



不可能への挑戦

株式会社日昇テクノロジー

低価格、高品質が不可能？  
日昇テクノロジーなら可能にする

## Modelsim 入門

株式会社日昇テクノロジー

<http://www.csun.co.jp>

[info@csun.co.jp](mailto:info@csun.co.jp)

作成日 : 2012/8/13

copyright@2012



## ・ 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2012/8/13

※ この文書の情報は、文書を改善するため、事前の通知なく変更されることがあります。最新版は弊社ホームページからご参照ください。

「<http://www.csun.co.jp>」

※ (株)日昇テクノロジーの書面による許可のない複製は、いかなる形態においても厳重に禁じられています。

## Modelsim SE 10.0.c 入門

弊社のCycloneIVボードに使われるQuartus 11とModelsim10.0cを利用して、Modelsim SE 10.0.cの使い方を簡単に説明します。

Quartus10以降のバージョンは波形シミュレーションソフトウェアを含まれません、シミュレーションを行いたい場合、他のツール、例えば、Modelsim、インストールが必要です。

では、Quartus 11にはModelsim 10.0cの呼び出し方及びtestbenchの書き方を説明します。

Modelsim 10.0c評価版ダウンロードURL :

32bit:

<http://www.dragonwake.com/download/FPGA/EP4CE15/modelsim-win32-10.0c-se.exe>

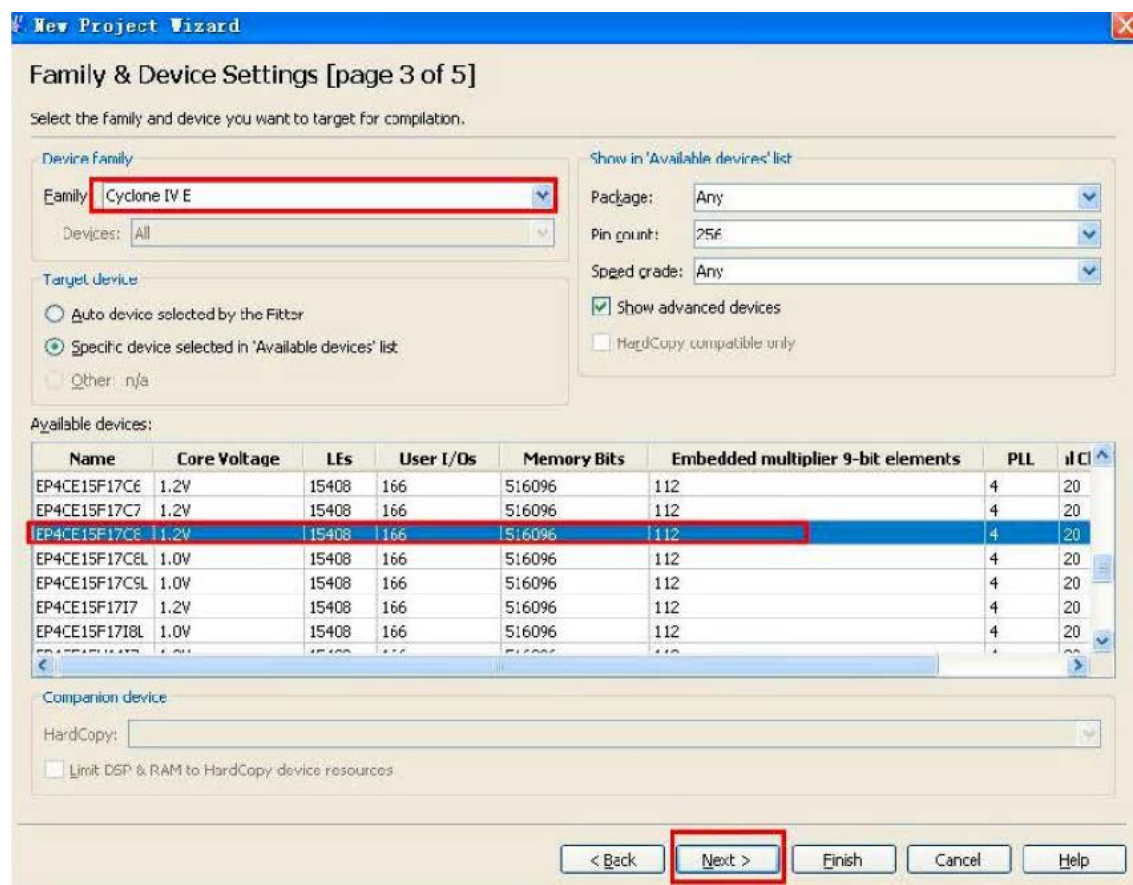
64bit:

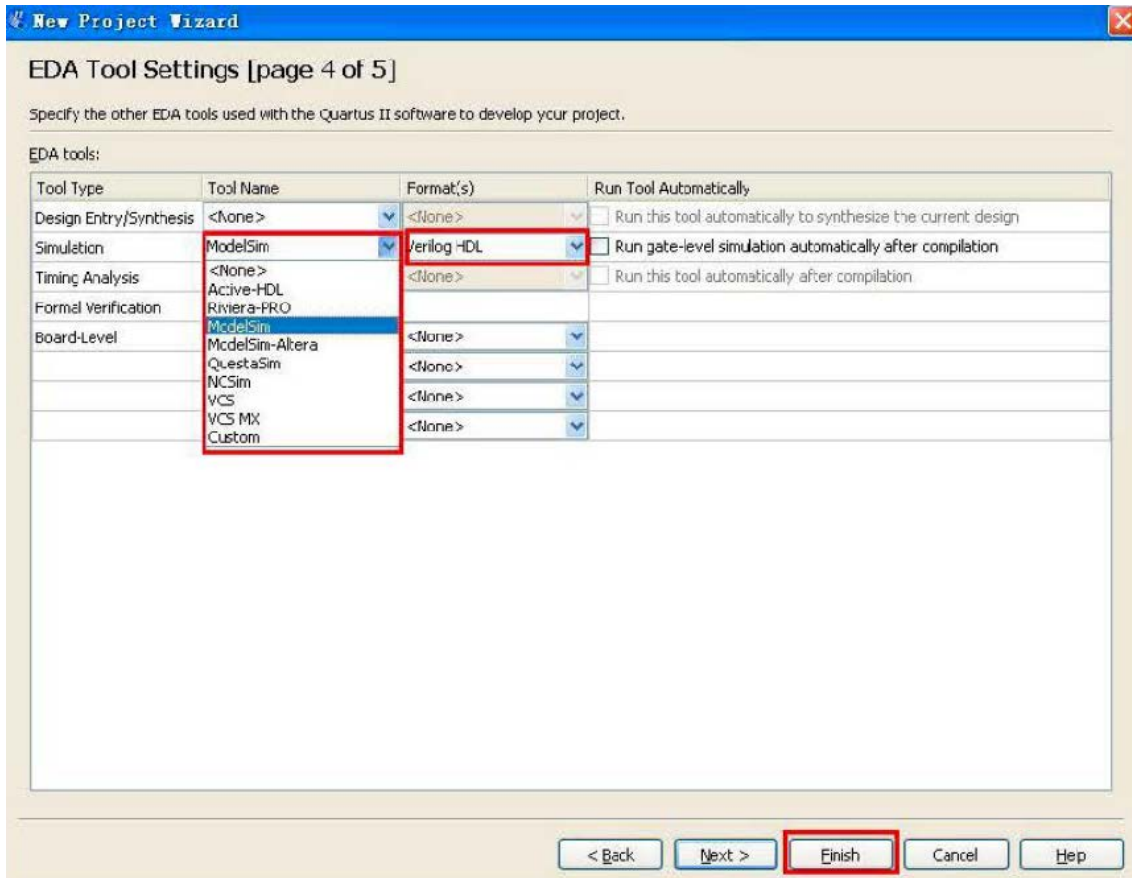
<http://www.dragonwake.com/download/FPGA/EP4CE15/modelsim-win64-10.0c-se.exe>

カウンタを例として説明します。

前提 : Quartus 11とModelsim 10.0cをインストール済

1. プロジェクトを新規時に以下通り設定





シミュレーションソフトウェアと言語により選択：

Tool Name : Modelsim

Format : Verilog HDL

カウンタプログラム「count128.v」を以下通り作成します。

```
module count128(clk,rst_n,divclk,data);
```

```
Input clk,rst_n;
```

```
output divclk;
```

```
output [7:0] data;
```

```
reg[7:0] data;
```

```
assign divclk:data[7].
```

```
always@(posedge clk or negedge rst_n)
```

```
if(!rst_n)
```

```
begin
```

```
data<=8'h00;
```

```
end
```

```
else
```

```
begin
```

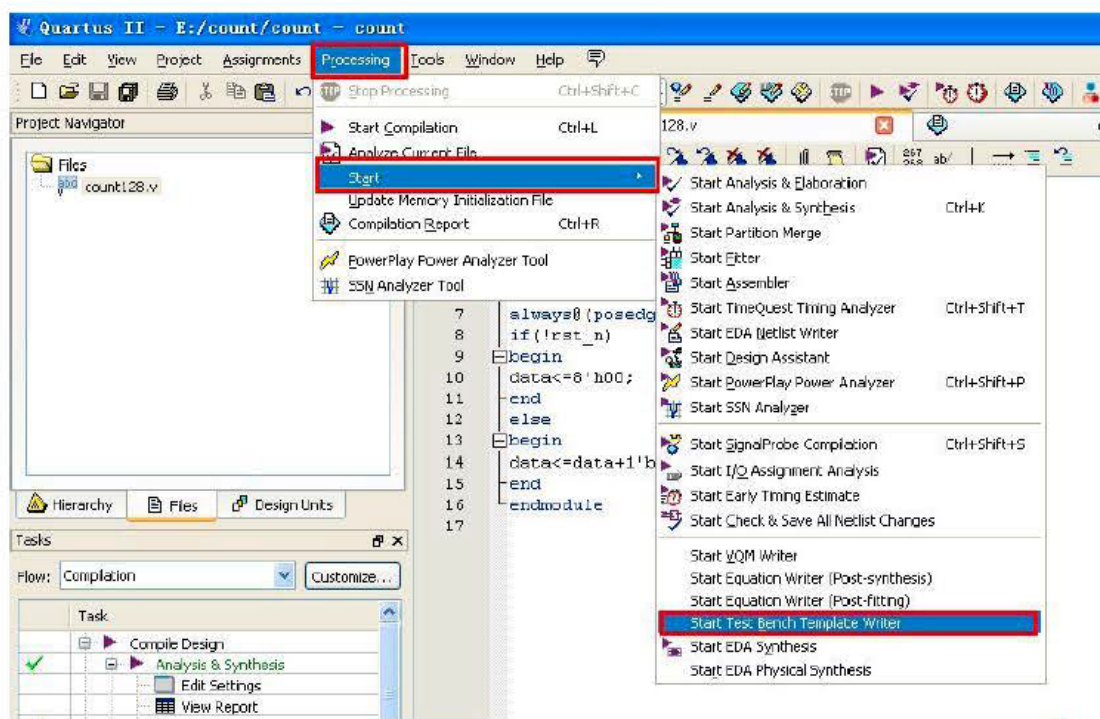
```
data<=data+1' b1;
```

```
end
```

```
endmodule
```

プログラムで周波数を128に分ける機能を実現

シミュレーションテスト用ファイルtestbenchの作成方法は下記です。



ここまでtestbenchのテンプレートを作成されました、生成されたファイルは.vtです、プロジェクトフォルダーの「simulation¥modelsim」であります。





テンプレート内容：

```
'timescale 1ps/1ps
module count128_vlg_tst();
//constants
//general purpose registers
reg eachvec;
//test vector input register
reg clk;
reg rst_n;
//wires
wire[7:0] data;
wire divclk;

//assign statements {if any}
count128 il{
//port map-connection between master ports and signals/registers
    .clk{clk},
    .data{data},
    .divclk{divclk},
    .rst_n{rst_n}
};
initial
begin
//code thata executes only once
//Insert code here->begin

//->end
$display{ "Running testbench" };
end
always
//optional sensitivity list
//@{event1 or event2 or ... eventn}
begin
//code executes for ervery event on sensitivity list
//insert code here->begin
```



```
@eachvec;
```

```
//->end
```

```
end
```

```
endmodule
```

メモ：

a) 全て入力の変数がreg型、出力がwire型

b) for, whileのような形式があっても良い

count128.vファイル内容により、testbenchは下記のように修正：

```
timescale 1ps/1ps
module count128_vlg_tst();
//constants
//general purpose registers
reg eachvec;
//test vector input register
reg clk;
reg rst_n;
//wires
wire[7:0] data;
wire divclk;

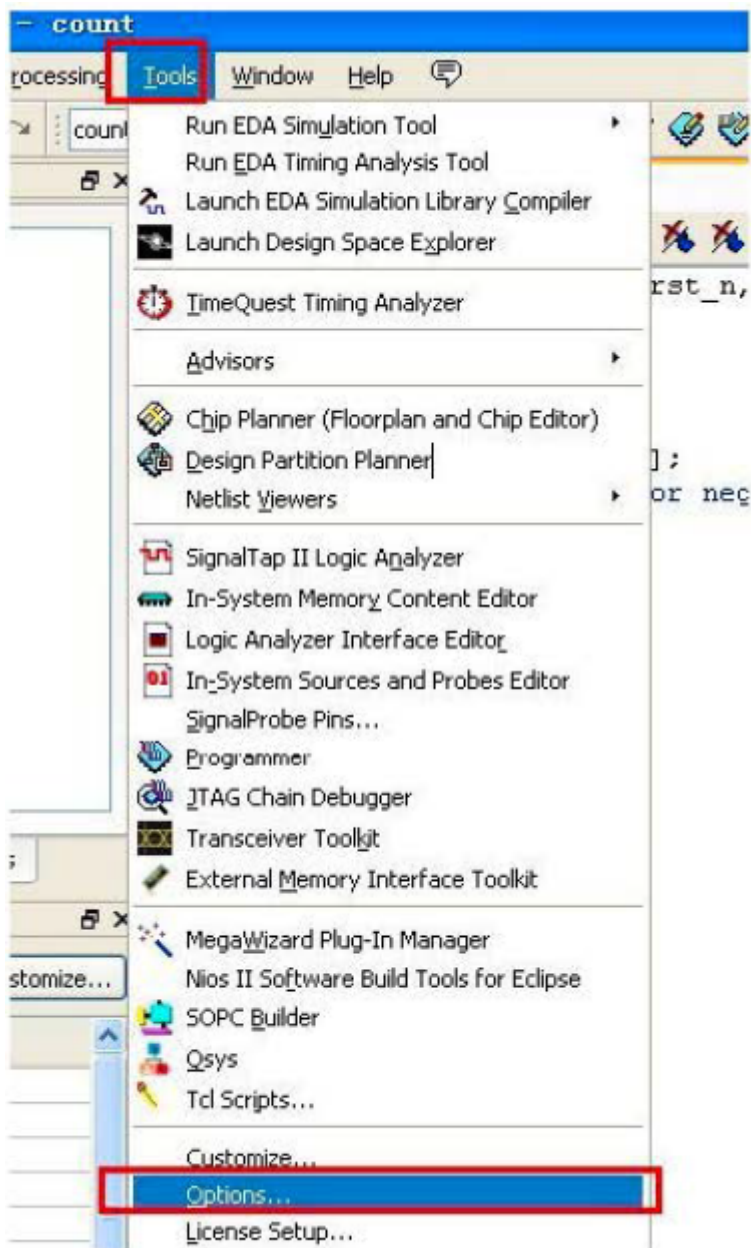
//assign statements {if any}
count128 il{
//port map-connection between master ports and signals/registers
.clk{clk},
.data{data},
.divclk{divclk},
.rst_n{rst_n}
};
initial
begin
clk=1;
rst_n=0;
#50 rst_n=1;
#100000 $finish
end
```

```
always #2 clk=~clk;  
endmodule
```

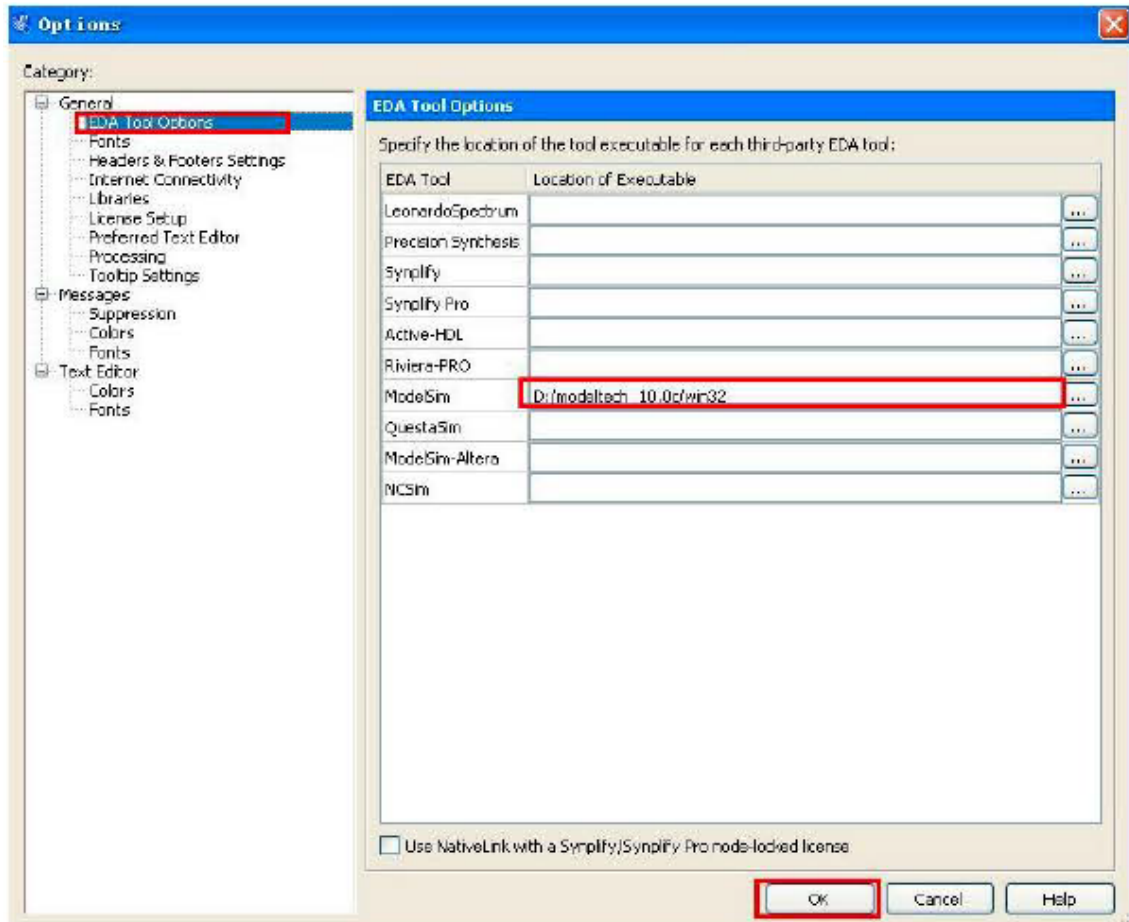
変数クロックclkとrst\_nだけに値を設定してもよい

次はプロジェクトの設定を行う

Quartusからmodelsimを呼び出し、以下のように設定

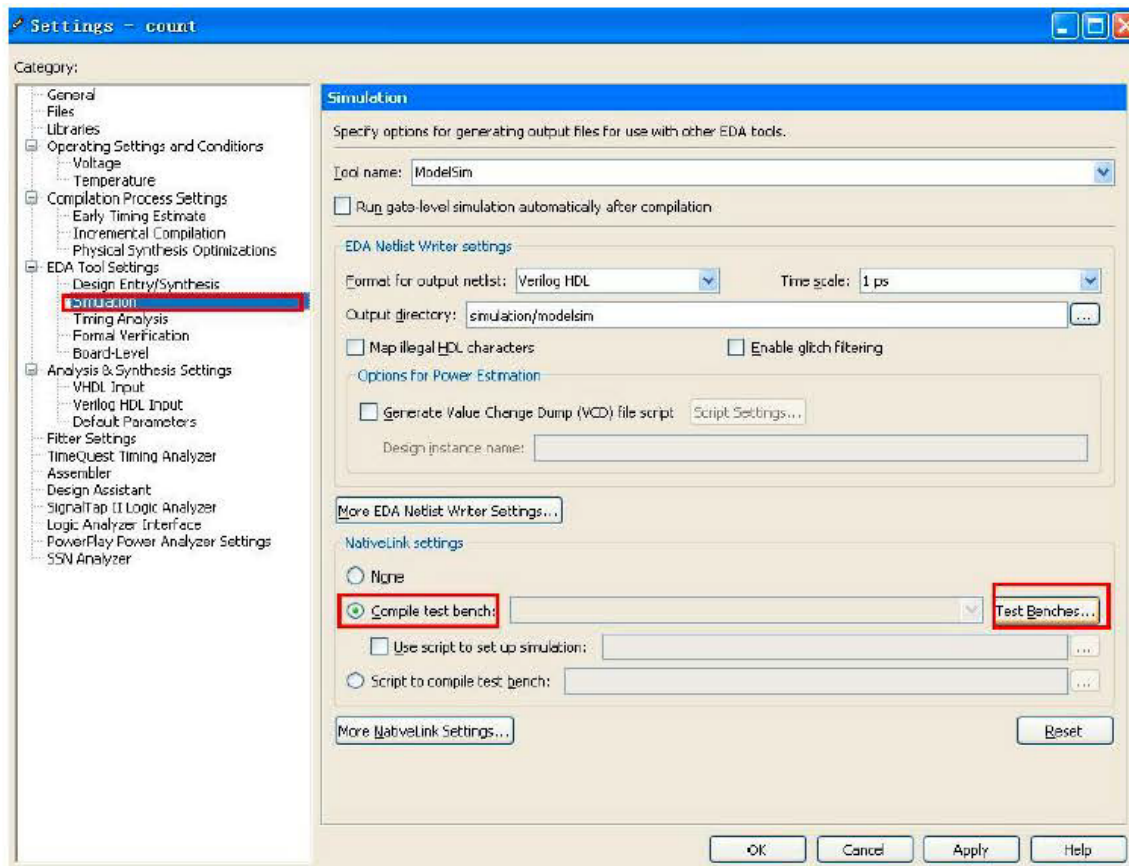
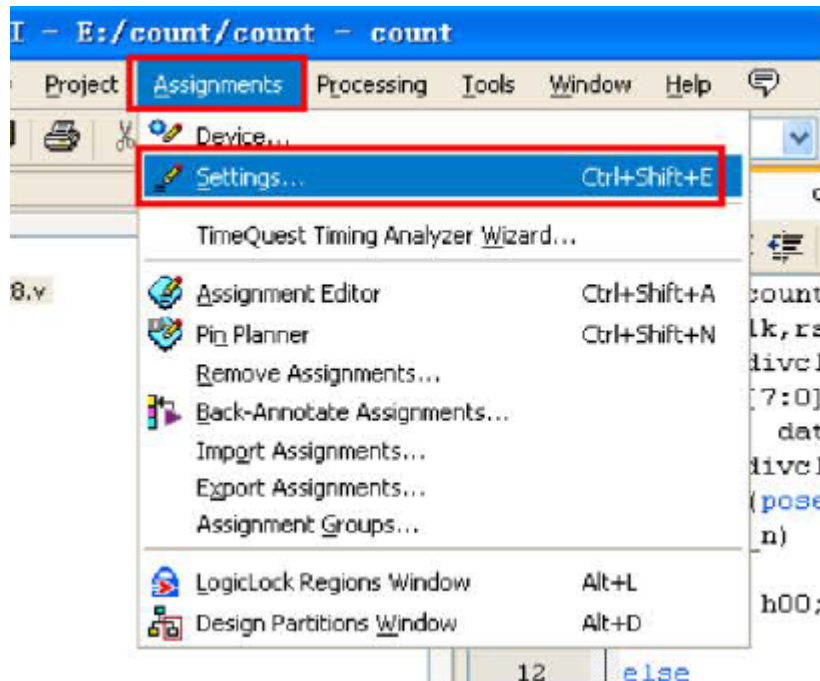


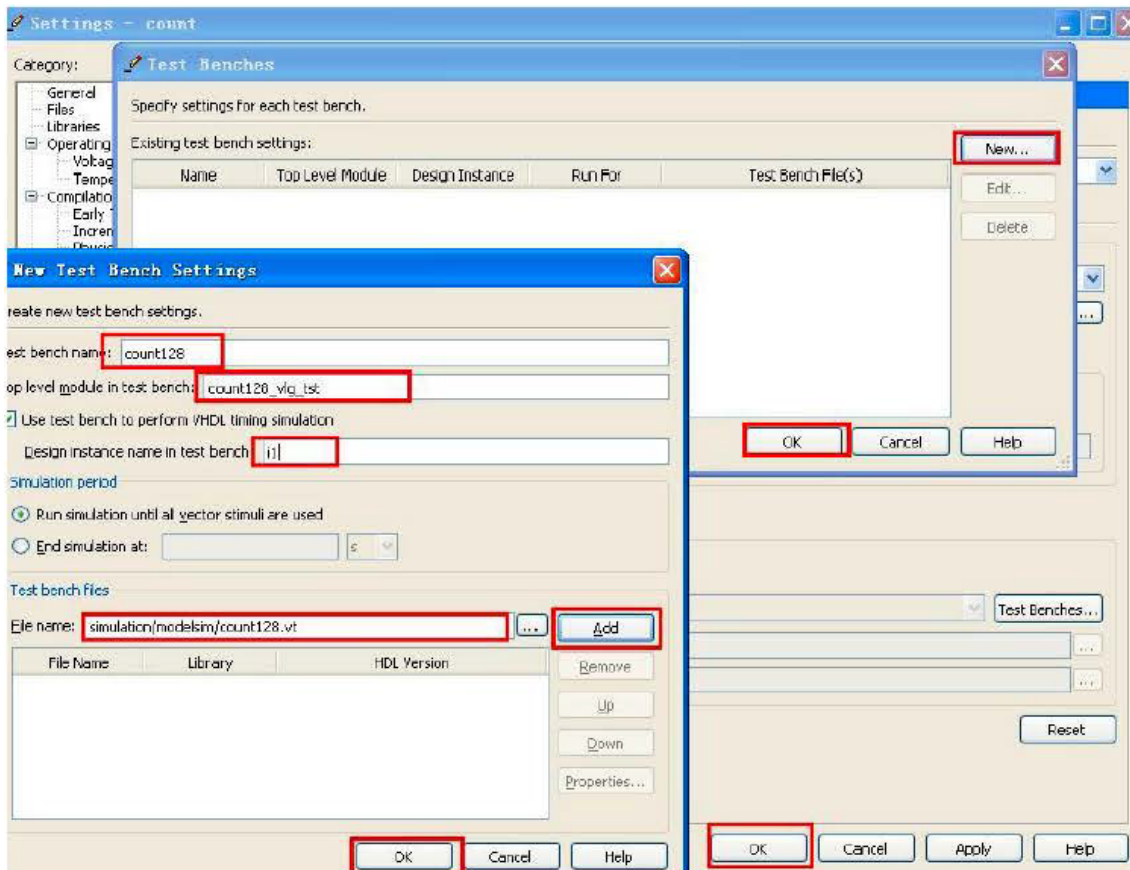




※Modelsimの場所はインストール先により異なる

シミュレーション設定：

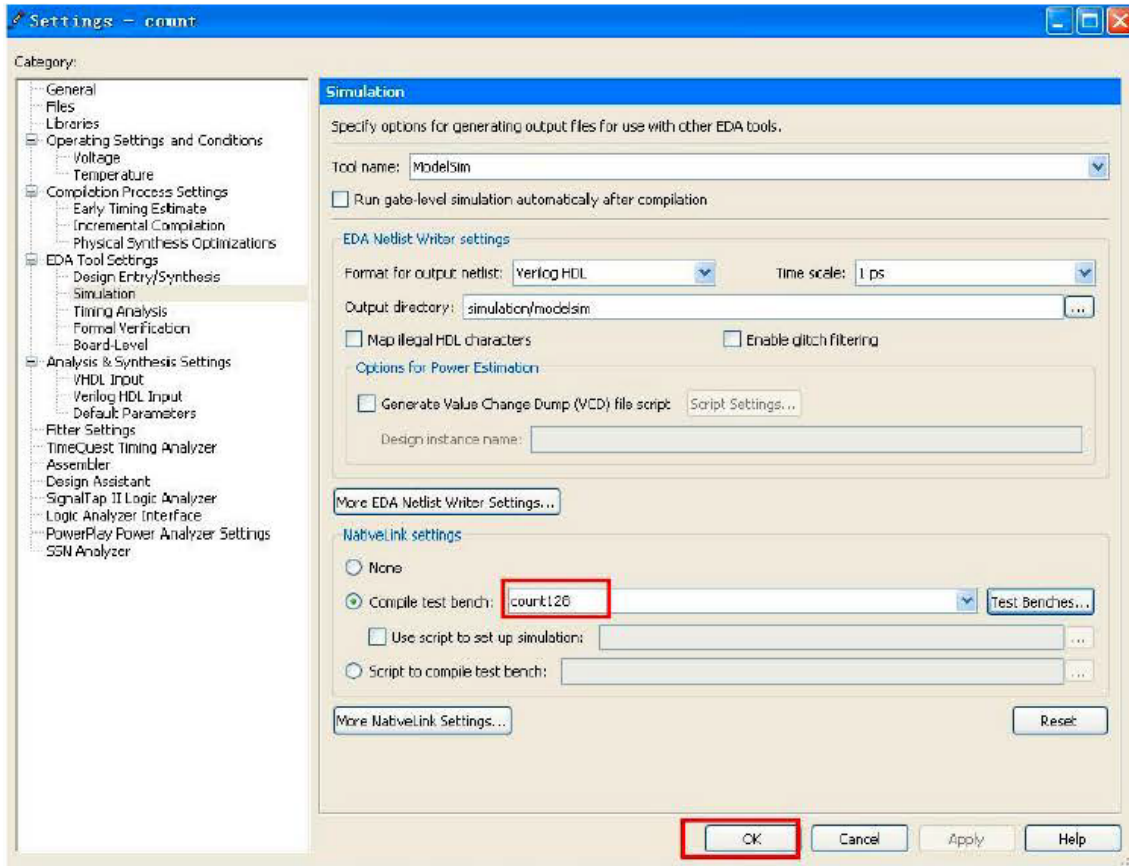




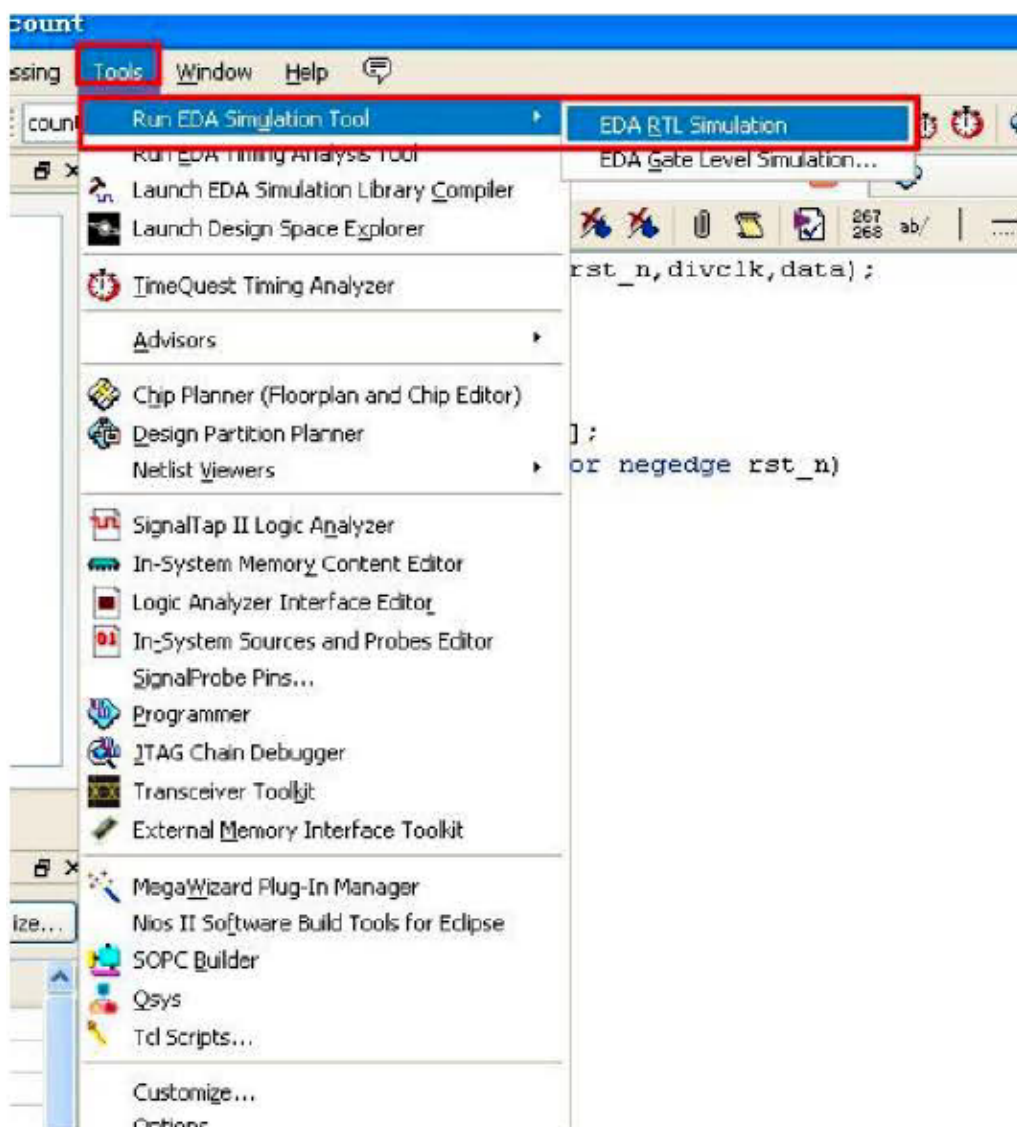


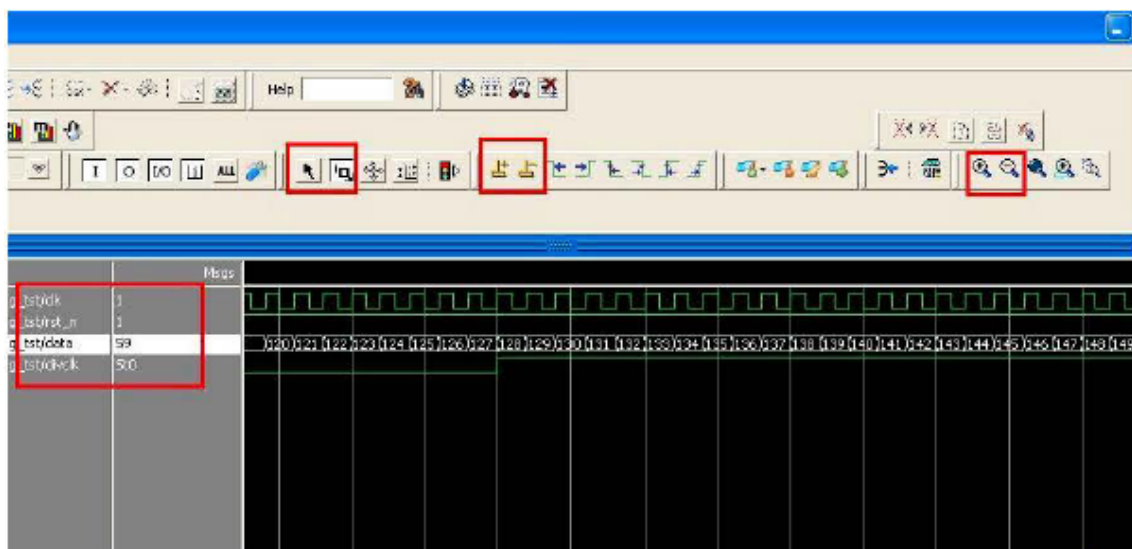
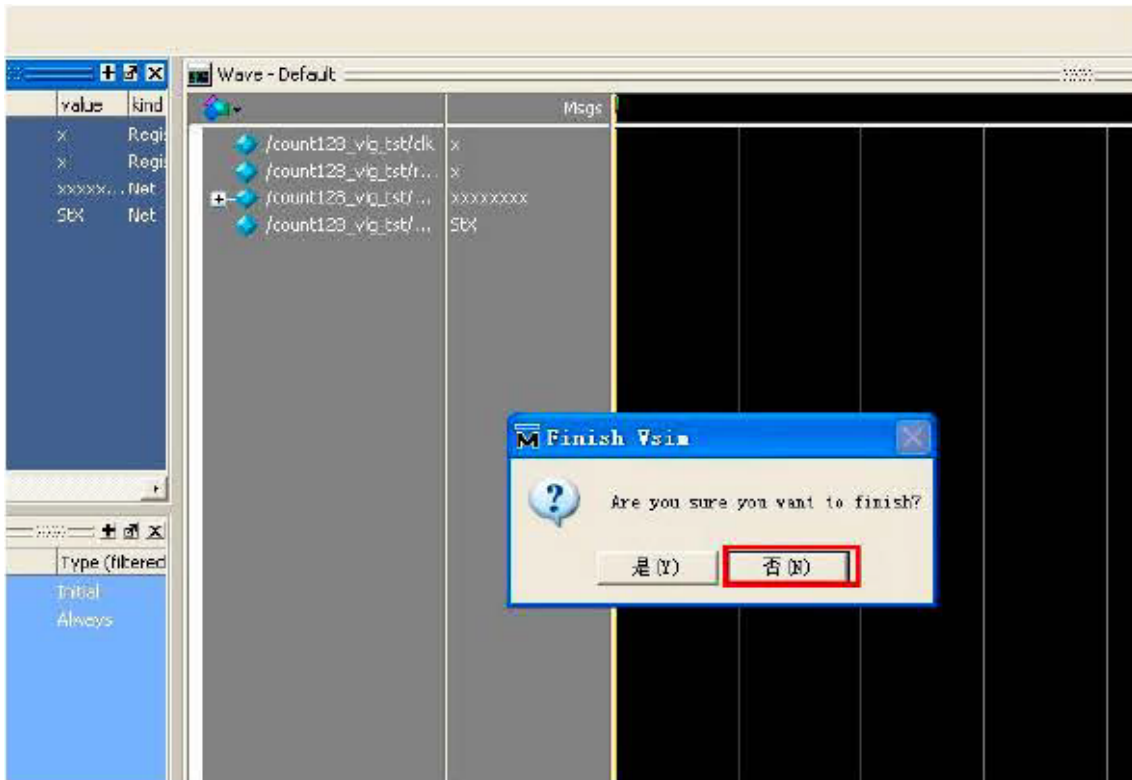
```
`timescale 1 ps/ 1 ps
module count128_vlg_tst();
  reg clk;
  reg rst_n;
  // wires
  wire [7:0] data;
  wire divclk;

  // assign statements (if any)
  count128 i1 (
    // port map - connection between master ports and s
    .clk(clk),
    .data(data),
    .divclk(divclk),
    .rst_n(rst_n)
  );
  initial
begin
  clk=1;
  rst_n=0;
  #50 rst_n=1;
  #10000 $finish;
end
  always #2 clk= ~clk;
endmodule
```



コンパイルしシミュレーションを行う





上記結果により、シミュレーションの波形がでます、いくつか波形ボタン（拡大、縮小、挿入マウス等）が使われます。（赤口に囲まれる）