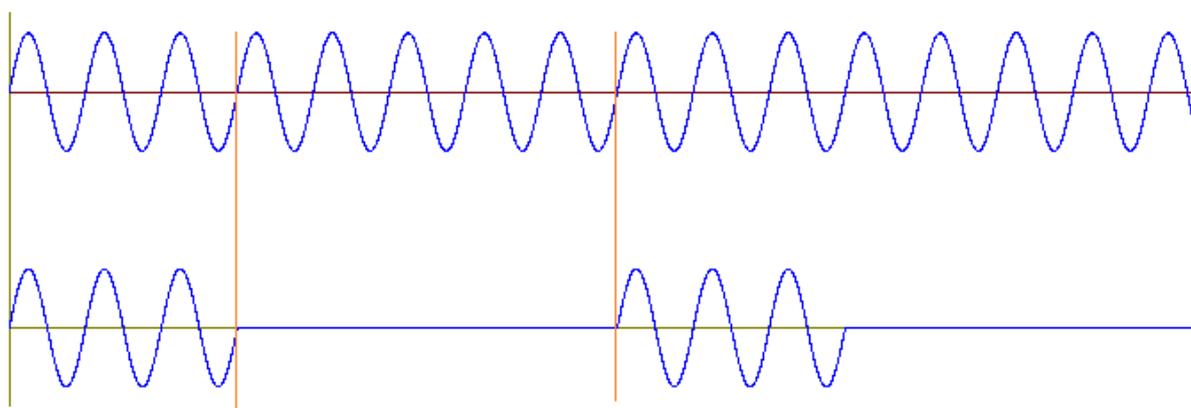
下図の図 1) は、Burst 周期数が 8 で、無信号(OFF)周期数が 24 の場合で、図 2) は、Burst 周期数が 3 で、無信号(OFF)周期数が 5 の場合です。 図中の上の波形は入力信号、下の波形は出力信号です。



Burst 8周期

OFF 24周期

図 1)



Burst 3周期

OFF 5周期

図 2)

出荷時の設定は 図 1) の Burst 周期数 : 8、無信号(OFF)周期数 : 24 となっており、これら

の周期数を変更するには、別途お渡しいたします ソースコード(VHDL)の定数を変更することで可能です (【6】プログラミング：参照)。

現在の設定は・・・

```
constant DAN : integer := 8;  -- Burst number  通過させるサイクル数
constant DBN : integer := 24; -- Off number   通過させないサイクル数
・・・となっています。
```

【 3 】 実動作の波形

各周波数に対する入力信号、コントロール波形および出力信号の様子を下図に示します。これらは全て Burst 周期数：8、無信号(OFF)周期数：24 のものです。画面の信号名は上から順に、入力信号・アナログ制御信号・出力信号・トリガ信号となっています

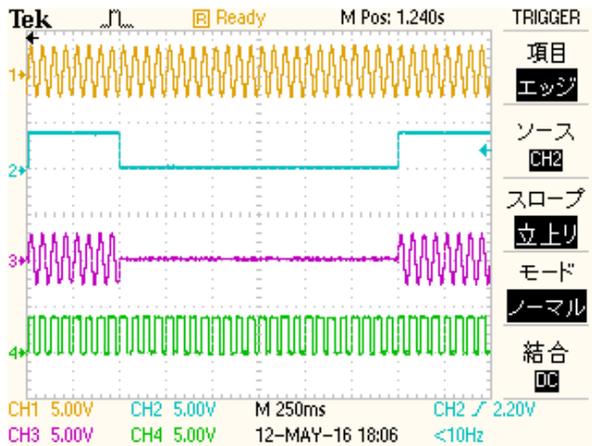


図 3) 入力信号 16Hz

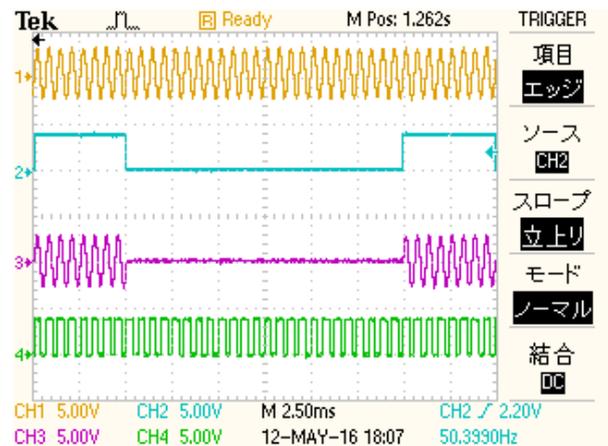


図 4) 入力信号 1600Hz

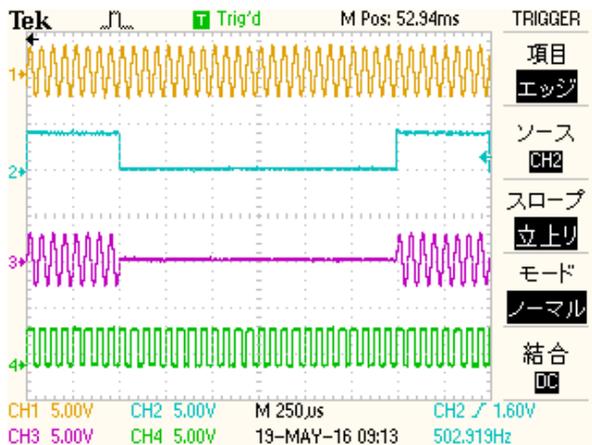


図 5) 入力信号 16KHz

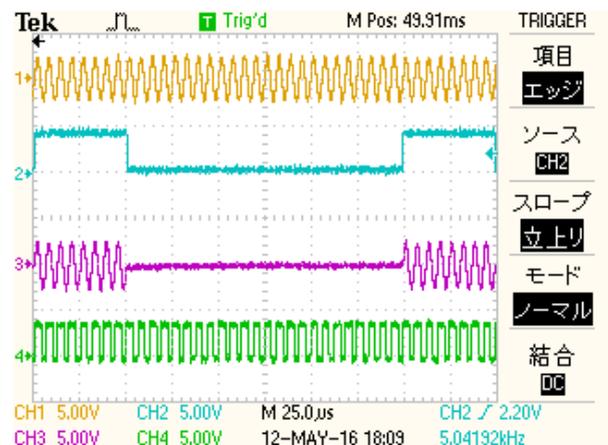


図 6) 入力信号 160KHz

バースト開始点と終了点を拡大して測定した様子を示します。

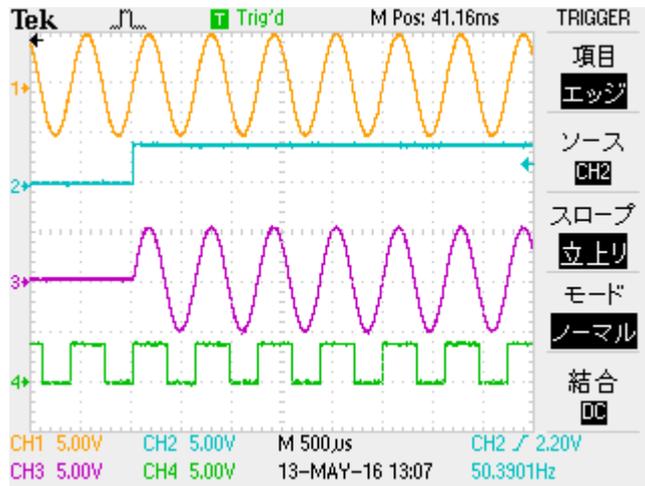


図 7) 入力信号 1600Hz の時のバースト立ち上がり部の拡大

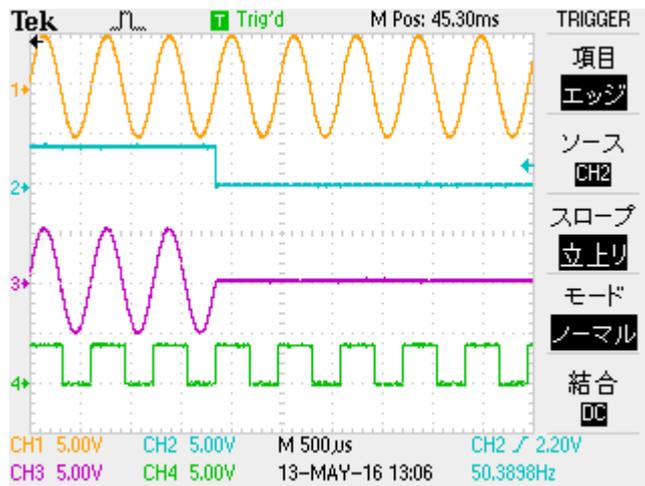


図 8) 入力信号 1600Hz の時のバースト立ち下がり部の拡大

【 4 】 実装基板写真

下図が実装した基板の写真です。

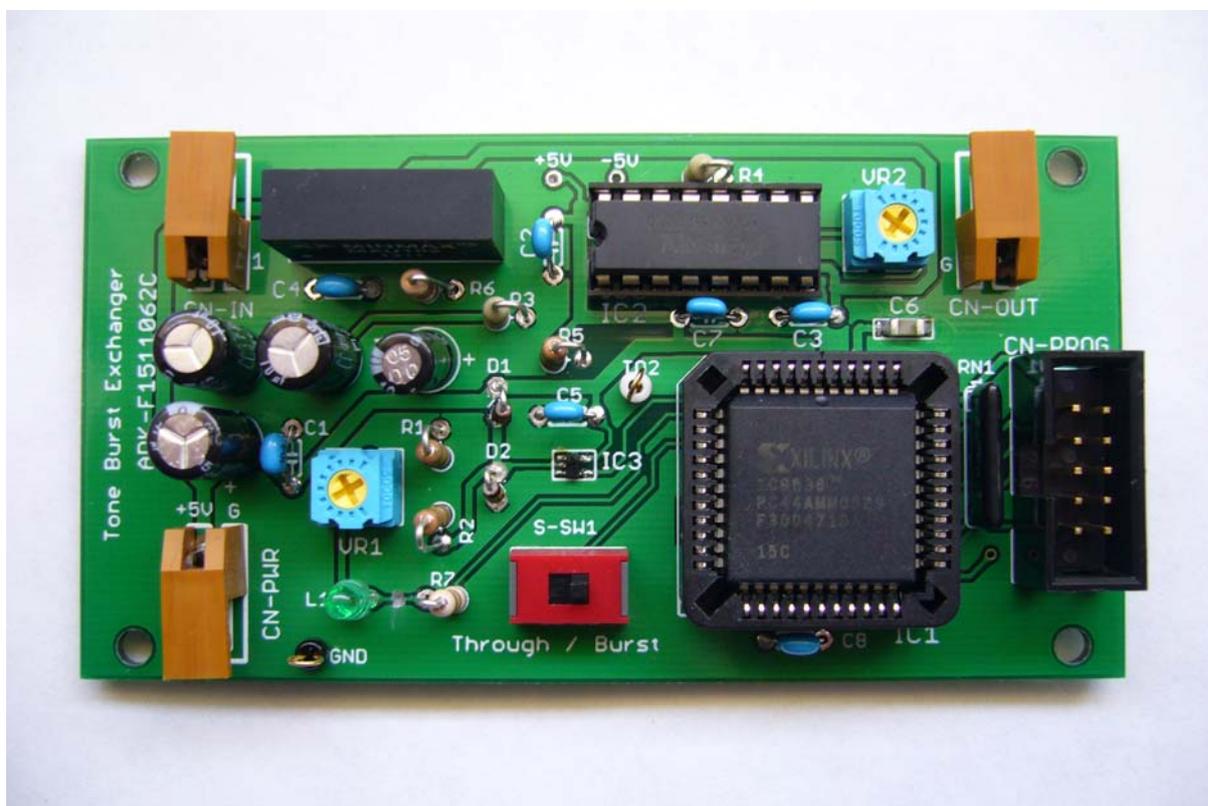


図 9) 実装基板

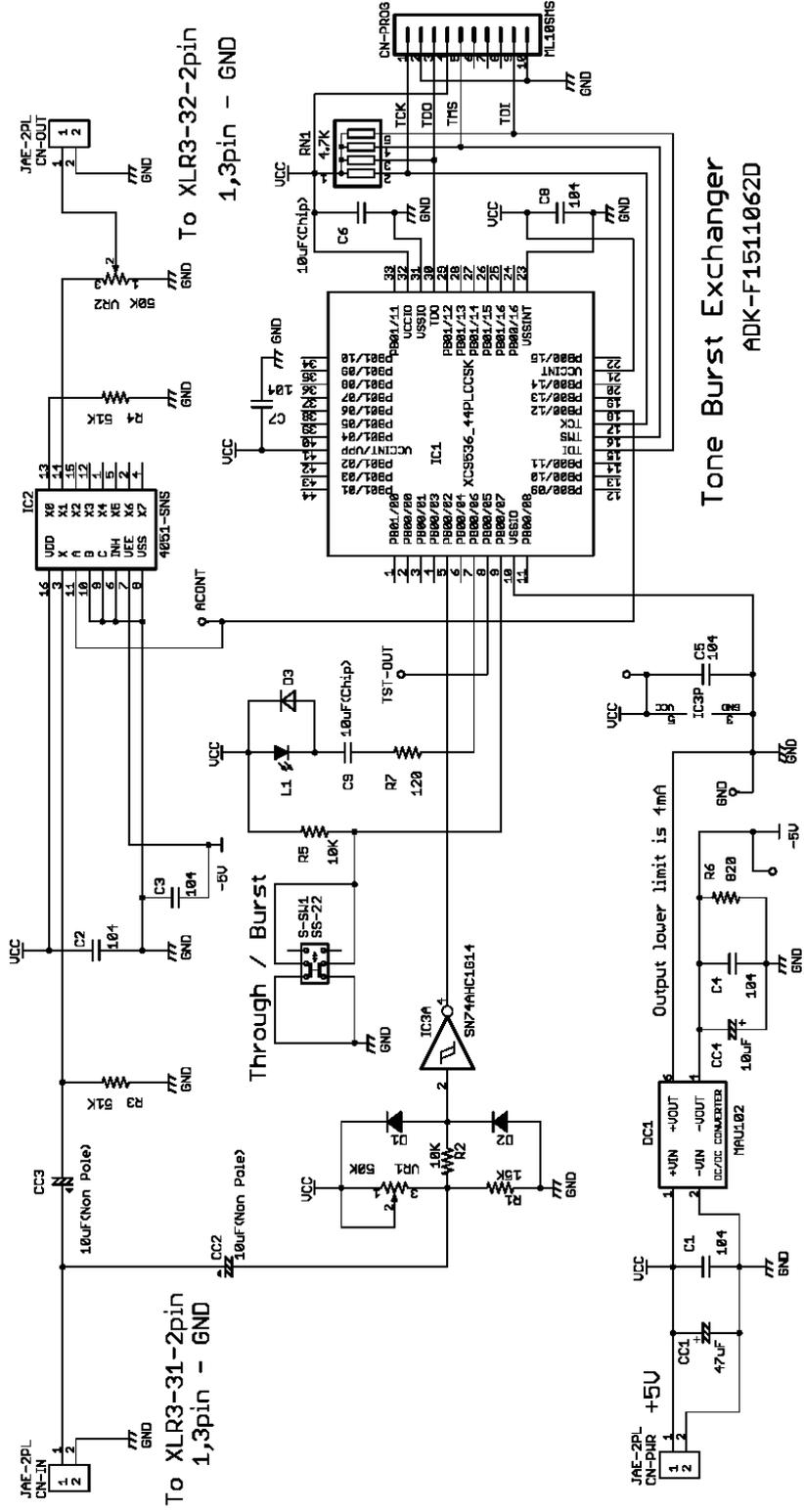
- CN-IN から入力信号を入れます。CN-OUT から出力信号を取り出します。
- CN-PWR から電源を供給します。電圧は 5V で、1PIN がプラスです。
- CN-PROG は CPLD をプログラムするための JTAG 信号が出ています。
- VR1 は入力信号からクロック(トリガ)を取り出すときのバイアスを変更します。
- VR2 は出力信号のレベルを調整します。
- L1(LED)は、入力信号にトリガ が掛かったときに点灯します。
- S-SW1(スライド SW)は、左側が全信号通過を指定し、右側で Burst を指定します。

電源の電圧や極性を間違えると故障いたしますので御留意ください。

JTAG の信号名(TCK,TDO,TDI,TMS,GND) については、回路図をご参照ください。

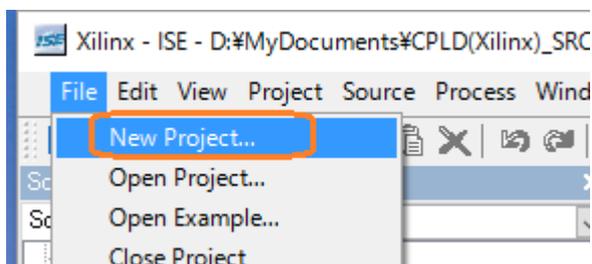
【 5 】 回路図

図 10) Tone Burst Converter の回路図

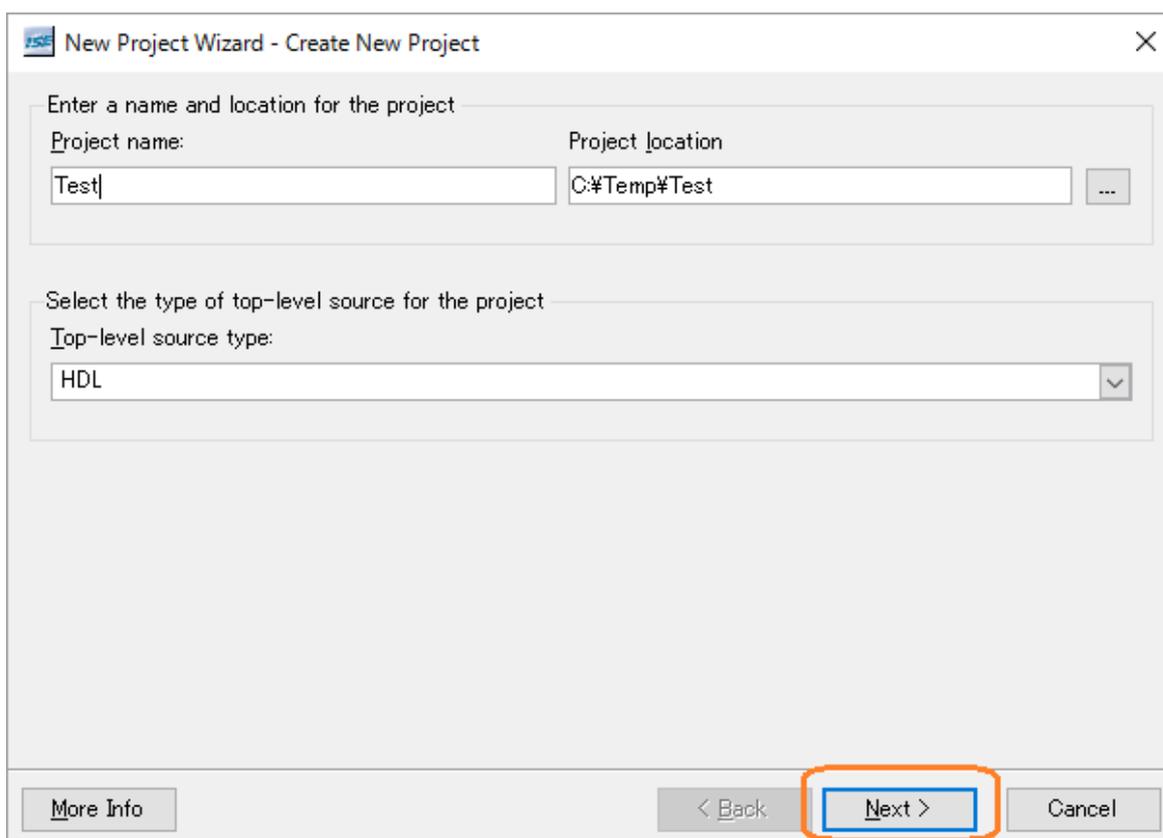


【 6 】 プログラミング

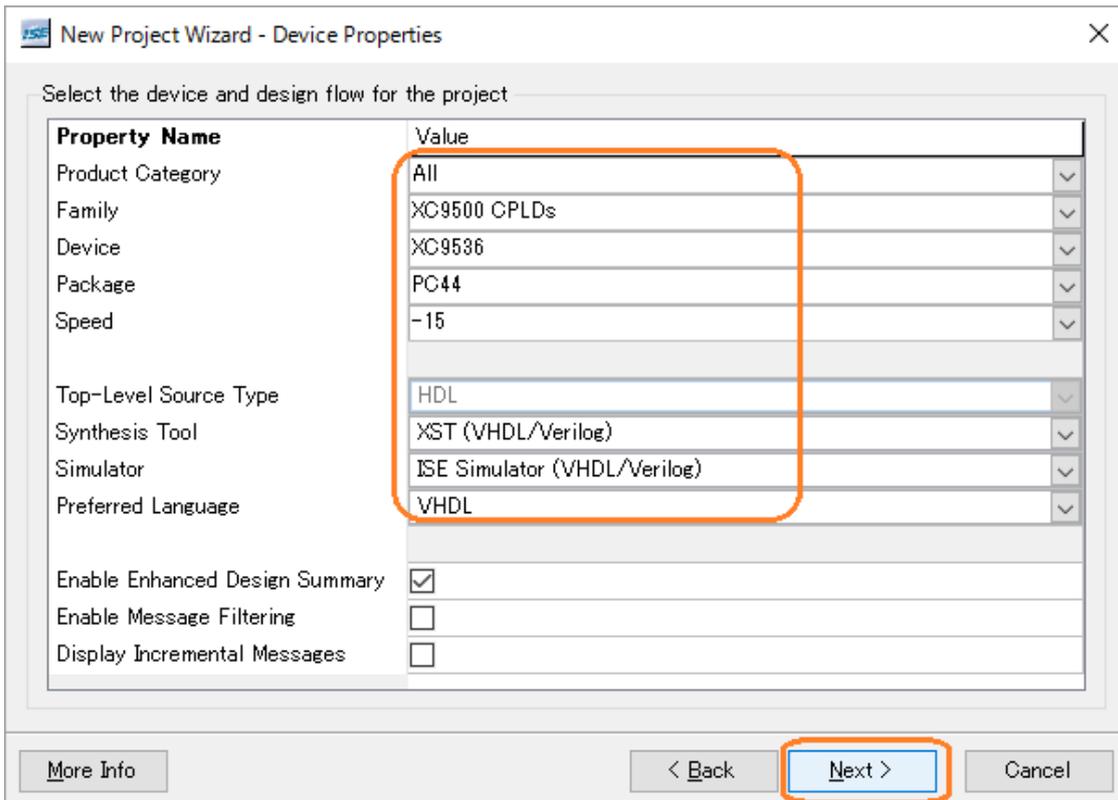
提供させていただいている ToneBurst12.vhd と ToneBurst12.ucf ファイルを準備してください。Xilinx 社の ISE を起動して、メニューの File – New Project... をクリックします。



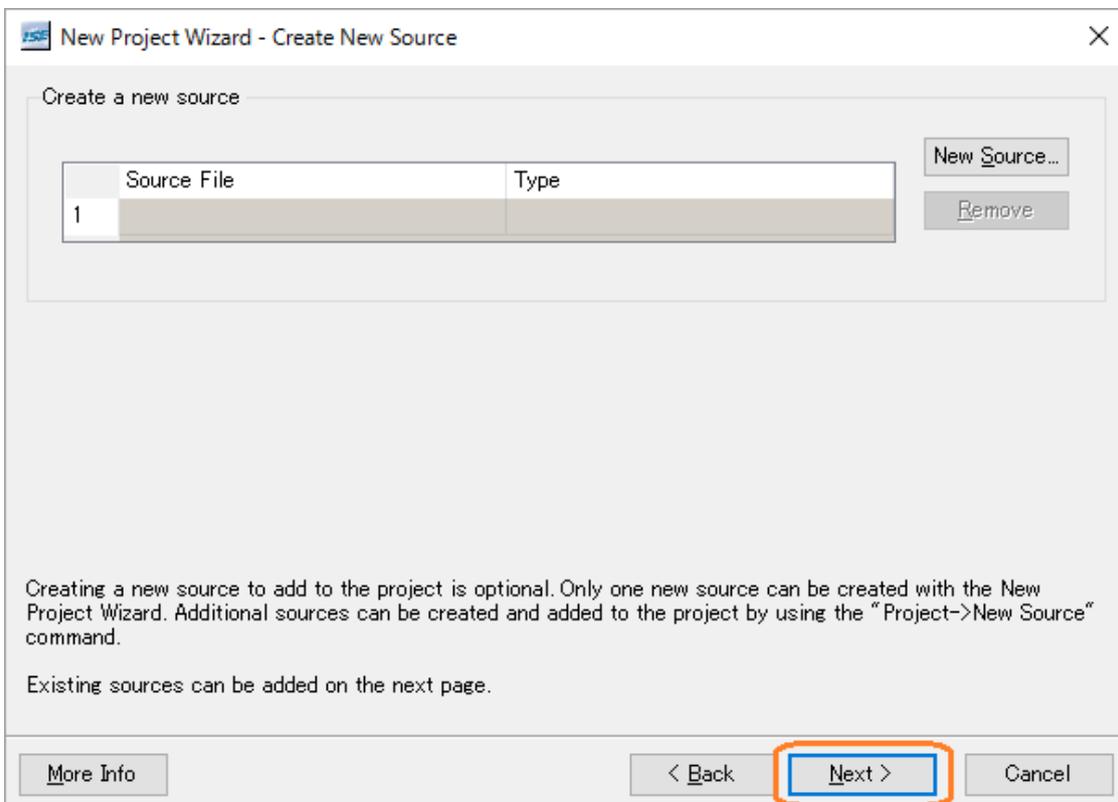
この ISE のバージョンは 10.1 で説明しますが、バージョンは違ってもほぼ同様な操作で作業できると思います。



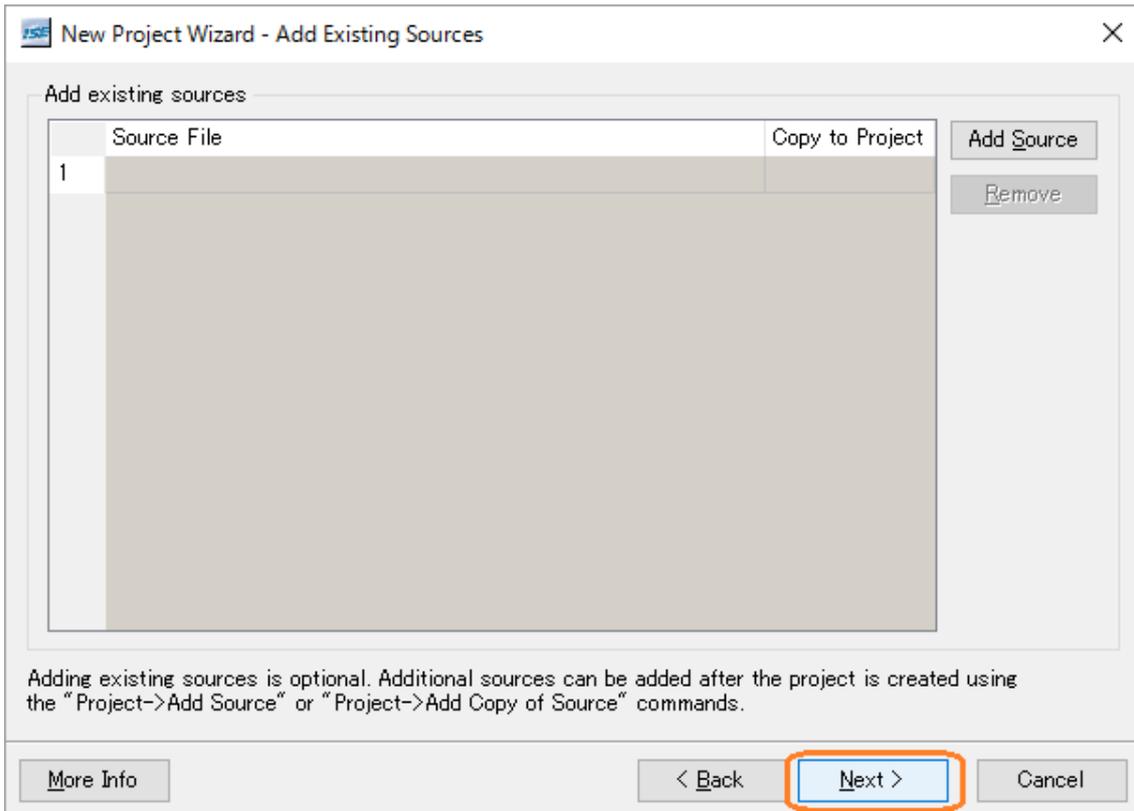
Project location にプロジェクトを置く場所を指定し(任意です)、次に Project name を入力します。ここでは、Test とします。すると Project location の後ろに Test が入ります。つまり、ホルダ名がProject名と同じになることに注意して、たとえばこれを ToneBurst12 とする事もできますが、ここでは Test のままで続けます。 Next > をクリックします。



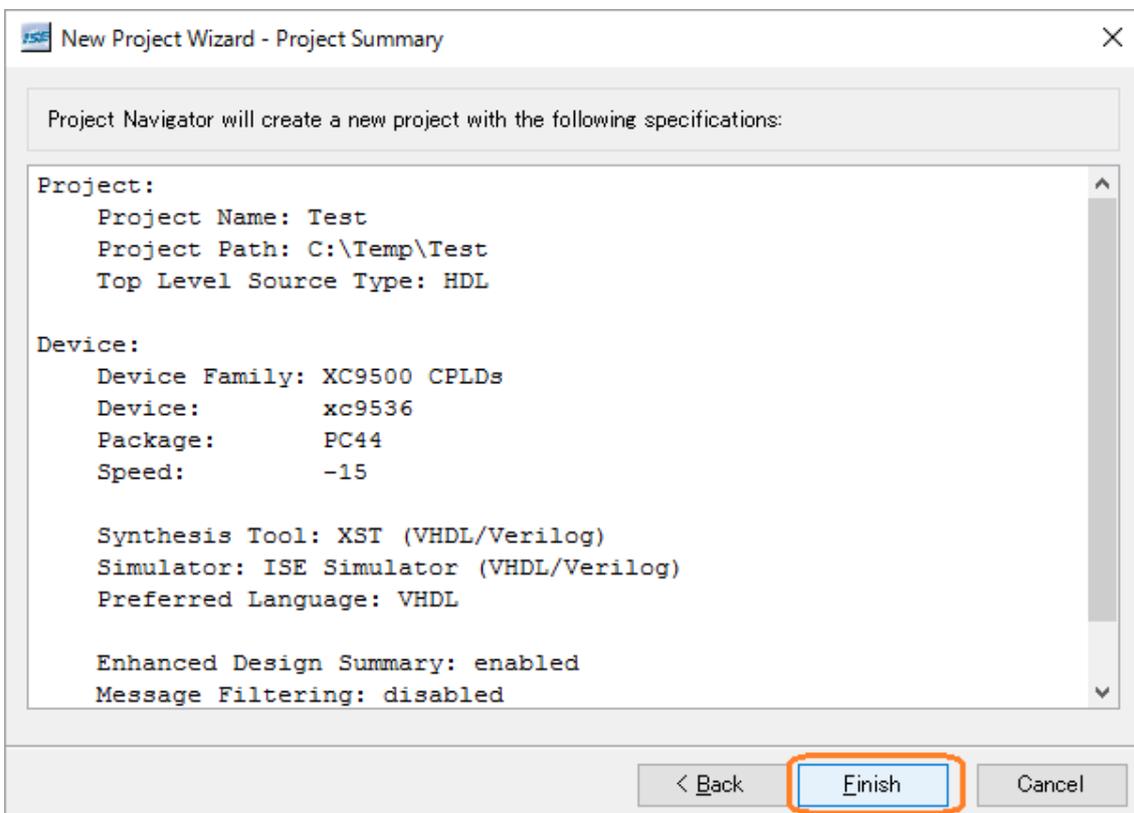
デバイスを上記の様に選択し Next > をクリックします。



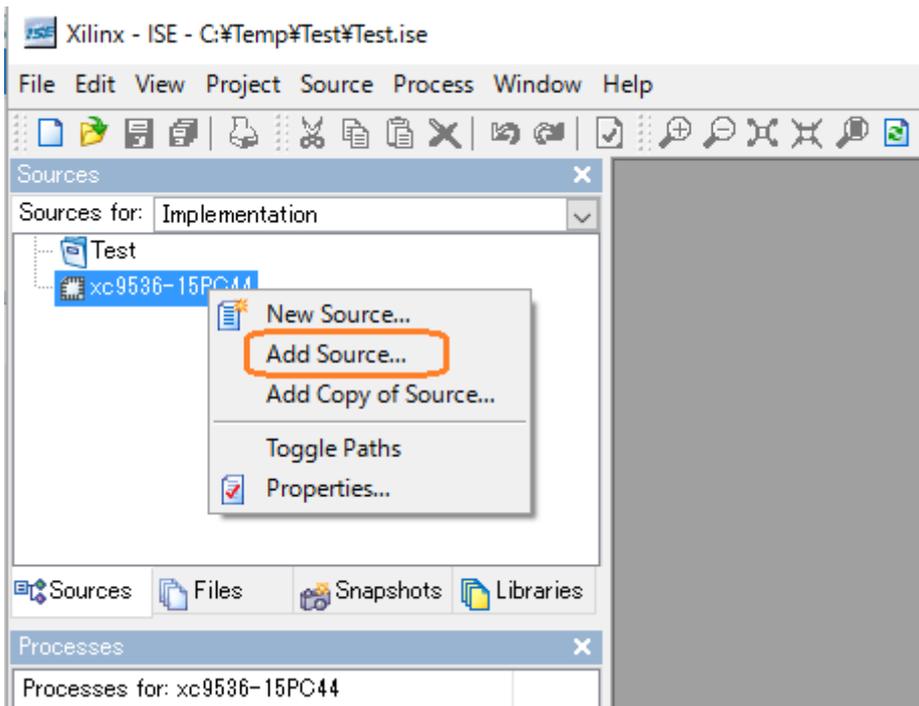
ここではソースを新しく制作しませんから、そのまま Next > をクリックします。



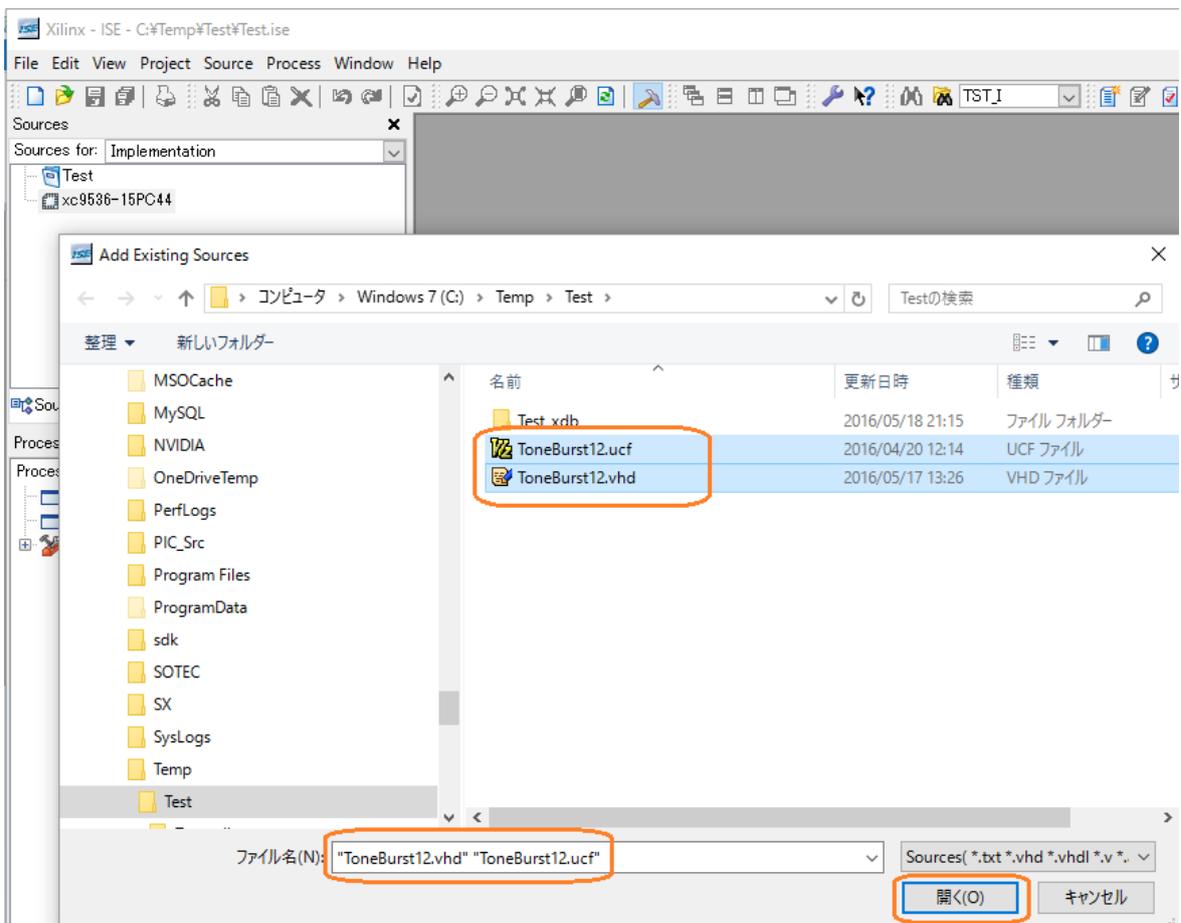
後で追加します。ここではソースを追加しません。そのまま **Next >** をクリックします。



確認です。良ければ **Finish** をクリックします。

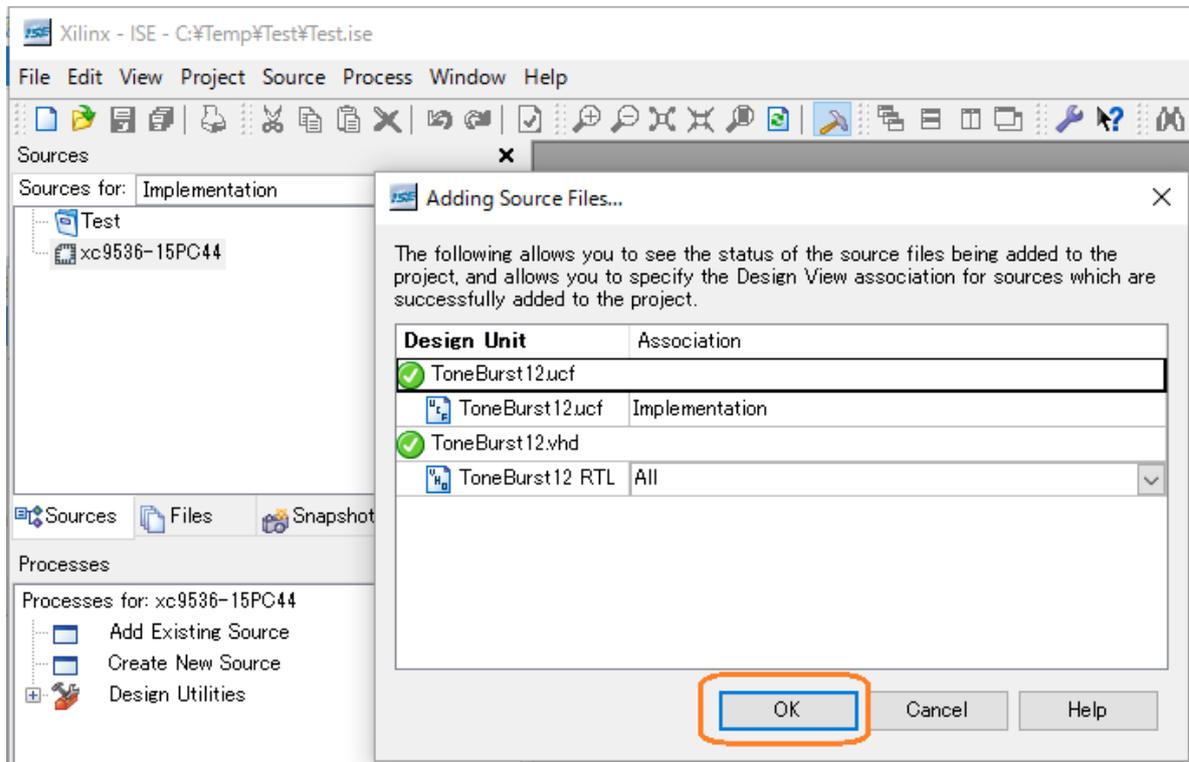


Source 窓の Xc9536-15PC44 を右クリックして Add Source... をクリックします。

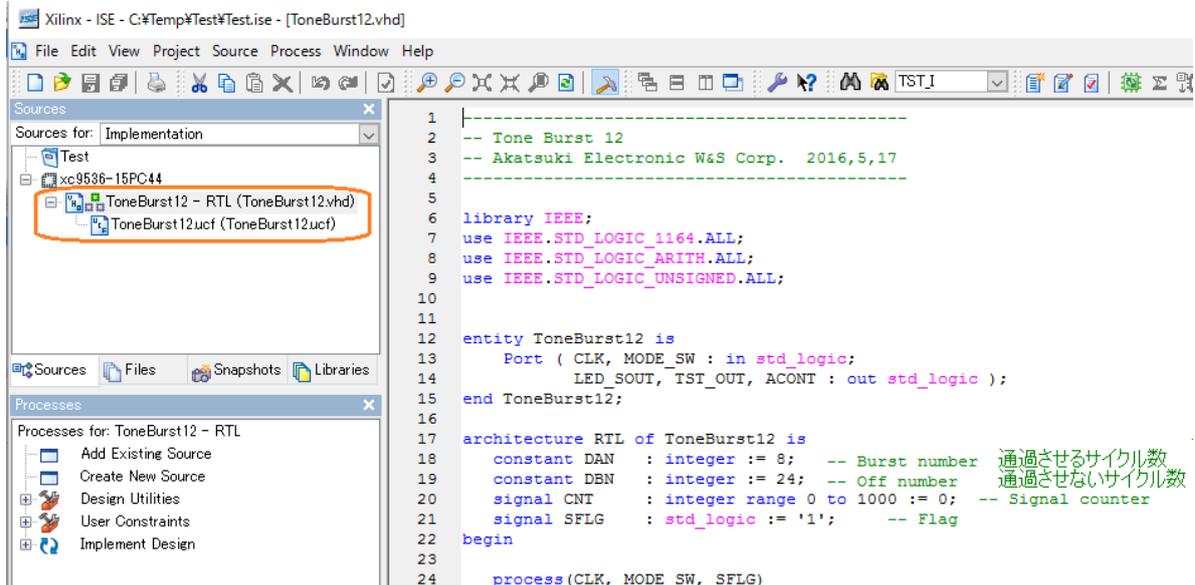


新しくできている「Test」ホルダに準備している ToneBurst12.vhd と ToneBurst12.ucf ファ

イルを入れて、この2つのファイルを選択(Ctrlを押したままそれぞれのファイルをクリック)して「開く」をクリックします。



上図の確認画面が出てOKをクリックすると・・・

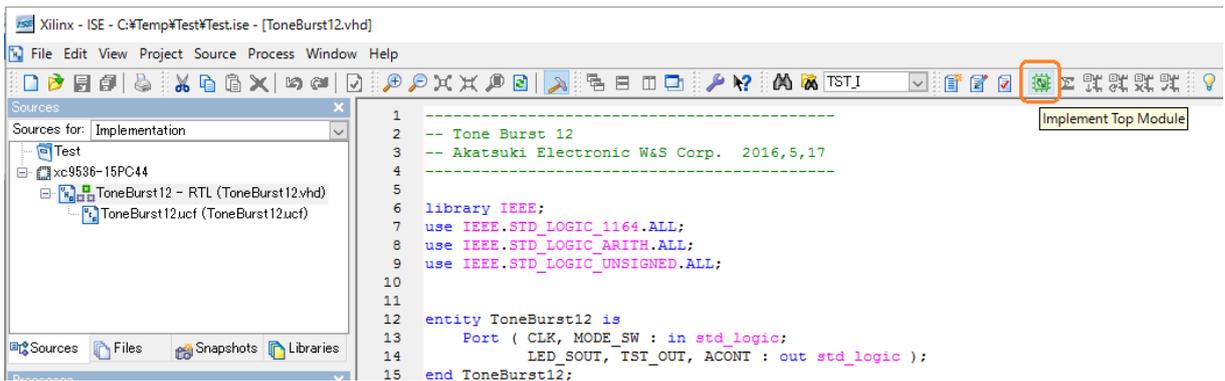


プロジェクトにファイルが取り込まれます。ToneBurst12 - RTL(ToneBurst12.vhd) をダブルクリックすると、右側にそのソースのエディタが出ます。

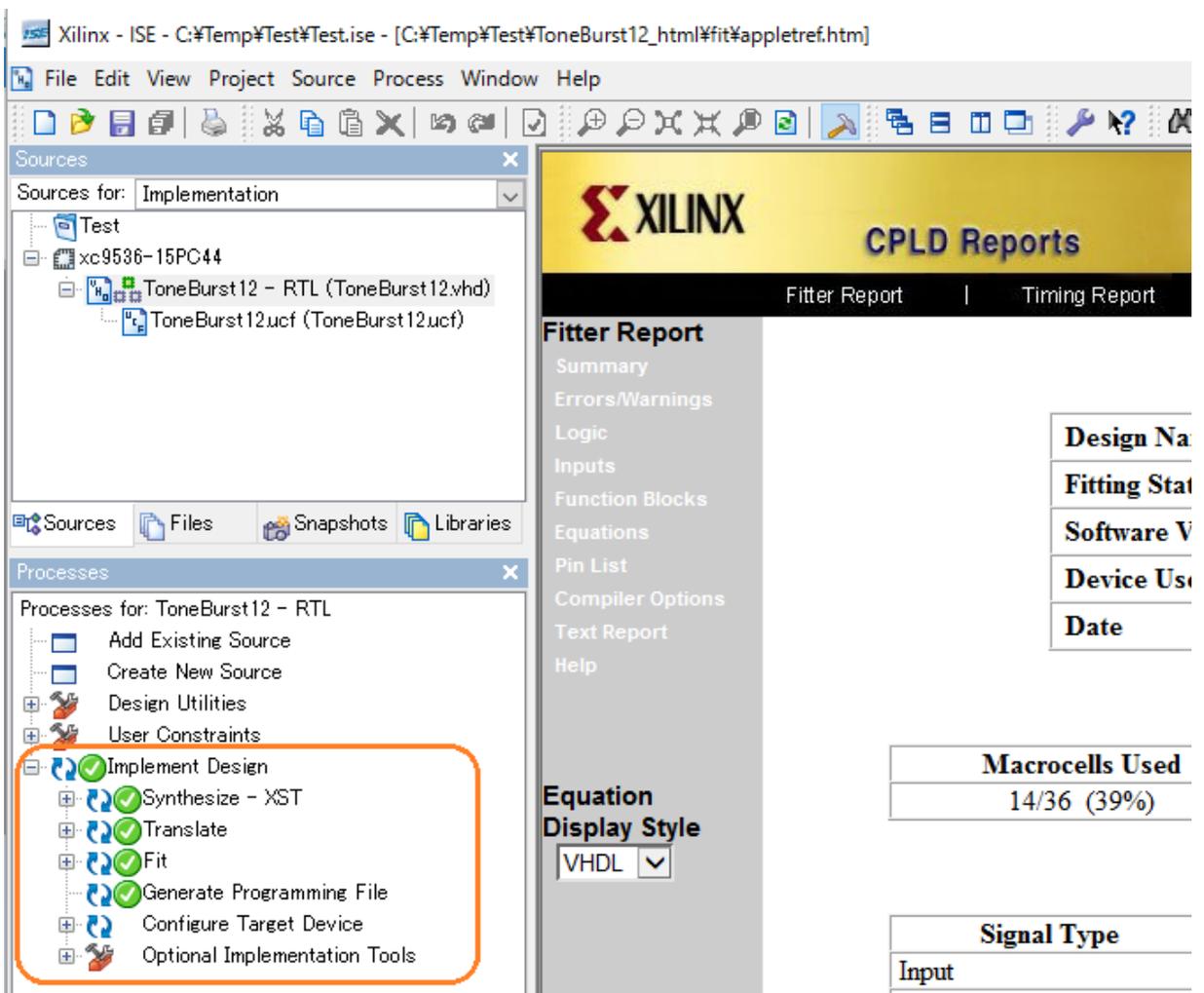
バーストサイクル数と無信号サイクル数を変更するには・・・

constant DAN : integer := 8; -- Burst number 通過させるサイクル数
constant DBN : integer := 24; -- Off number 通過させないサイクル数

・・・この数値を書き換えます。



メニューの Process – Implement Top Module を実行するか右上の朱角のマークをクリックすると Implement 動作が始まります。



上図のように Processes 窓の Implement Design のチェックが全て緑となれば成功です。これでホルダ内には ToneBurst12.jed (書き込み用)ファイルができていますから、これを書込器から指定して書き込めばOKです。



上図は、書き込み時のセッティングの例です。

【 7 】 Windows 10 に ISE14.7 をインストール

Windows10 で ISE 14.7 を動作させるには、以下のリンクに詳しい解説があります。

<http://www.haljion.net/index.php/2013-04-12-08-15-44/82-2015-03-26-06-32-38/fpga-cpld/369-window-s10-ise-webpack>

日本語でここまでの詳しい解説はそうはありません。情報の提供、ありがとうございます！
OS が 64 ビットでの解説です、32 ビットの場合は、ホルダ名 "nt64" を "nt" に読み替えば良いようです。このバージョンでは従来のプロジェクトを読み込むと拡張子が、.ise から .ixise に変更されます。

また、ISIM が起動しない・・・については下記リンクを参考にしてください。

http://openit.kek.jp/training/2014/fpga/docs/Windows8_trouble.pdf ・ ・ ・ありがとうございます。

提供致していますソースによる移植は、ほぼ上記で解説した手順でプロジェクトを制作することが確認できました。