

# POCO の論理合成

## 論理合成とは

今までは、Verilog はハードウェア構造を記述し、シミュレーションするだけだった。しかし、記述されたハードウェアは何らかのデバイス上で実現しなければ意味がない。このためには Verilog 記述をゲート部ベルの接続図(ネットリスト)に変換し、様々の最適化を行う作業が必要である。これを論理合成、圧縮と呼ぶ。今日はこの合成ツールの使い方を習得し、これを使って今まで設計してきた POCO を合成、圧縮することが目的である。

## 論理合成圧縮ツール

論理合成圧縮ツールは、プログラミング言語におけるコンパイラに相当するが、対象がハードウェアであるために、コンパイラよりもはるかに処理が複雑で、時間がかかり、また、ツール自体も高価である。現在、学生レベルで無料あるいは安価に使える論理合成圧縮ツールには以下のものがある。

- FPGA ベンダの合成ツール、Xilinx 社の ise、Altera 社の Quartus II などは、web パックと呼ばれる簡易版は無料である。web パックは一部の機能が制限されているが、POCO レベルの回路ならば十分合成可能である。
- PARTHENON、この授業で、以前、用いていた合成ツール。元々 NTT が開発したが現在は NPO 法人が管理し、無料である。しかし、独自の記述言語である SFL のみを受け付け、Verilog は受け付けない。

ここでは、タダで誰でも使えるということから Xilinx 社の ise/WebPack を用いる。これは皆さんが自分の PC にダウンロードして使うことができる。設計には制限はあるが、演習程度には十分である。

## ise の使い方

synth.tar をダウンロードして展開する。このディレクトリで、

### ise

と打ち込むと立ち上がる。Tip や ise の宣伝の窓が立ち上がるが、これらは全て閉じてしまいメインのウインドウである ISE Project Navigator を前面に出す。

File→ New Project を選択

作業を行うディレクトリを指定する。

Next を選択し、デバイスを指定する。ここではもっとも安くて一般的な Spartan-3E ファミリーを選ぶ。ここでピン数が豊富な TQ144 というのを選択して欲しい。

Next を選択し、New Source という選定はパスし、再び Next を選択する。Add Source が出てくるので、poco1.v、alu.v、rfile.v を全て指定する。(def.h は指定しない)

Next、Finish を選択し、初期設定を終える。うまく行くと緑色のチェックのついた表が出てくるので、OK を押してやる。

これで準備は完了である。左上の窓に設計階層が見えると思う。左下の窓の Synthesize-XST の所を右クリックして Run してやると合成がスタートする。次に Implement Design を同様に Run すると、配置配線が行われる。全て終了すると緑色のマークが付く。

本当に FPGA に設計データを送る時は Generate Programming File をクリックして JTAG などを経由して配置配線情報を転送してやるが、ここでは実機がないのでそこまでやらない。

右の窓の Design Overview の Summary の下の方を見てみよう。利用されている LUT 等が表示される。流石にこの FPGA は小さいため、かなりの論理が利用されているのがわかる。次に、Detailed Reports の所を見てみよう。

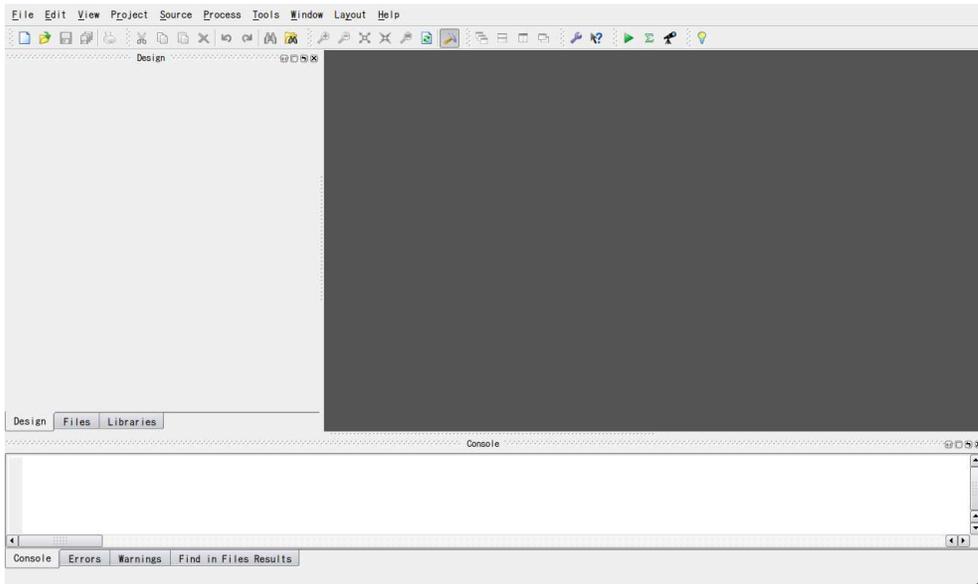


図 1: ISE Project Navigator

Create New Project  
Specify project location and type.

Enter a name, locations, and comment for the project

Name:

Location:

Working Directory:

Description:

Select the type of top-level source for the project

Top-level source type:

図 2: 作業ディレクトリの指定

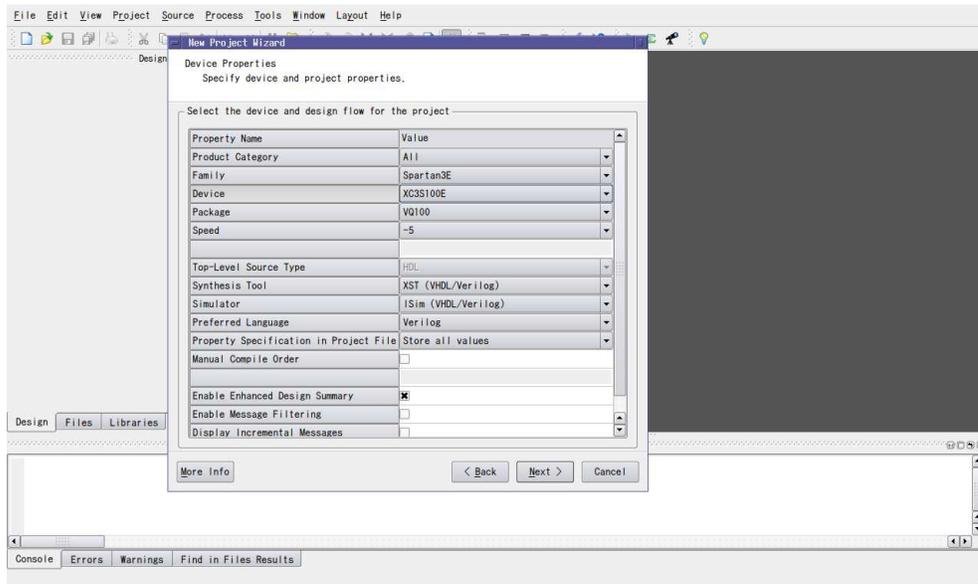


図 3: デバイスの指定

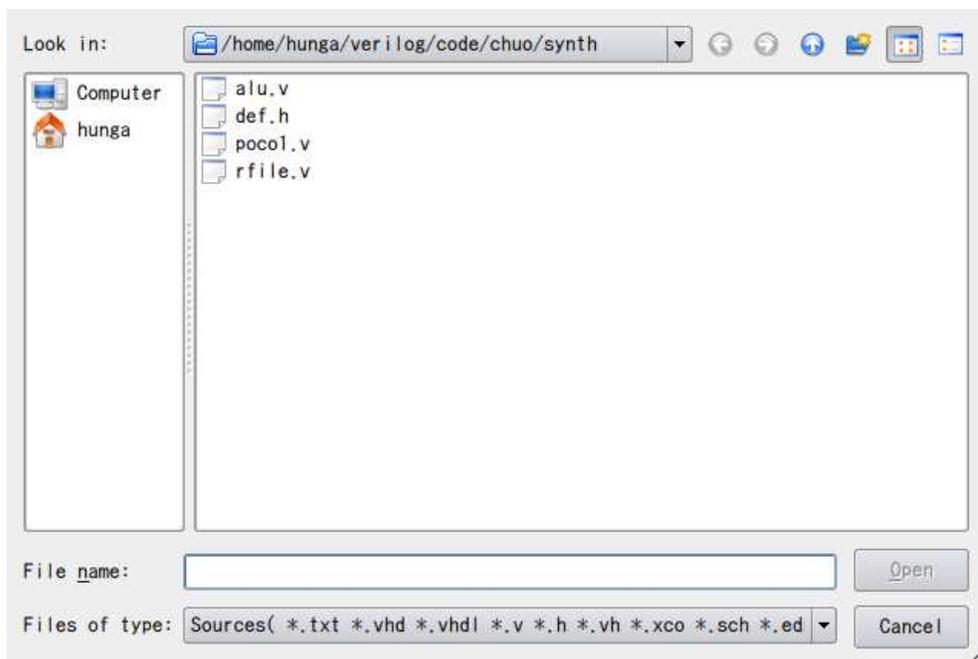


図 4: ファイルの指定

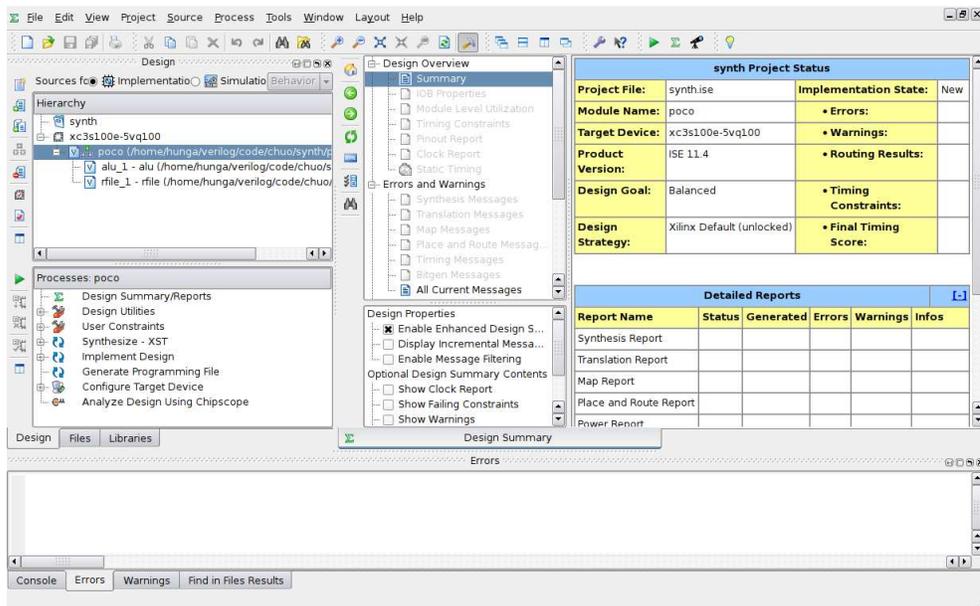


図 5: 設計初期画面

Post-PAR Static Timing Report をクリックすると、配置配線後の遅延が表示される。これを読み取ってどの程度の周波数で動作するかを見てみよう。

#### 演習

さすがにこの FPGA は小さすぎるようである。SPARTAN-3A、Virtex-5 などで合成して結果を確認せよ。