Z80 互換のソフト CPU コア
「T80」の使い方.

概要

1.1 はじめに

Z80 互換のソフト CPU コア「T80」を FPGA(アルテラ社の Cyclone)に入れて動作させてみました。ポケットロジアナ IP で波形を測定しています。保証もサポートもありませんので、自己責任でお使いください。

下記のホームページを参考にしています。

1 T80 ソース

OPENCORS

http://www.opencores.org/

OPENCORS の T80 のページ

http://www.opencores.org/projects.cgi/web/t80/overview

2 T80 の使い方

Cyclone で Z80 互換の CPU を動かす。

http://hijiri.yitc.go.jp/mb/index.php?T80%2F%A5%CF%A1%BC%A5%C9%A5%A6%A5%A8%A5%A2

FPGA-Donkey Kong

http://office-dsan.hp.infoseek.co.jp/dkong/dkong_prj.htm

Veritak の T80 のベンチマーク

http://japanese.sugawara-systems.com/opencores/benchmark/benchmark_z80.htm

おかもとシステムの SPARTAN-3 に Z80-IP(T80)を実装

http://www1.ocn.ne.jp/~oksys/xilinx/xpg4.htm

3 Z80 用 C コンパイラ SDCC の使い方

C コンパイラ SDCC のインストールとポケコン PC-E200 での動作確認

http://noritsugu.at.webry.info/200707/article_1.html

T80-ソフトウエア

http://hijiri.yitc.go.jp/mb/index.php?T80%2F%A5%BD%A5%D5%A5%C8%A5%A6%A5%

A8%A5%A2

Z80 の C 言語クロスコンパイル(SDCC)
http://www.markn.org/blog/z80/

1.2 サンプルファイル

今回作成したサンプルのファイル構成です。T80_sample.lzh を解凍すると、T80_sample フォルダが作成されます。T80_sample を適当なフォルダ（ここでは C:¥work）に置いてください。

C:¥work¥T80_sample
¥T80_io FPGA 用ソースファイルです。Quartus Ⅱ Ver6.0 対応
¥SDCC プログラムソースです。C コンパイラ SDCC 対応
¥波形 ロジアナ IP で測定した T80 の波形

1.3 仕様

T80_sample を下記基板で動作させました。
ヒューマンデータ製 FPGA 基板 CSP-024-6 Cyclone EP1C6Q240C8 搭載

T80 のクロック 40MHz
T80 のプログラムメモリ 2Kbyte
T80 とロジアナ IP が使用するロジックエレメント数 3077（51%）
T80 とロジアナ IP が使用するメモリ 59776（65%）

ソフト編

ソフト編を飛ばして、次のハード編を先に実行することもできます。

2.1 SDCC のインストール

下記ホームページを参考に SDCC をインストールします。このサンプルで使用しているのは、sdcc-2.8.0-setup.exe です。PATH を通すことを必ず行ってください。

C コンパイラ SDCC のインストールとポケコン PC-E200 での動作確認
http://noritsugu.at.webry.info/200707/article_1.html

2.2 スタートアップルーチン（アセンブラ）

スタートアップルーチン crt0.s を作成します。crt0.s のオリジナルは、SDCC をインストールすると、下記にあります。

C:¥ProgramFiles¥SDCC¥lib¥src¥z80¥**crt0.s**

これを下記を参考に改造します。
T80-ソフトウエア
http://hijiri.yitc.go.jp/mb/index.php?T80%2F%A5%BD%A5%D5%A5%C8%A5%A6%A5%A8%A5%A2

1 下記3行をコメントアウトします。

```
call    gsinit
.area   _GSINIT
gsinit::
```

2 スタックポインタの初期化は、このサンプルのメモリ容量が 2Kbyte なので、

```
ld      sp,#0x7ff
```

とします。

2.3 C プログラム

IO ポート SW から読んだ値に 0～3 を加算して、IO ポート LED に出力するプログラムです。

＜t80_test.c の内容＞

```
sfr at 0x00 SW;
sfr at 0xFF LED;

void
main()
{
    int i;

    while(1) {
        for (i=0; i <= 3; i++) {
            LED = SW + i;
        }
    }
}
```

2.4 コンパイル

crt0.s のアセンブル、t80_test.c のコンパイルと crt0.o のリンク、および hex ファイルの作成を行うバッチファイルです。sdcc-z80.bat をダブルクリックすると、実行できます。

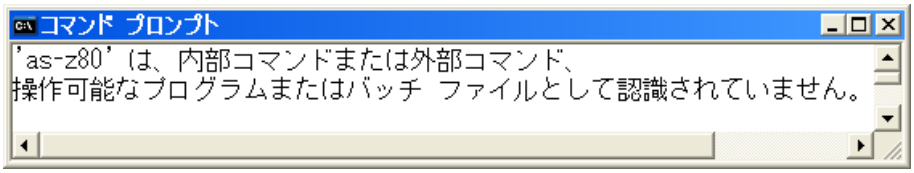
crt0.s に変更がなく、crt0.s のアセンブルを行わないときは、rem をつけてコメント化することができます。

```
rem as-z80 -logs crt0.o crt0.s
```

＜sdcc-z80.bat の内容＞

```
as-z80 -logs crt0.o crt0.s
sdcc -mz80 t80_test.c crt0.o --no-std-crt0
packihx t80_test.ihx > t80_test.hex
```

下記のようなエラーが出る場合は、「2.1 SDCC のインストール」で PATH を通しているかどうか確認してください。



2.5 hex ファイルのコピー

t80_test.hex を C:¥work¥T80_sample¥T80_io フォルダにコピーします（上書き）。

3.1 T80 ソースの収集

必要なファイルは、サンプルファイルを解凍してできた C:¥work¥T80_sample ¥T80_io フォルダに既に入っています。

T80 の VHDL ソースは下記にあります。ダウンロードには会員登録が必要です。

OPENCORS の T80 のページ

http://www.opencores.org/projects.cgi/web/t80/overview

オリジナルを使いたいところですが、データバス D[7..0]が双方向なので使いにくく、下記にデータバスを分離した、T80as.vhd がありますので、使わせていただきます。

FPGA-Donkey Kong

http://office-dsan.hp.infoseek.co.jp/dkong/dkong_prj.htm

altera_fpga_dkong_v301.zip をダウンロードし、解凍します。¥dkong_prj¥t80_ip フォルダ内の下記ファイルを使います。

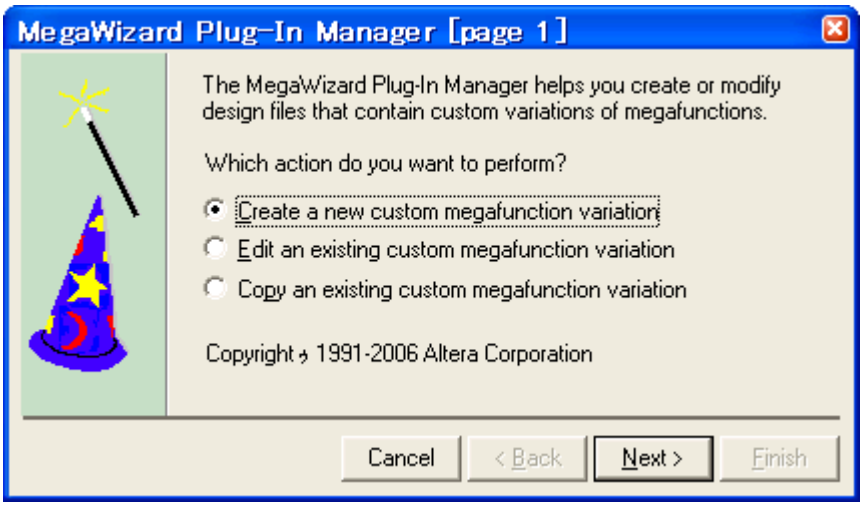
3.2 周辺回路の作成

メモリや PLL などが必要となりますが、サンプルファイルを解凍してできた C:¥work¥T80_sample¥T80_io フォルダに既に入っています。

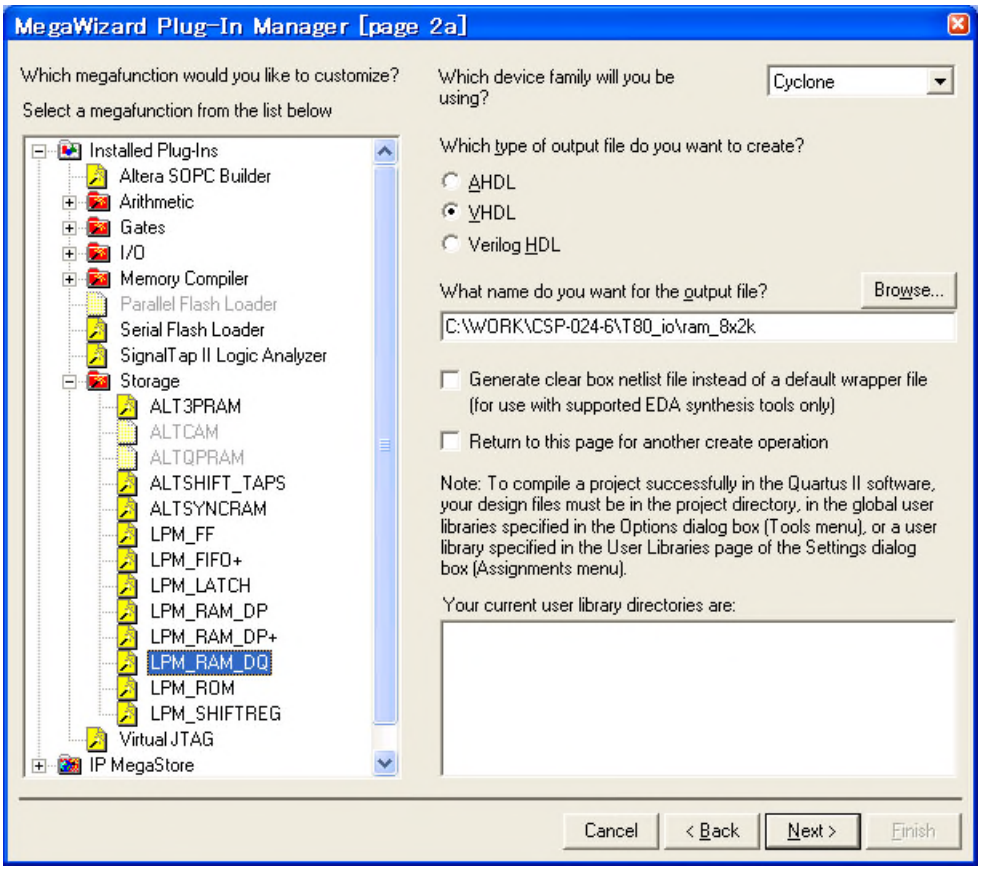
1 2Kbyte のメモリを作成します。

(1) メニューの[Tools | MegaWizard Plug-In Manager]を選びます。

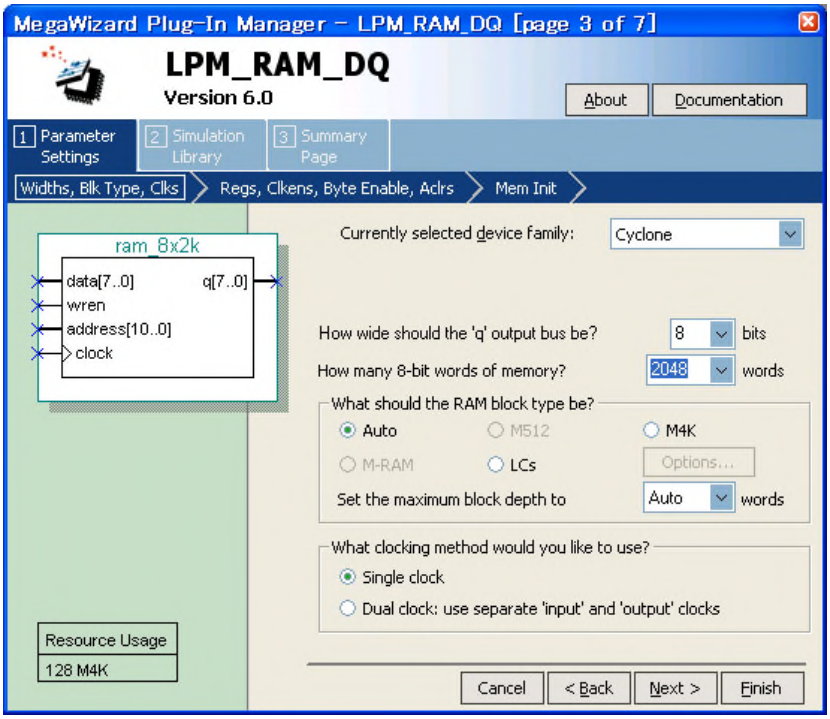
(2) Next を押します。



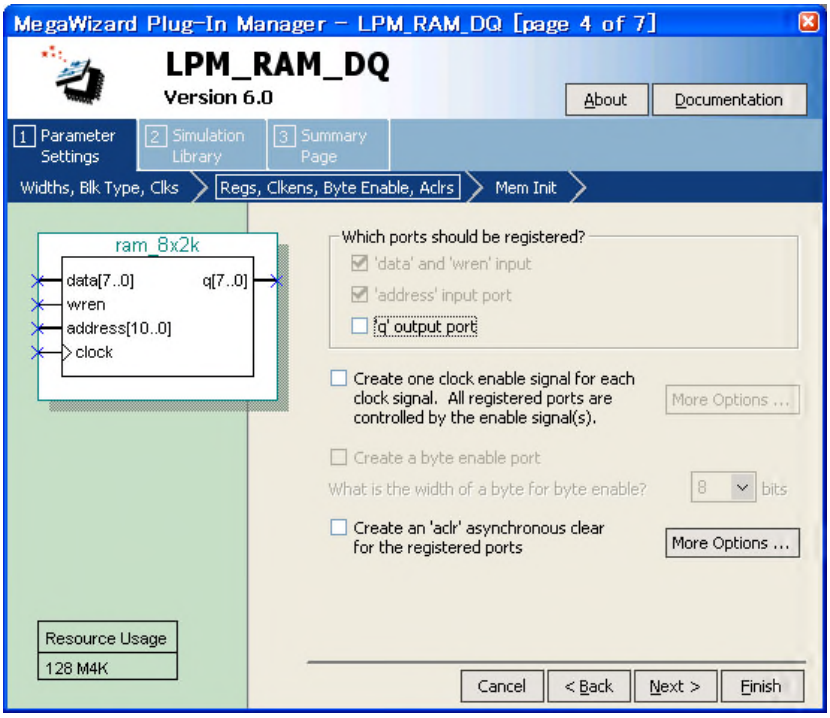
(3) LPM_RAM_DQ をクリックし、ファイル名欄に ram_8x2k と記入します。



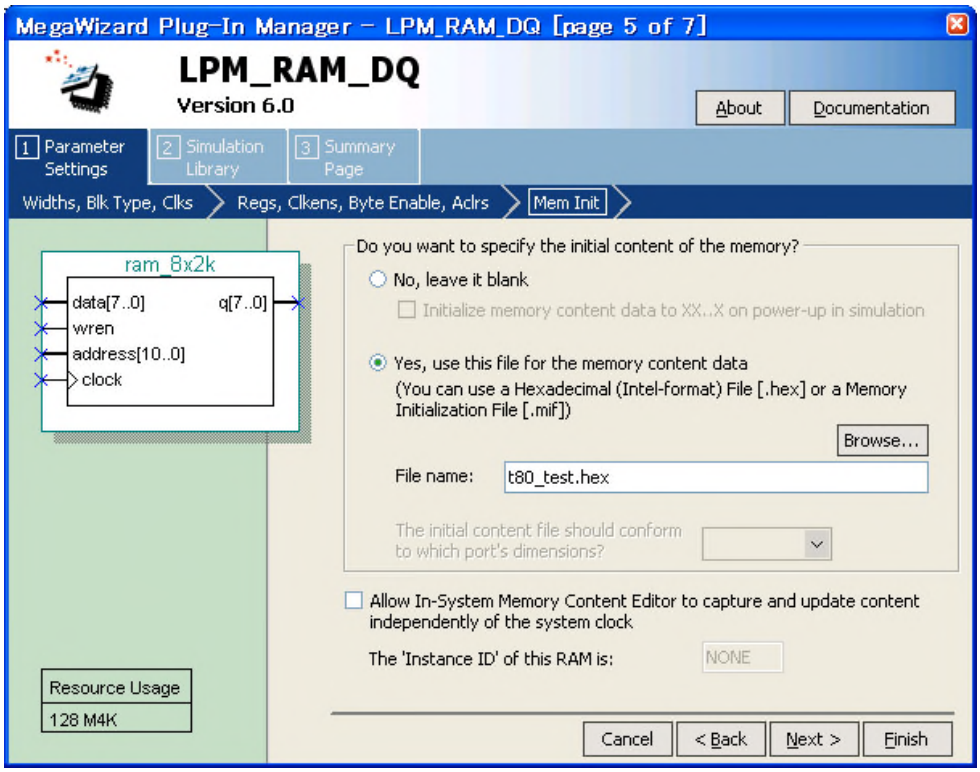
（4） 8 bits、2048 words を選びます。



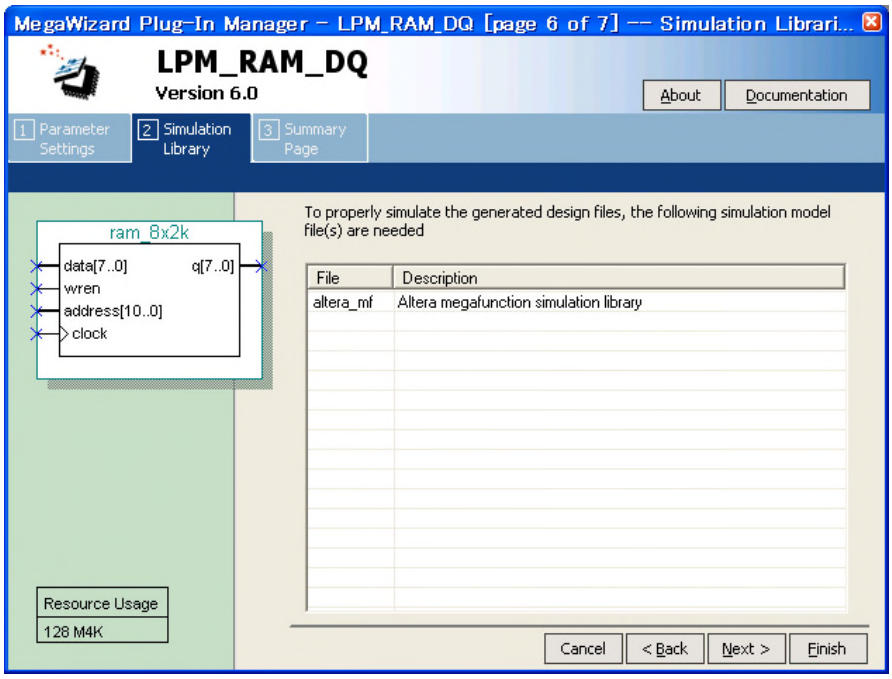
（5） 'q' output port のチェックを外します。



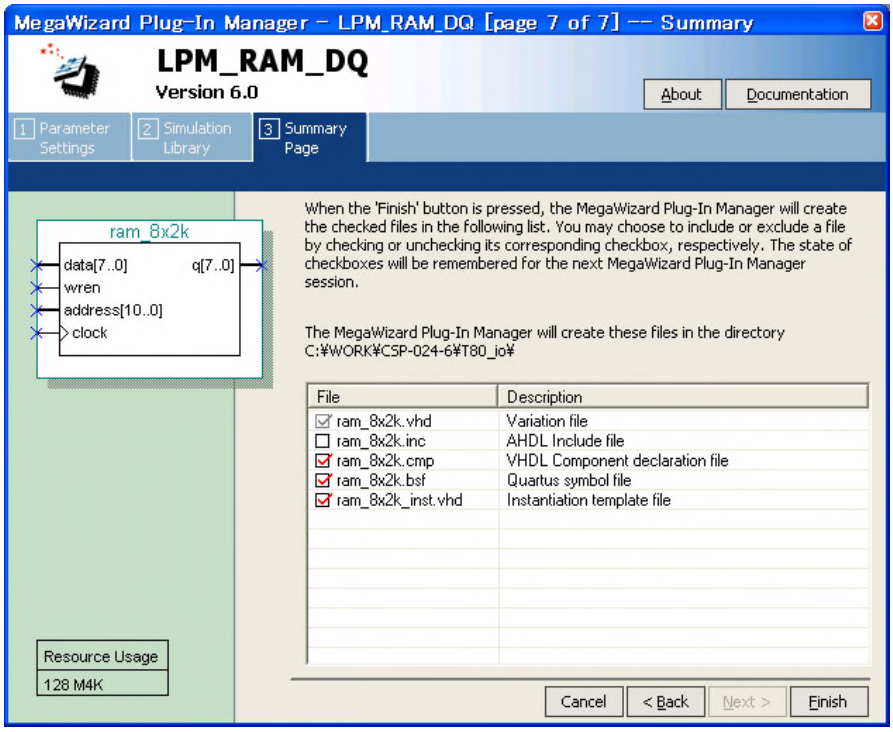
(6) File name の欄に t80_test.hex と記入します。



(7) Next を押します。



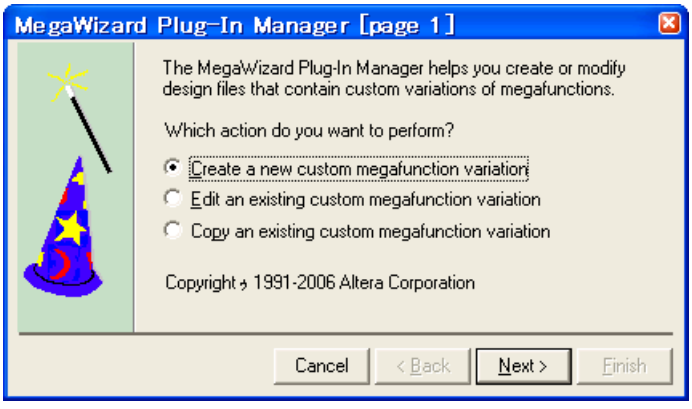
（8） Finish を押します。



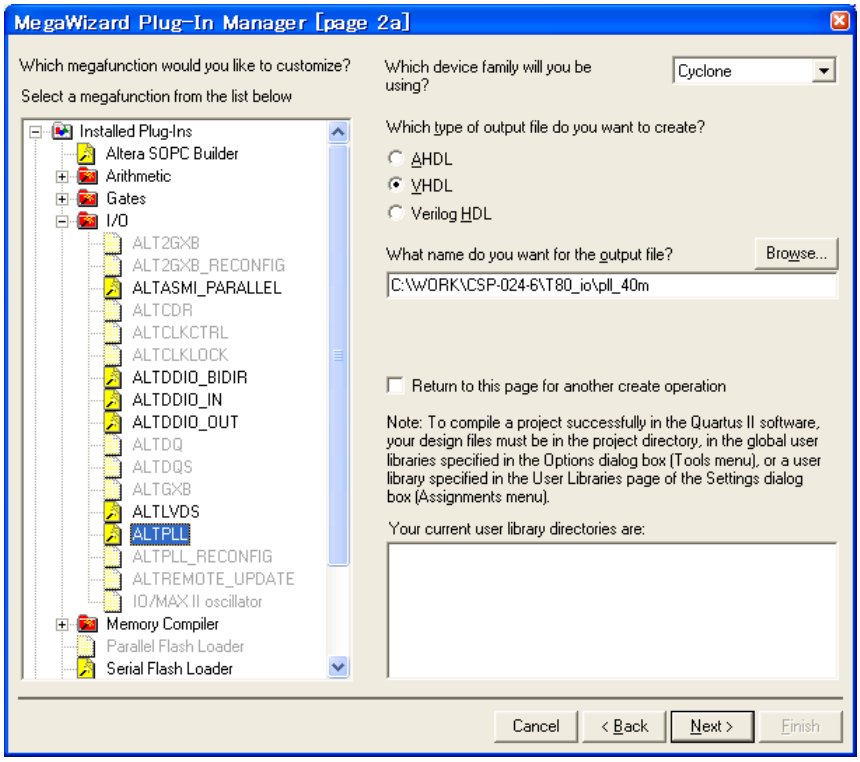
2 FPGA 基板のクロックが 50MHz ですので、PLL を使って 40MHz に落とします。今回使用するデバイス Cyclone EP1C6Q240C8 に T80 を入れた場合、最高動作周波数が 40 数 MHz のためです。

（1） メニューの[Tools | MegaWizard Plug-In Manager]を選びます。

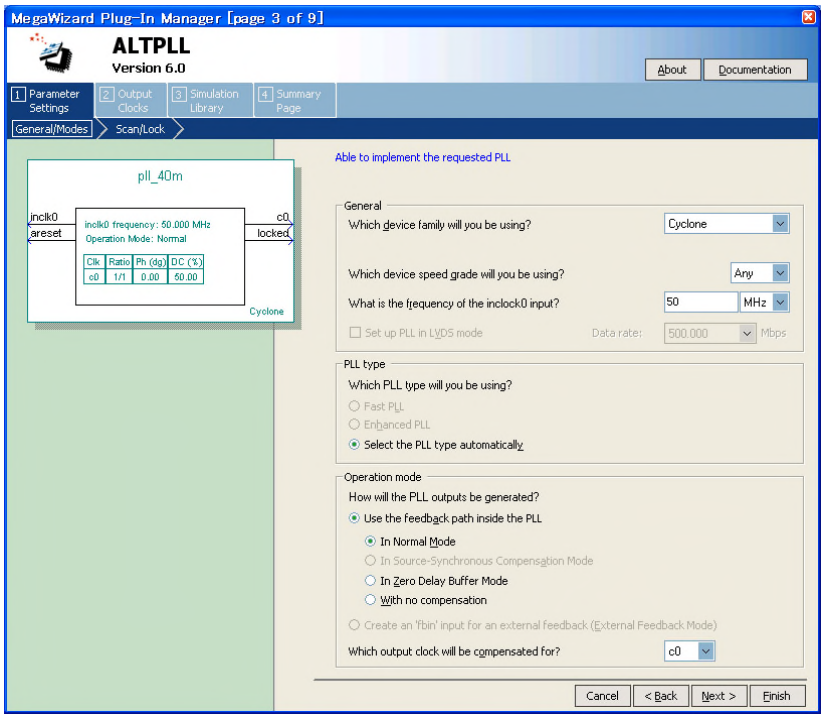
（2） Next を押します。



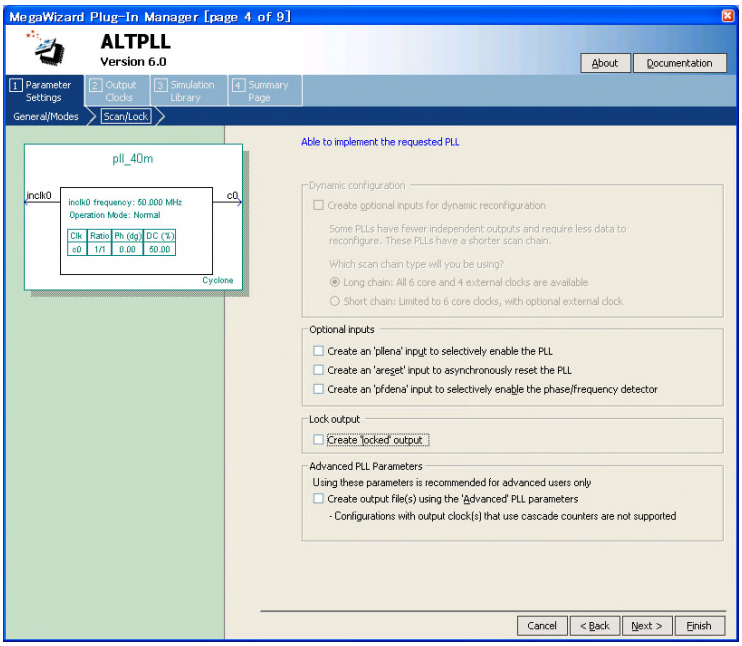
（3） ALTPLL をクリックし、ファイル名欄に pll_40m と記入します。



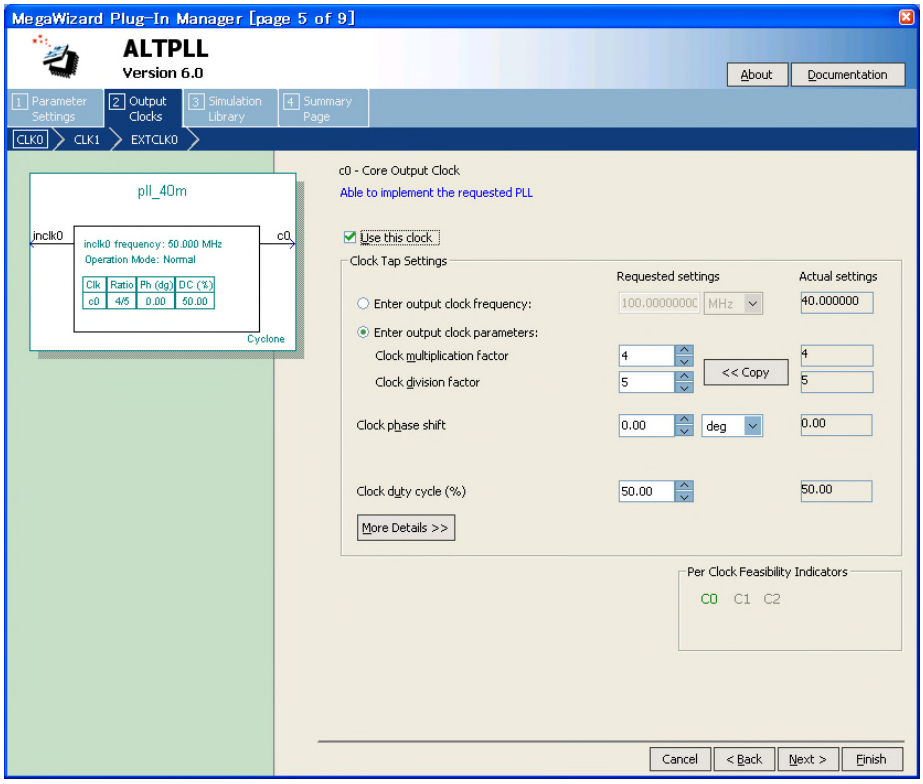
（4） 50MHzと記入します。



（5） Next を押します。



（6） Enter output clock parameters に 4 と 5 を記入します。Finish を押し、完成です。



3.3 Altera 社の開発ソフト Quartus Ⅱ の操作

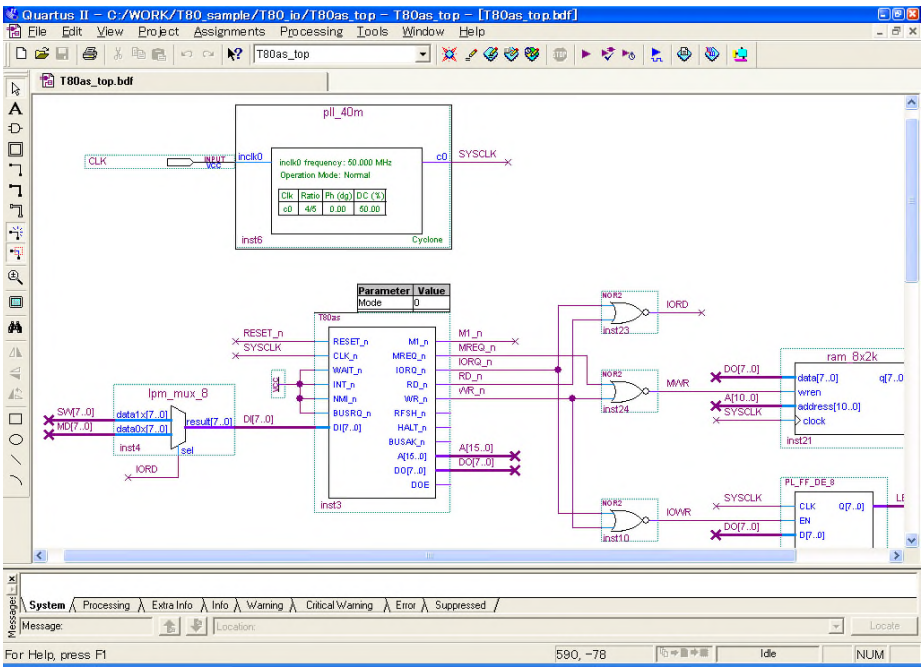
このマニュアルでは、Quartus Ⅱ Ver6.0 を使用しています。

1 ポケットロジアナソフト Ver5.50 をインストールしたパソコンの C:¥ProgramFiles¥PocketLogiana¥Logiana IP¥Sample¥quartus¥cyclone¥ana32x1k compress gen16x256 フォルダ内の plogi.bdf と plogi.qpf を除くファイル全てを C:¥work¥T80_sample¥T80_io にコピーします。

2 Quartus Ⅱを起動します。

3 メニューの［File ｜ Open Project］を選択し、プロジェクトファイル C:¥work¥T80_sample¥T80_io¥ T80as_top.qpf を開きます。

4 メニューの［File ｜ Open］を選択し、T80as_top.bdfを開きます。



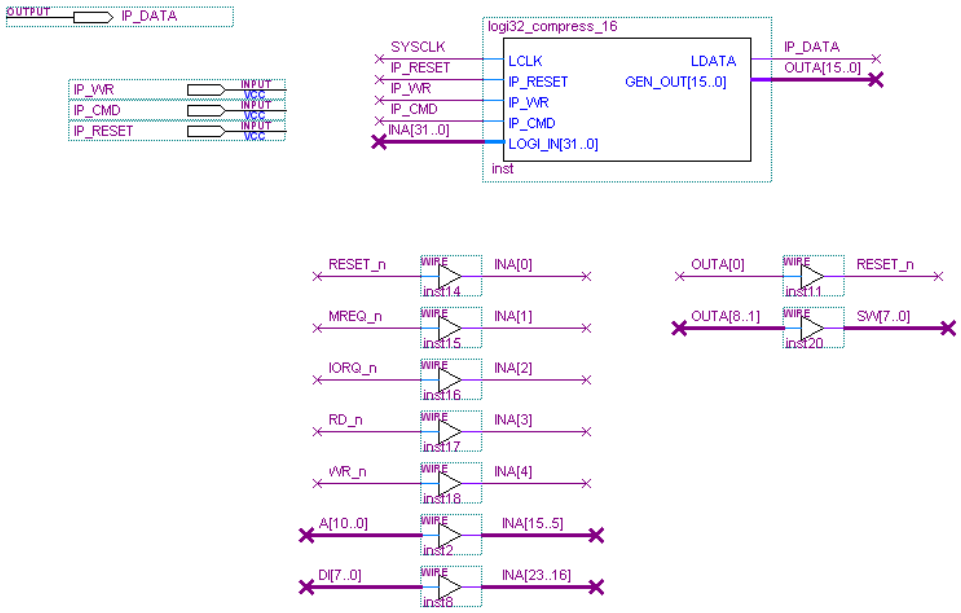
＜回路図の解説＞

（1） pll_40m は 50MHz クロックを 40MHz に変換します。今回使用するデバイス Cyclone EP1C6Q240C8 に T80 を入れた場合、最高動作周波数が 40 数 MHz のためです。

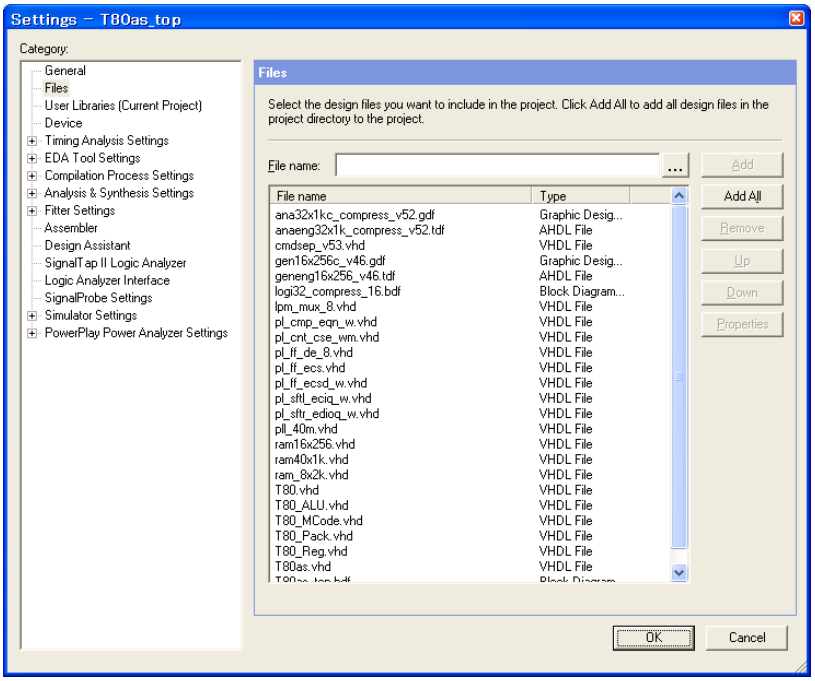
（2） T80 の RESET_n は、パターンジェネレータ IP の出力 OUTA[0]で制御します。FPGA のコンフィグレーション直後、パターンジェネレータ IP の出力は全て L です。OUTA[0]をポケットロジアナソフトの操作で H にすると、リセットが解除され、T80 の動作が始まります。操作方法は、「3.4 ポケットロジアナソフトによる波形測定」で解説しています。

（3） IO ポート SW[7..0]の値を読んで、0～3 を加算した値を、IO ポート LED[7..0]に出力します。
SW[7..0]は DIP スイッチなどを想定していますが、このサンプルではパターンジェネレータ IP の出力 OUTA[8..1]を接続しています。
LED[7..0]は LED の点灯などを想定していますが、このサンプルではロジックアナライザ IP で波形を測定しています。

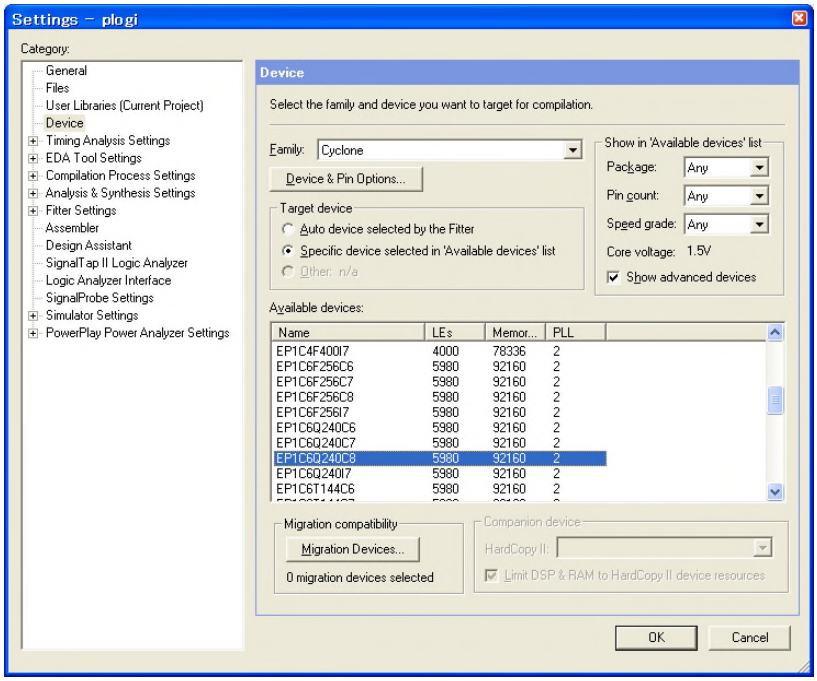
（4） T80 とパターンジェネレータ IP、ロジックアナライザ IP の接続に、WIRE シンボルを使っています。WIRE は名称の異なる信号を接続するのに使います。



5 メニューの[Project | Add/Remove Files in Project]を選びます。Add All ボタンを押して、ファイルをプロジェクトに加えます。



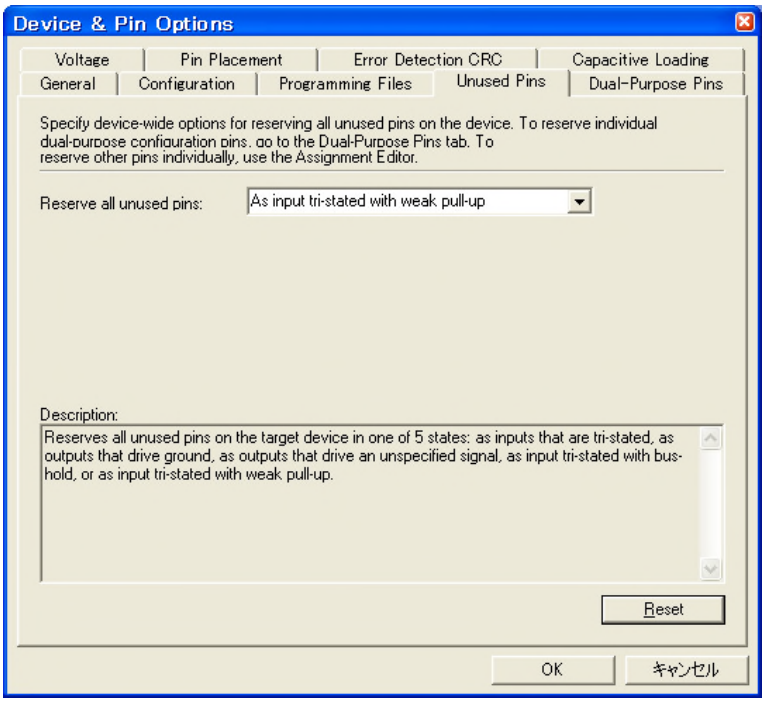
6 メニューの[Assignment | Device]でデバイスを選択します。



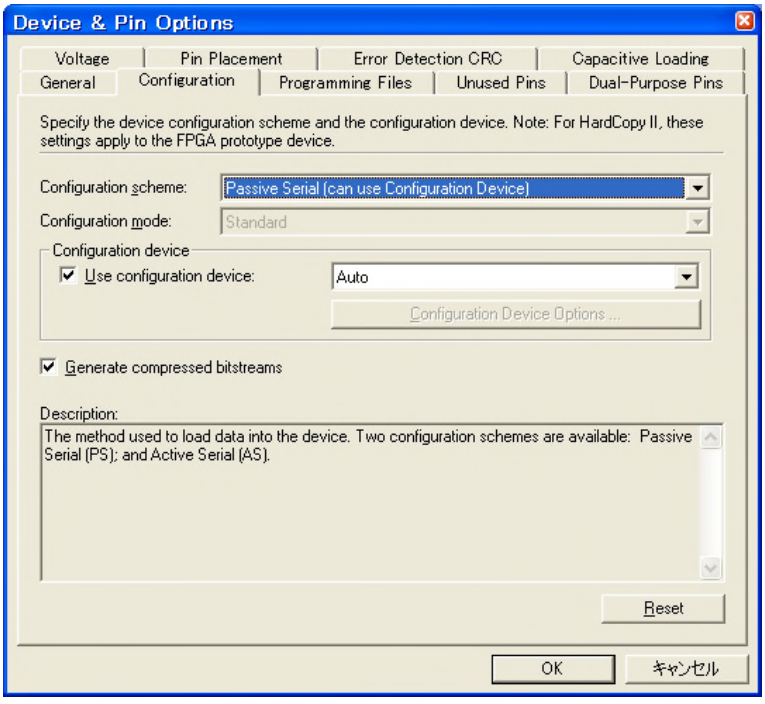
7 メニューの[Assignments | Device]で開いたダイアログの中の

Device & Pin Options ボタンを押します。

(1) Unused Pins タブで、As input try-stated with weak pull-up を選びます。

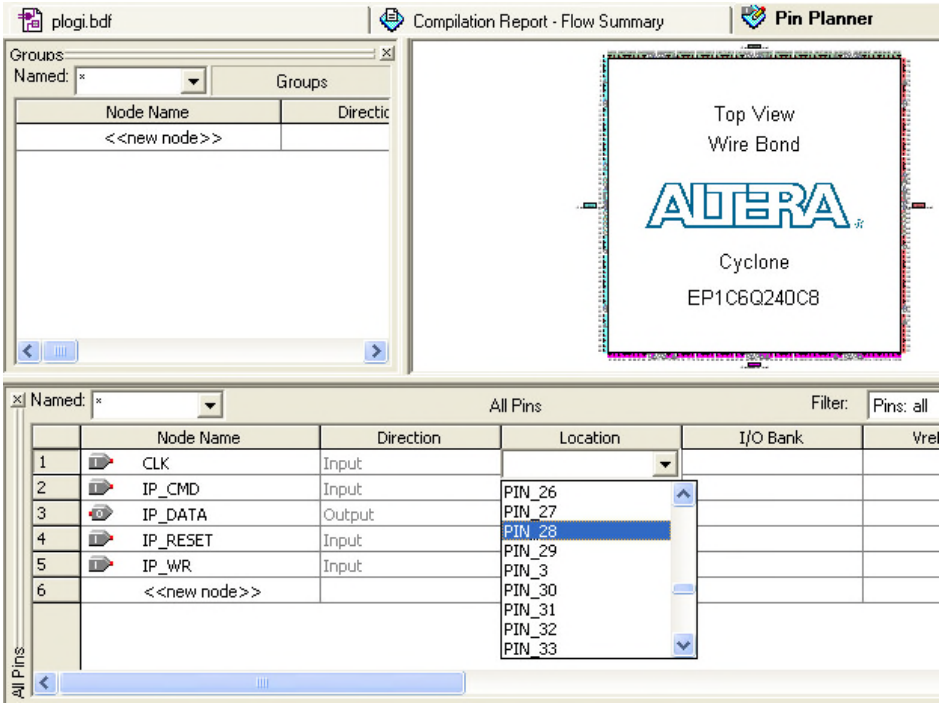


(2) Configuration タブで、Passive Serial を選びます。

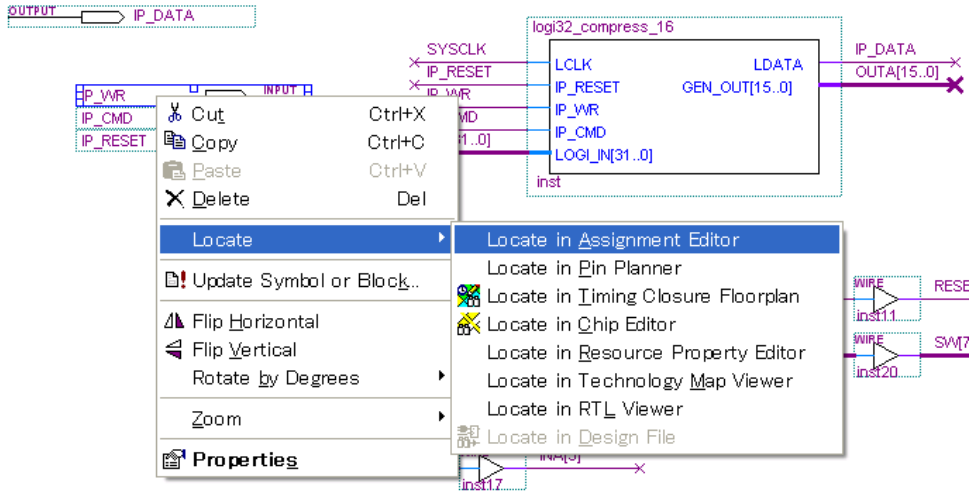


8 メニューの[Processing | Start Compilation]でコンパイルを行います。

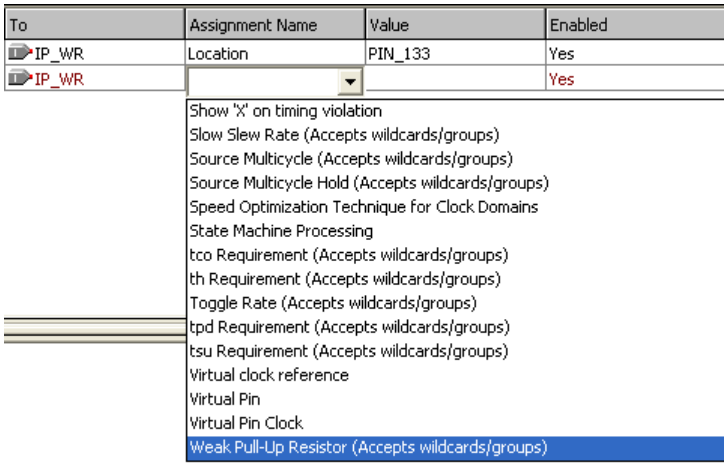
9 メニューの[Assignments | Pins]で、入力ピン・出力ピンの割当を行います。
FPGA 基板に合わせて設定してください。Location のセルをダブルクリックして、ピン番号を選択します。



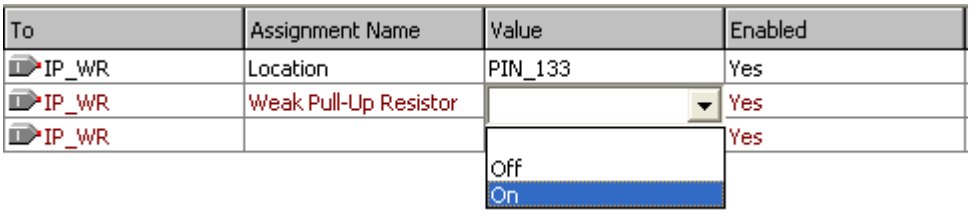
10 入力ピンをWeak Pull-upします。ポケットロジアナのフラットケーブルをFPGA基板に接続していないとき、入力にノイズが乗るのを防ぎます。入力ピン IP_WR の上で右クリックして、Locate in Assignment Editor を選びます。



Assignment Name のセルをダブルクリックして、Weak Pull-Up Registor を選びます。

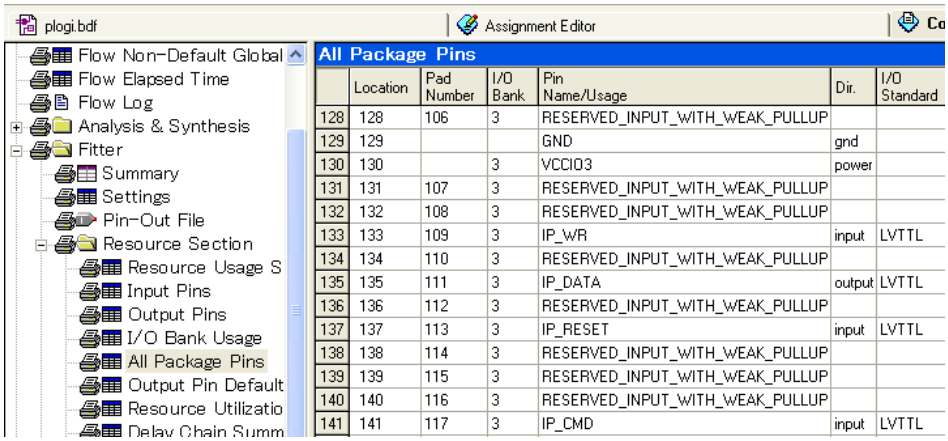


Value のセルをダブルクリックして、On を選びます。

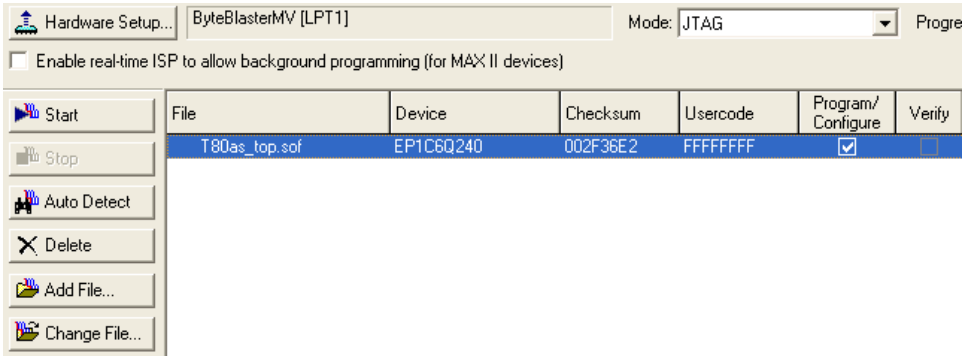


同様にして入力ピン IP_CMD、IP_RESET についても、Weak Pull-up します。メニューの[File | Save]で設定を保存します。

11 メニューの[Processing | Start Compilation]で、再度コンパイルを行います。
ピンの設定状態は、Compilation Report の All Package Pins で分かります。



12 メニューの[Tools | Programmer]を選び、回路をダウンロードします。
ダウンロードケーブルを認識していない場合、Hardware Setup ボタンを押して、認識させます。Program/Configure にチェックをして、Start ボタンを押します。

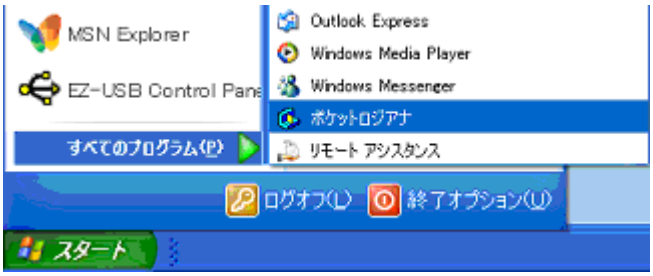


メニューの[File | Save]で、Programmer の設定を保存します。

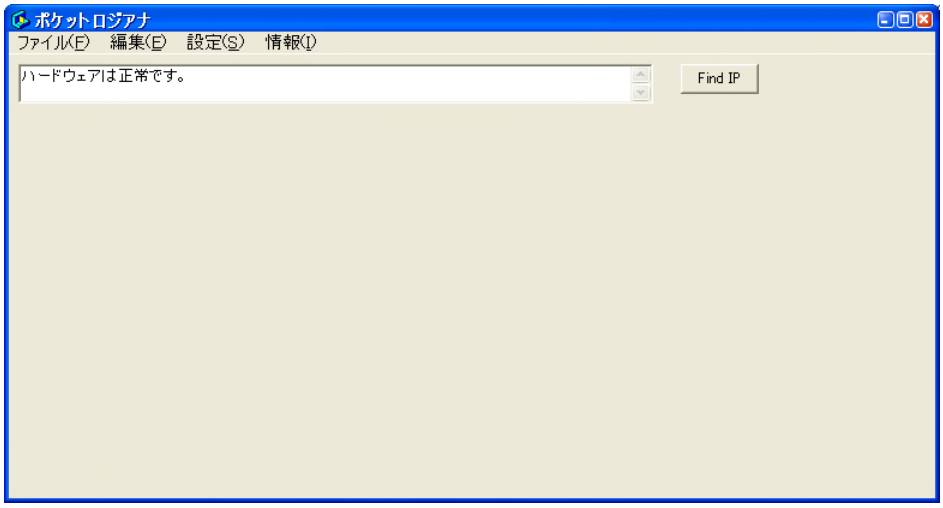
3.4 ポケットロジアナソフトによる波形測定

ポケットロジアナと FPGA 基板がフラットケーブルで接続されているものとします。

1 スタートメニューからポケットロジアナを選び、ソフトを立ち上げます。

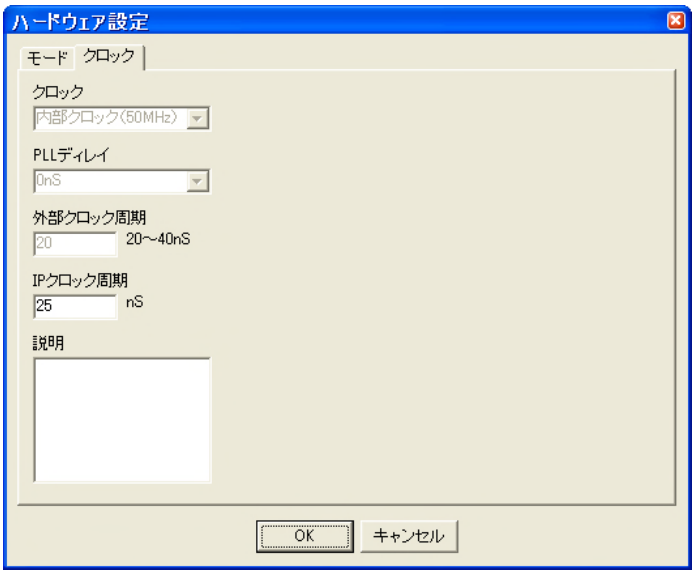


2 メッセージボックスに“ハードウェアは正常です”と表示されます。

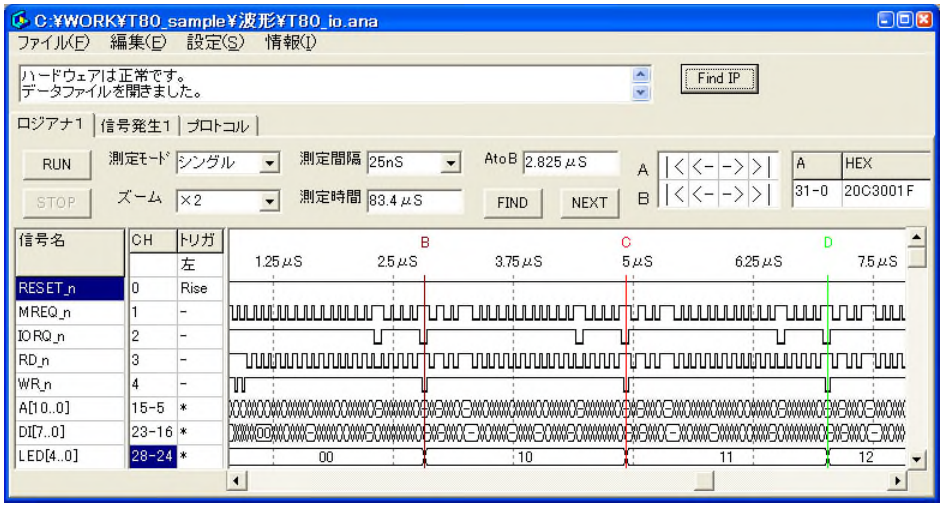


3 Find IP ボタンを押し、5 秒程待ちます。ロジアナ IP と信号発生 IP が見つかり、下記のような画面になります。

4 メニューの[設定｜ハードウェア]を選択し、「ハードウェア設定」ダイアログを表示します。「クロック」タブをクリックし、IP クロック周期を 25 に変更してください。25nS=40MHz



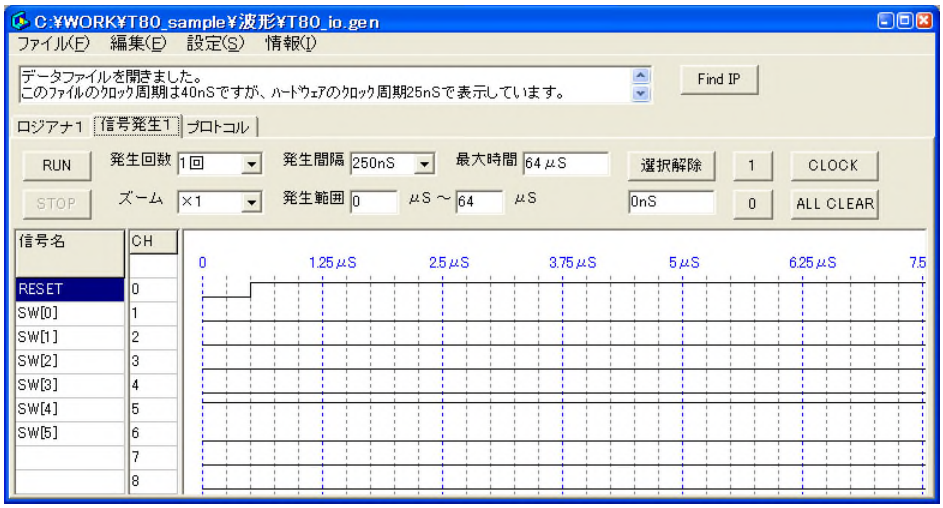
5 メニューの[ファイル｜開く]を選択し、C:¥work¥T80_sample¥波形¥T80_io.ana を開きます。



6 RUN を押します。測定が開始され、メッセージボックスに“ロジアナ 1 測定中です。”と表示されます。

6 信号発生 1 タブをクリックし、ページを切り換えます。

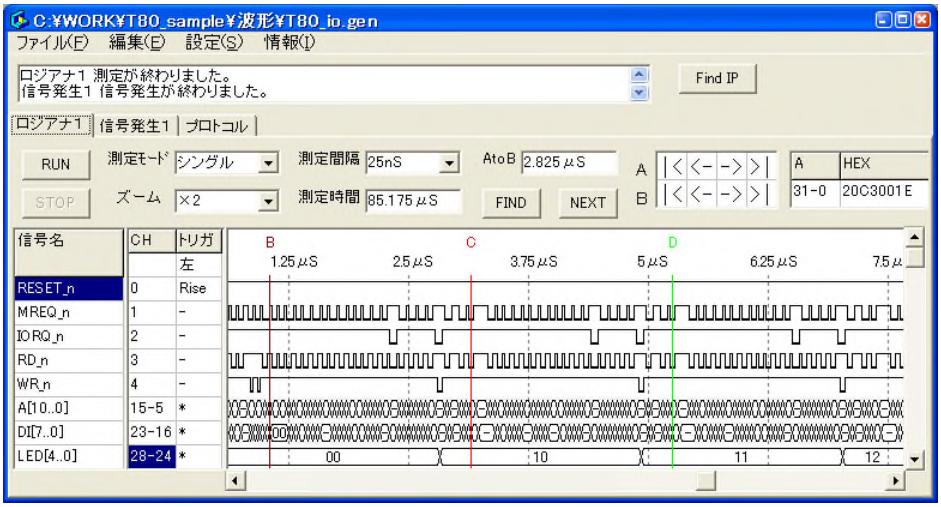
メニューの[ファイル | 開く]を選択し、C:¥work¥T80_sample¥波形¥ T80_io.gen を開くと、信号発生用の波形が表示されます。



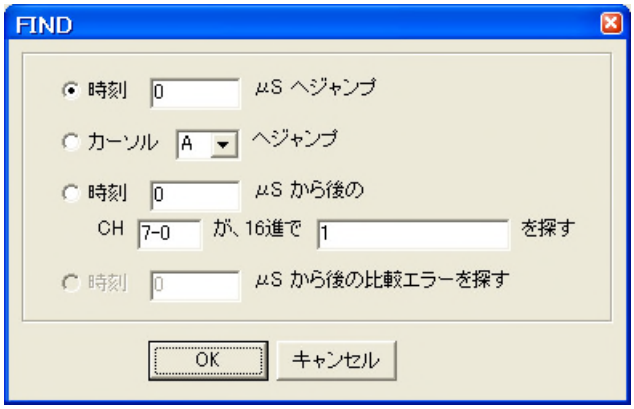
7 RUN を押します。メッセージボックスに“ロジアナ 1 トリガを検出しました。”,“ロジアナ 1 データ転送中です。”と表示されます。

信号発生, 測定が終了すると、“ロジアナ 1 測定が終わりました。”, “信号発生 1 信号発生が終わりました。”と表示されます。

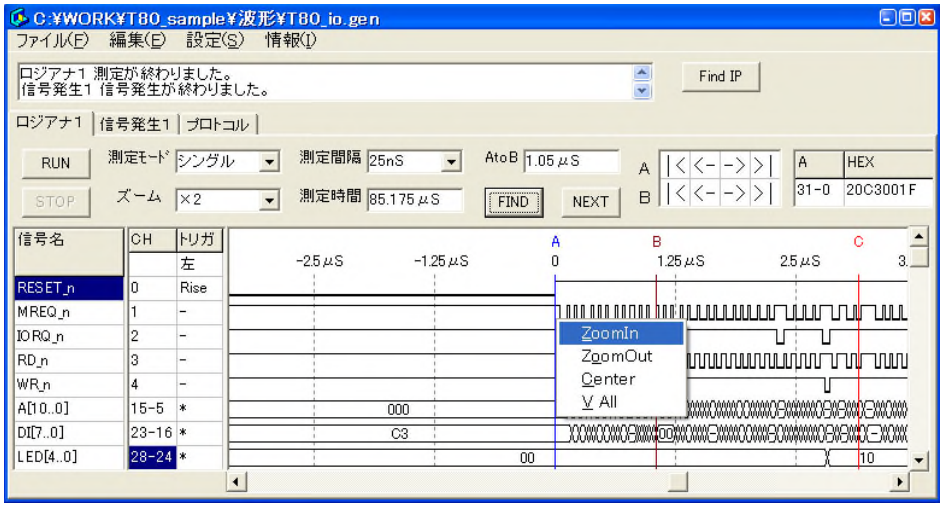
8 ロジアナ 1 タブをクリックし、ページを切り換えると、測定した波形が表示されています。



9 RESET 解除後の動作を見るには、Find ボタンを押し、時刻 0uS へジャンプします。



波形画面上で右クリックし、Zoom In を選ぶと、波形を拡大することができます。



信号 M1_n の立上り時の DI[7..0]の値を読むと、T80 が実行するのオペコードになっています。