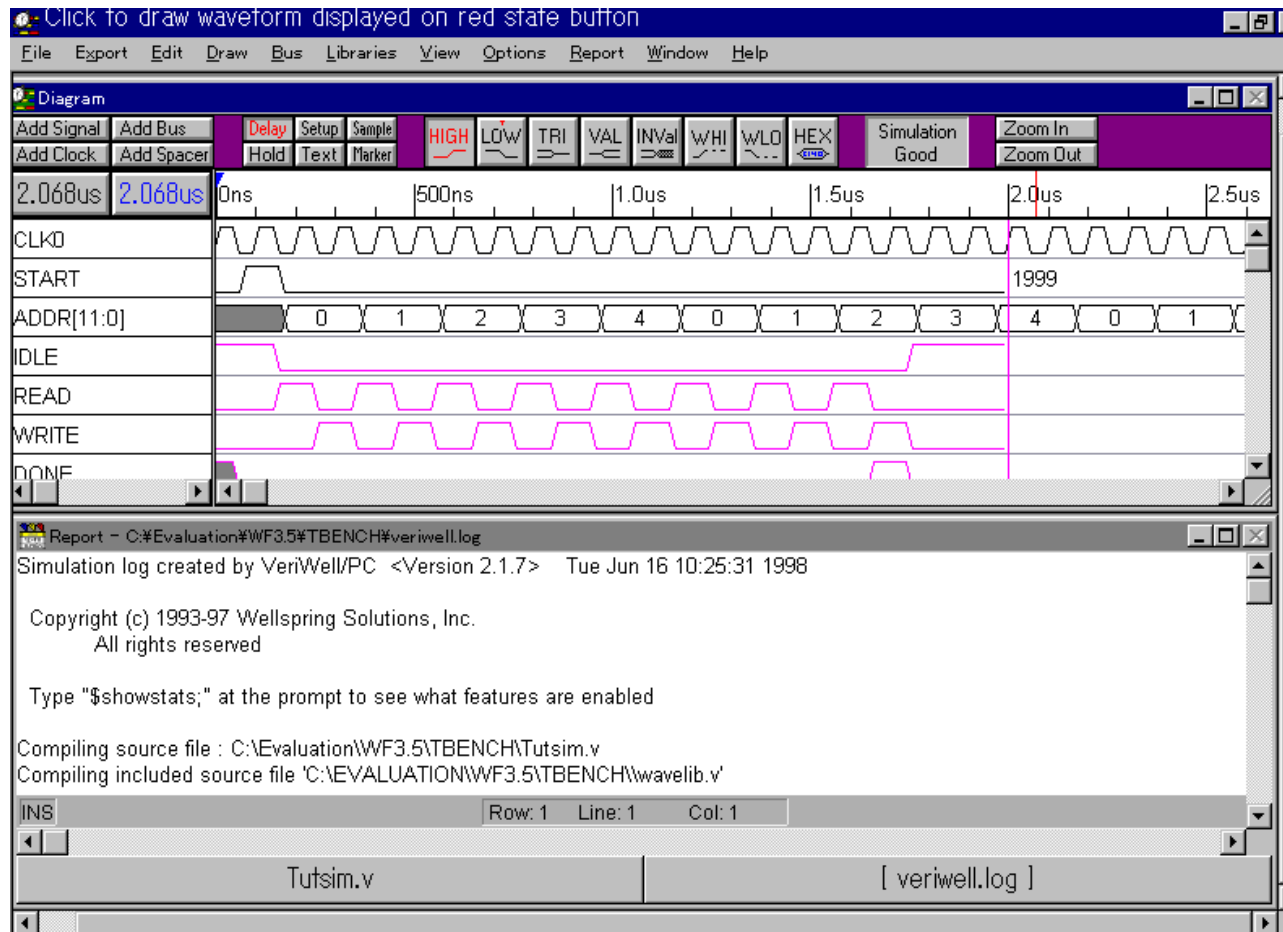


# SynaptiCAD

## WaveFormer Pro

### Verilogger Pro/ TestBencher Pro



**Interactive Verilog Simulator の概要と操作**  
**Verilogger Pro/ TestBencher Pro にも共通 !!**

# What is WaveFormer Pro ?

---

- タイミング・ダイアグラムの編集と  
スタティック・タイミングアナライザでの解析
- ステイミュラスの生成とシミュレータ、ATEのI/Fをサポート  
VHDL、Verilog、ABEL、Minc、ViewLogic、Mentor、  
Altera、Aldec-Xilinx、Spice . . . . .VCD(Verilog )  
HPロジックアナライザ、STIL IEEEテスター・フォーマット  
TDML標準化タイミング記述フォーマット(SI2-ECIX)  
!! インターフェース仕様公開でPerl言語で追加/カスタマイズ可能 !!
- インターラクティブVerilog-HDLシミュレータ
  - マウス操作で入力された波形やRTL論理式から  
Verilog設計 コードを自動生成
  - 変更毎に再シミュレーションが自動実行
  - 入力と確認が同時のためRTL設計入力の効率化
  - Verilogコードの直接入力とインクルード可能

# WaveFormer Pro 超！簡単な基本操作(1)

## 基本操作ボタン



主な操作は、上記のアイコンで実行できます。

# WaveFormer Pro 超！簡単な基本操作(2)

---

基本時間単位設定



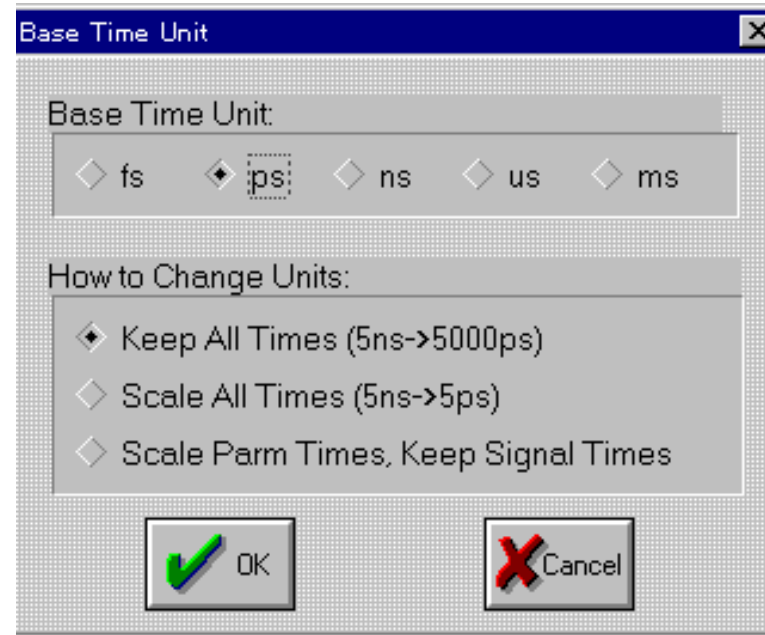
表示時間単位設定



信号の定義/追加



信号の属性を決定

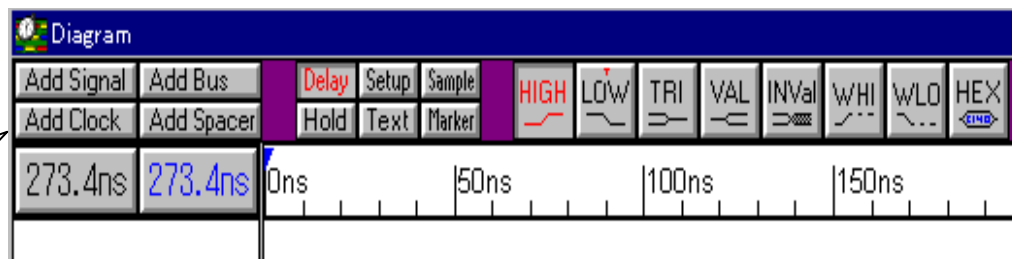


基本時間単位設定  
ダイアログ・ボックス

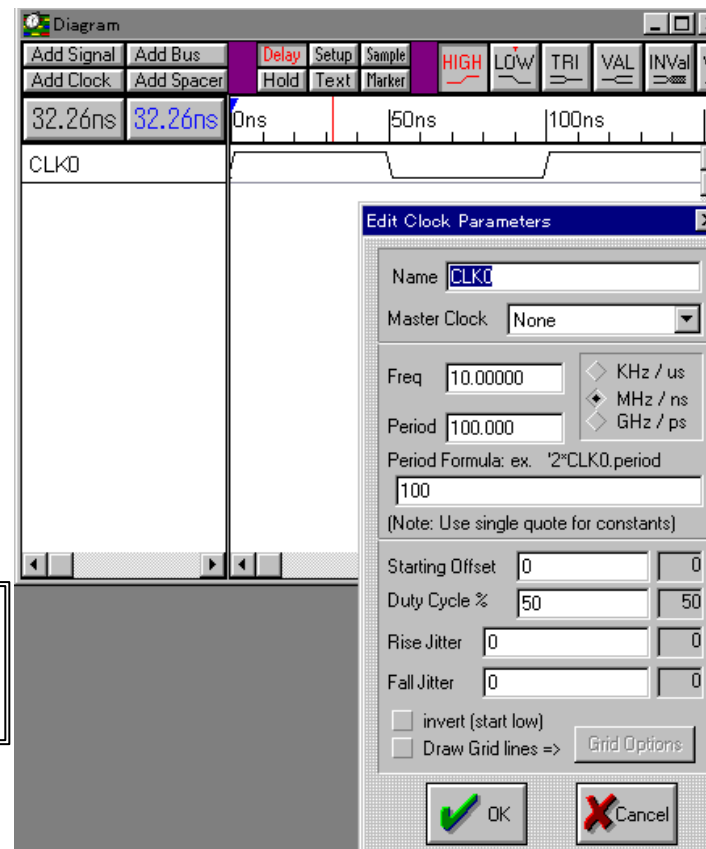
# WaveFormer Pro 超！簡単な基本操作(3)

clockの定義は、

Add Clock ボタンを押す



すると、このようにデフォルトで  
clk0の波形ができ、  
Edit Clock Properties ダイアログ  
が表示されます。  
デフォルト値を確認して、必要なら変更し  
OKを押します。



通常の信号もバスもマウス・クリックまたは簡単な式で素早く入力可能

# WaveFormer Pro 超！簡単な基本操作(4)

信号の定義

信号のHigh, Low ボタン

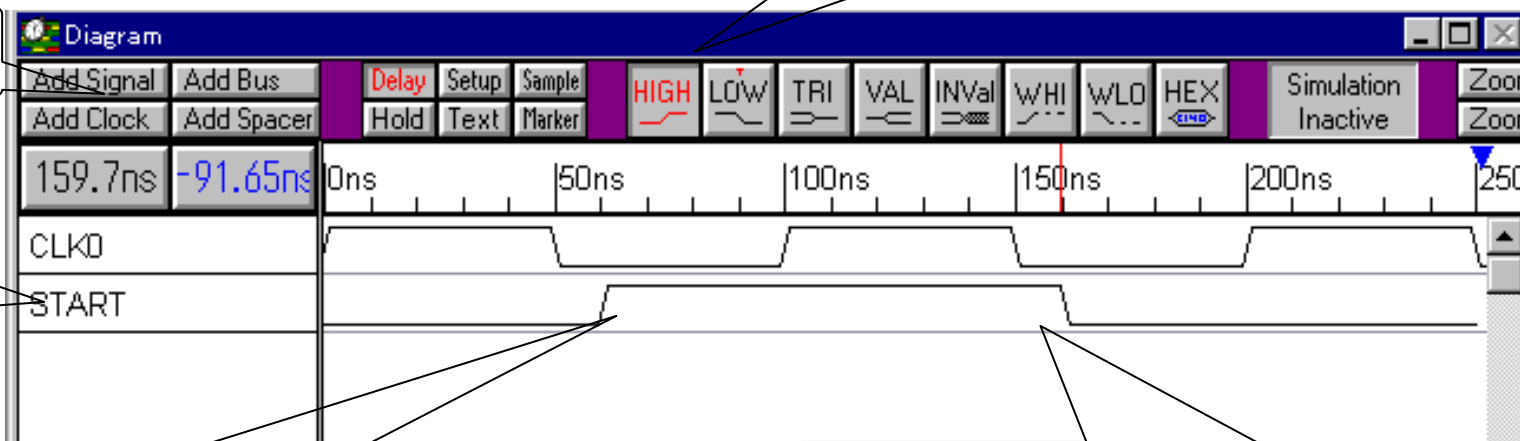
Add Signal ボタン

信号名を  
STARTに

Lowボタンの選択後、  
60nsあたりで左クリック  
\*時間スナップ・グリッドの  
設定可能

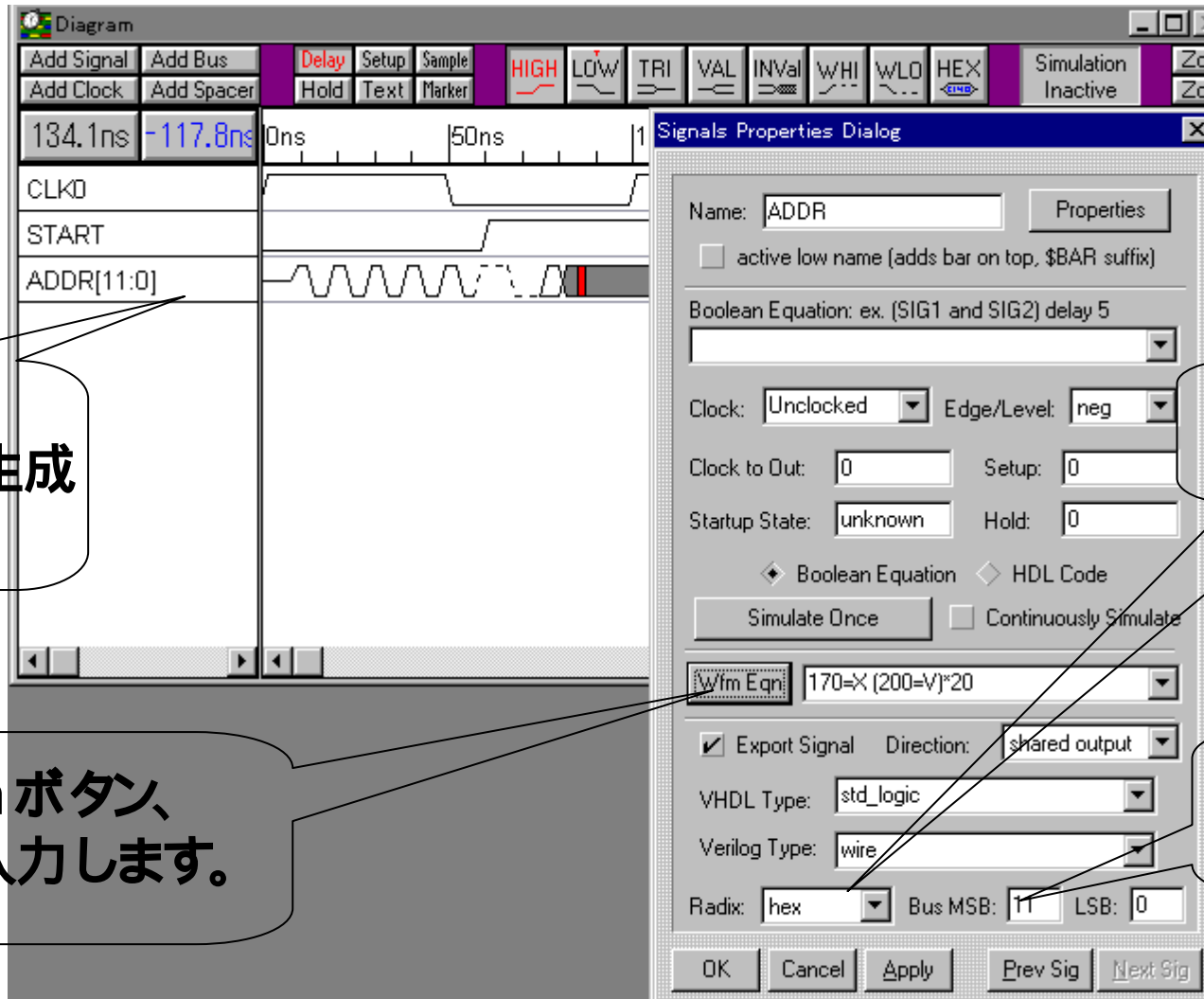
160nsあたりで左クリック

信号はLow, Highが自動で選択  
されます。



# WaveFormer Pro 超！簡単な基本操作(5)

## 時間式での波形入力



時間式入力  
により自動生成  
された波形

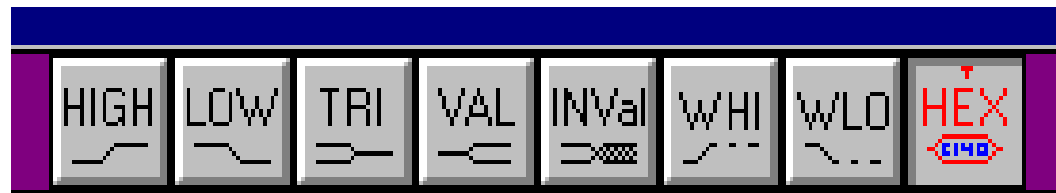
Wfm Eqnボタン、  
右に式を入力します。

Radixをhex

MSBを  
11に設定

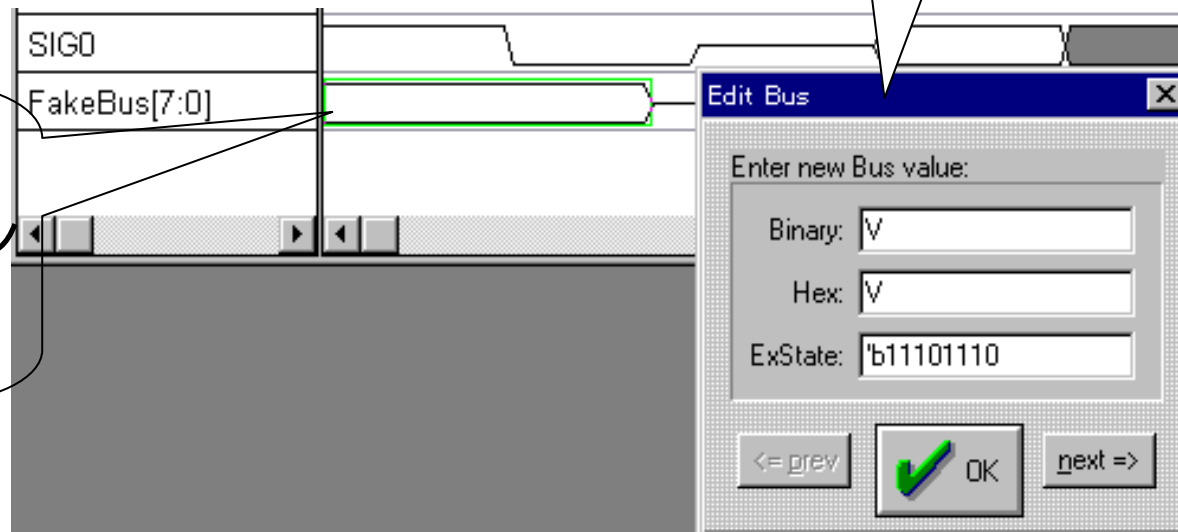
# WaveFormer Pro 超！簡単な基本操作(6)

## バスの値の設定



HEXボタンを押すとダイアログが表示され値を入れることができます。

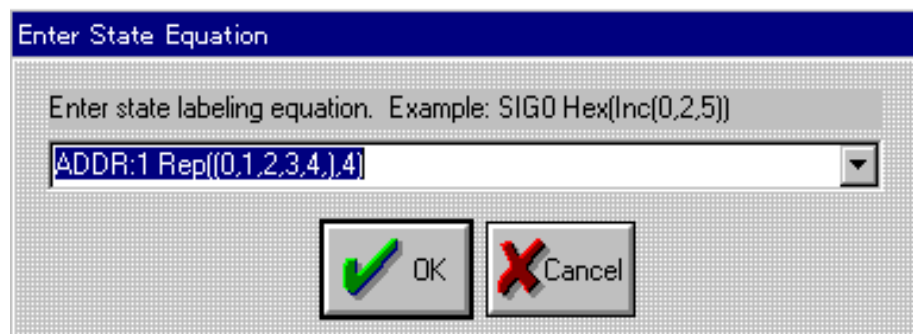
セグメント内で左マウスクリックで選択。グリーン  
の枠で囲まれます。



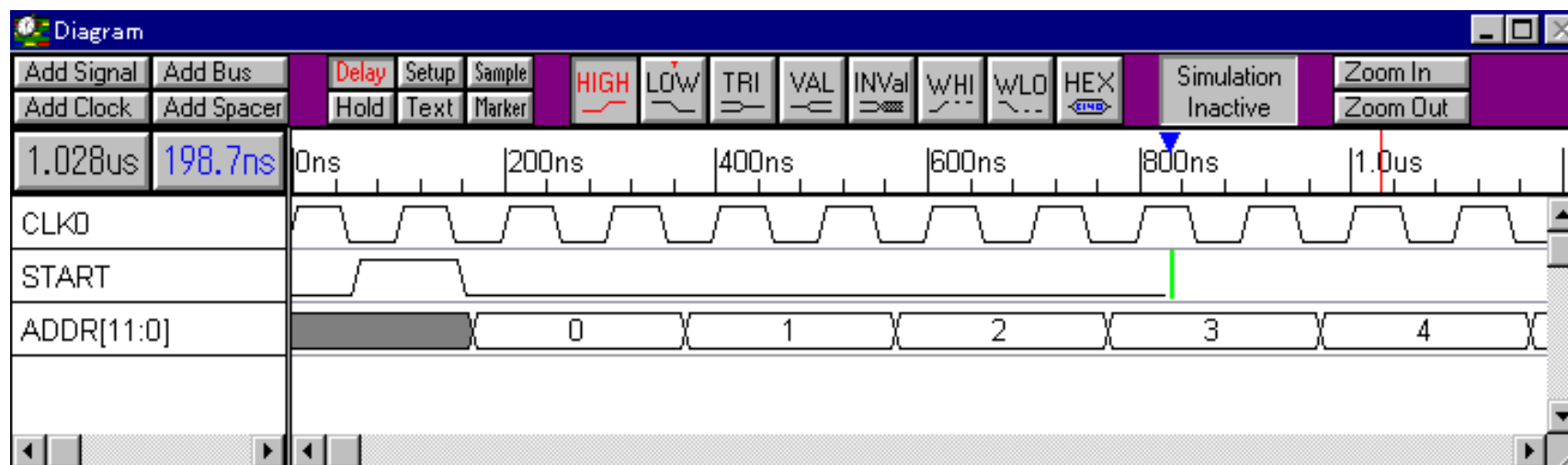


# WaveFormer Pro 超！簡単な基本操作(7)

## -1 バスの値 (ステート名) の入力の自動化



## -2 上記定義で下記のように波形が描けます。



# WaveFormer Pro 超！簡単な基本操作(8)

## RTL 設計入力

ブーリアン論理式を他の信号へ代入

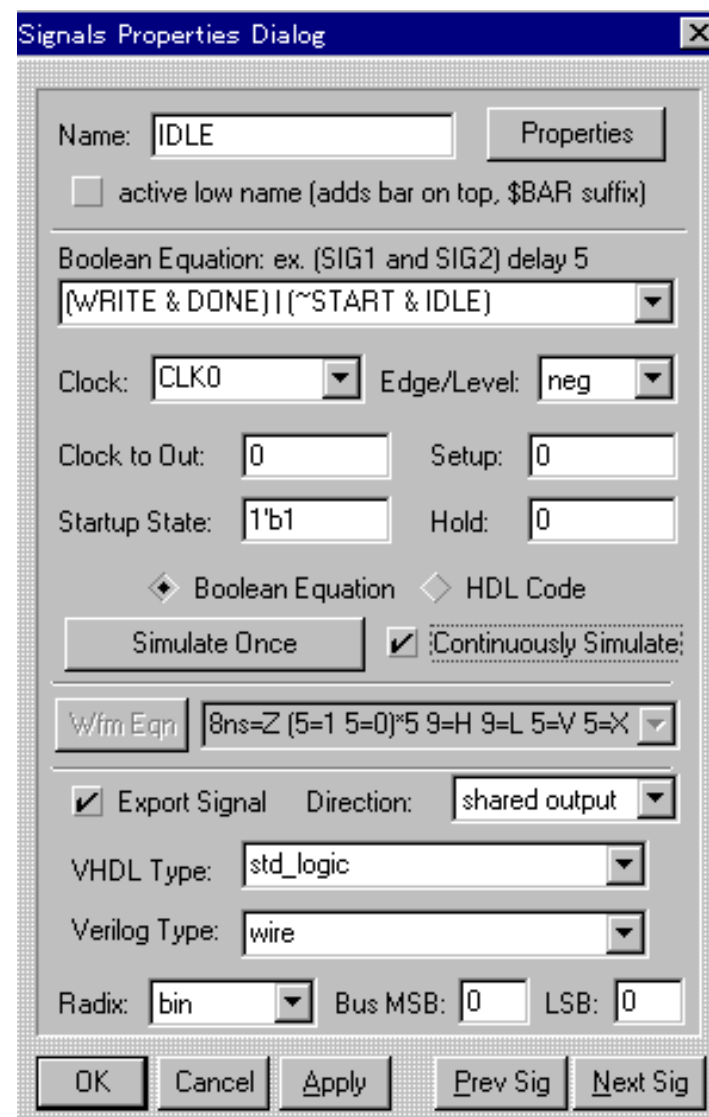
\* 組み合わせ論理として

\* レジスタへ

!! WaveFormer が Verilogコードに変換し

!! VeriWellで自動シミュレーション

(注) HDLコード入力ダイアログに直接に  
Verilogコードを入力することも可能



# WaveFormer Pro 超！簡単な基本操作(9)

WaveFormerにより  
生成されたVerilog コード

充実したエラーログも出力され、  
効率的なデバッグが可能です。

```
TestBench Pro - UNTITLED.TIM - [Report - C:\Evaluation\WF3.5\TBENCH\UNTITLED.sdf]
File Export Edit Draw Bus Libraries View Options Report
timescale 1ns / 1ps

module top;

  wire CLKD;
  wire START;
  wire [11:0] ADDR;
  wire IDLE;
  wire READ;
  wire WRITE;

  testbed tbed(
    CLKD,START,ADDR,IDLE,READ,WRITE
  );

  initial
  begin
    $sdf_annotate("C:/Evaluation/WF3.5/TBENCH/UNTITLED.sdf");
    #4170 $finish;
  end
endmodule

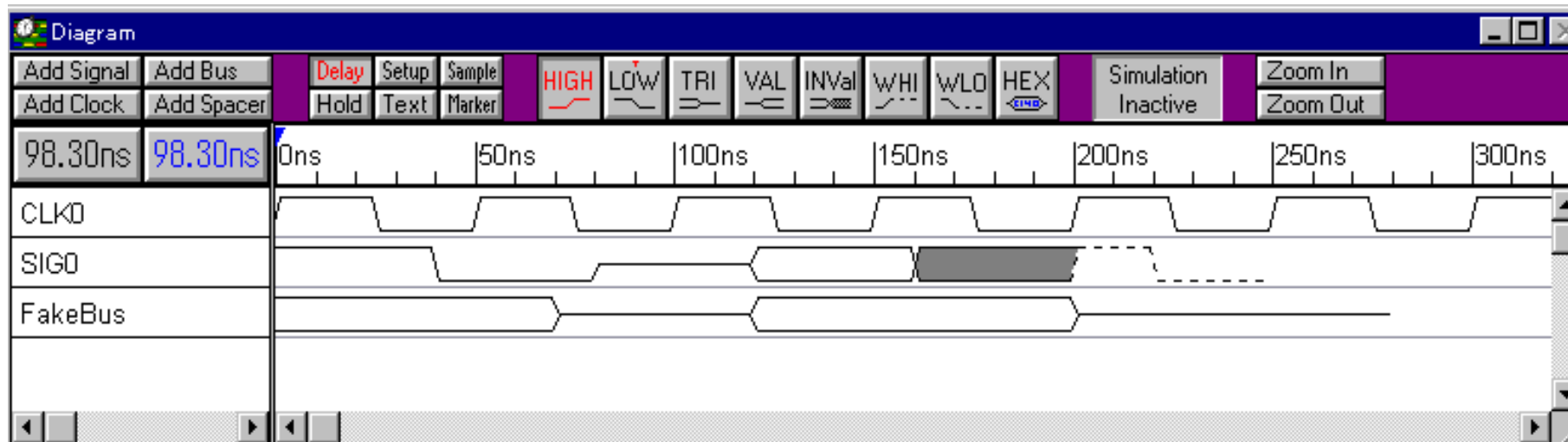
`include "wavelib.v"

module testbed(CLKD,START,ADDR,IDLE,READ,WRITE);

output CLKD;
```

# ステイミュラスの自動生成 Verilog コード(1)

下記 3つの信号によりVerilogステイミュラスの生成例を見てみましょう。



CLK0 : 一周期 50nsのクロック

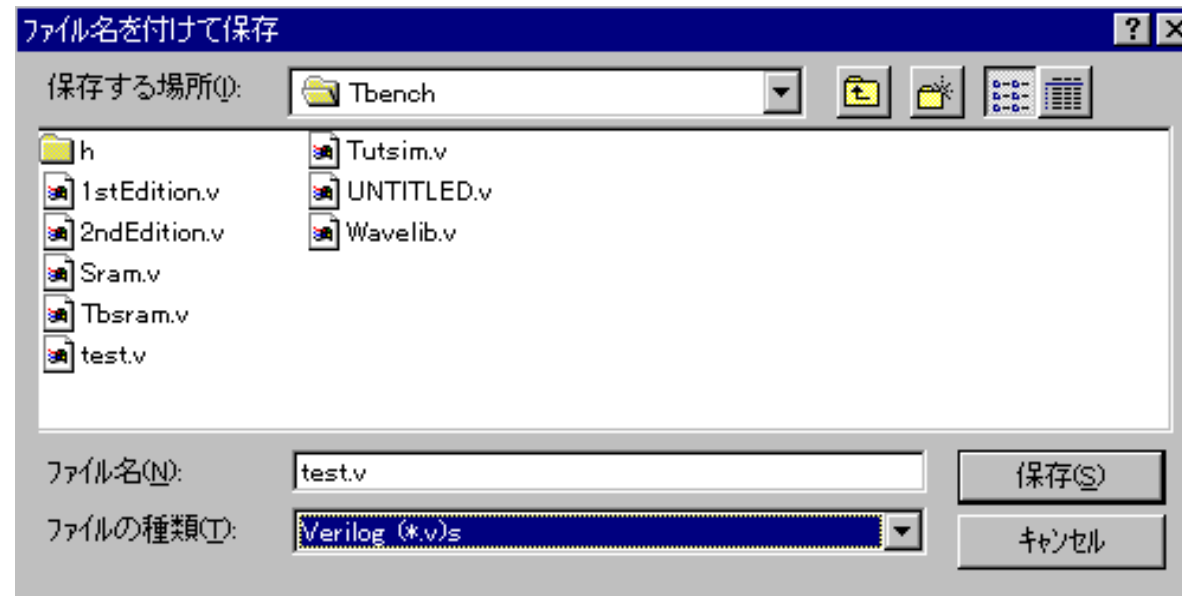
SIG0 : WaveFormerで扱える全ての波形ステートを持った信号

FakeBus : トライステートセグメントを持った仮想バス信号

# ステイミュラスの自動生成 Verilog コード(2)

---

## -1 Verilog-HDL コードの生成



-2 では実際に生成されたテストベンチファイルを見てみましょう。

# ステイミュラスの自動生成 Verilog コード(3)

---

生成されたVerilog  
コードのテストベンチ

```
test.v - 帳帳
ファイル(F) 編集(E) 検索(S) ヘルプ(H)
`timescale 1ps / 1ps
module testbed(CLK0,SIG0,FakeBus);

output CLK0;
reg CLK0;
output SIG0;
reg SIG0;
output [7:0] FakeBus;
reg [7:0] FakeBus;

integer CLK0_stop_time;
integer CLK0_period, CLK0_duty,CLK0_offset;
integer CLK0_d1, CLK0_d2;
initial
begin
//CLOCK CLK0
CLK0_stop_time = 280576;
CLK0_offset = 0;
----- 中略 -----
initial
begin
//SIGNAL FakeBus
FakeBus = 8'b11101110;
#70000
FakeBus = 8'bzzzzzzzz;
#50000
FakeBus = 8'b00001010;
#80000
FakeBus = 8'bzzzzzzzz;
#80576
```

# ステイミュラスの自動生成 Verilogコード(4)

## WaveFormerの大きな特長

システムのインターフェースは全てPerl言語で記述され、WaveFormer Pro本体とのプロトコル仕様が公開されており、カスタマイズ可能  
(例)

Verilogステイミュラスを生成する Verilog.epl ファイルを見てみましょう

```
Verilog.epl - 残帳
ファイル(F) 編集(E) 検索(S) ヘルプ(H)
# Copyright 1996 SynaptiCAD
# Requires TWF v3.0 or greater
#----- Export to Verilog
#(strict compliant)
$|=1;
#uses "future state" format
require 'twfsubs.pl';

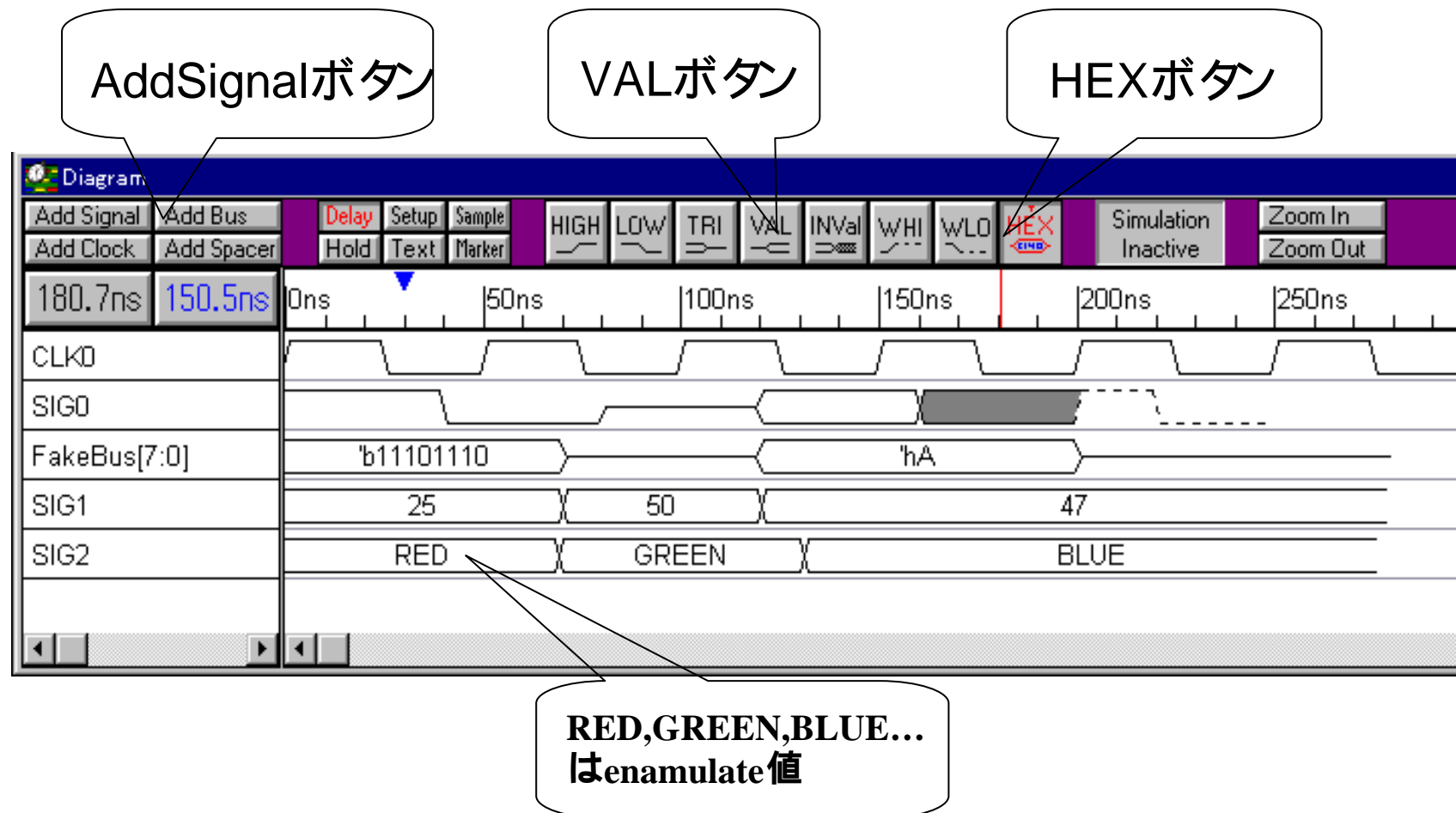
%ToState = ('1' => '1',
            '0' => '0',
            'H' => 'weak1',
            'L' => 'weak0',
            'Z' => 'z',
            'V' => 'x',
            'X' => 'x',
            '! ' => '! ',
            '' => '' );

IF_To_Verilog();
return 1;

sub IF_To_Verilog {
    $LastOutputTime = twf::GetLastOutputTime();
    $BUnits = $twf::ToUnits[ twf::GetOptions()->GetBaseTimeUnit() ];
    $DUnits = $twf::ToUnits[ twf::GetOptions()->GetBaseTimeUnit() ];
    print "`timescale 1$DUnits / 1$BUnits\n";
}
```

# ステイミュラスの自動生成 VHDL コード(1)

VHDLでは直接値入力以外に enamulation をサポート

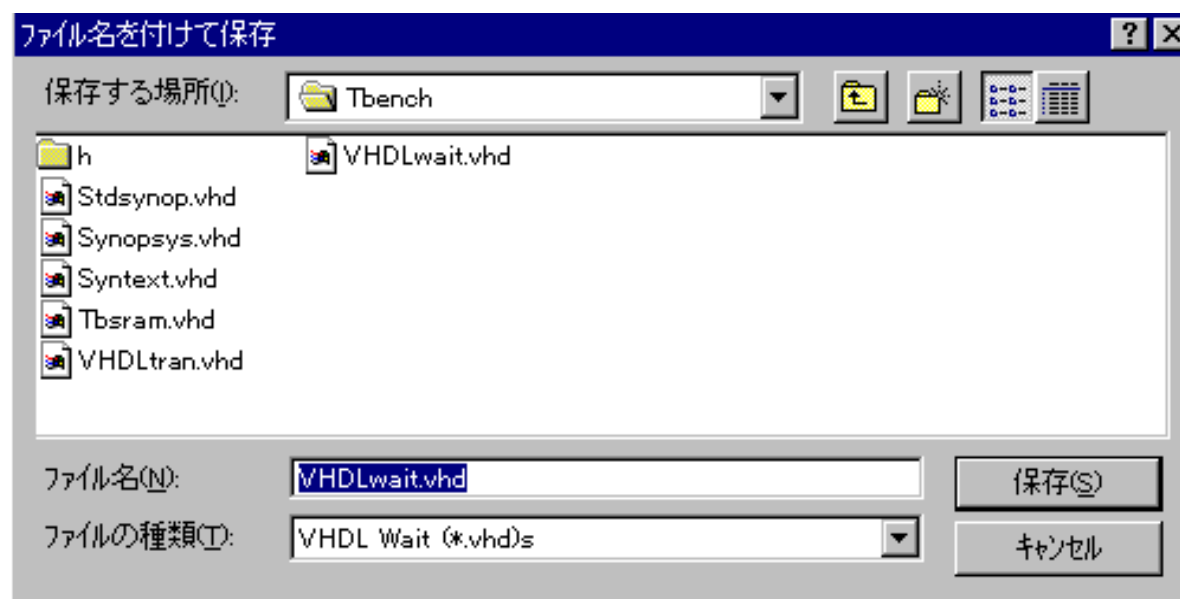




## ステイミュラスの自動生成 VHDL コード(2)

---

VHDLtran.vhd とVHDLwait.vhd を作成します。



# ステイミュラスの自動生成 VHDL コード(3)

VHDLtran.vhd

```
process
begin
  SIG0 <=
    transport '1',
      '0' after 40 ns,
      'Z' after 80 ns,
      'X' after 120 ns,
      'X' after 160 ns,
      'H' after 200 ns,
      'L' after 220 ns;

  FakeBus <=
    transport "11101110",
      "ZZZZZZZZ" after 70 ns,
      "00001010" after 120 ns,
      "ZZZZZZZZ" after 200 ns;

  SIG1 <=
    transport 25,
      50 after 70.144 ns,
      47 after 120.32 ns;

  SIG2 <=
    transport RED,
      GREEN after 69.12 ns,
      BLUE after 131.072 ns;

  wait;
end process;

end test;
```

VHDLwait.vhd

```
process
begin
  CLK0 <= '0';
  wait for 0 ns;
  while true loop
    CLK0 <= '1';
    wait for 25 ns;
    CLK0 <= '0';
    wait for 25 ns;
  end loop;
end process;

process
begin
  SIG0 <= '1';
  FakeBus <= "11101110";
  SIG1 <= 25;
  SIG2 <= RED;
  wait for 40 ns;

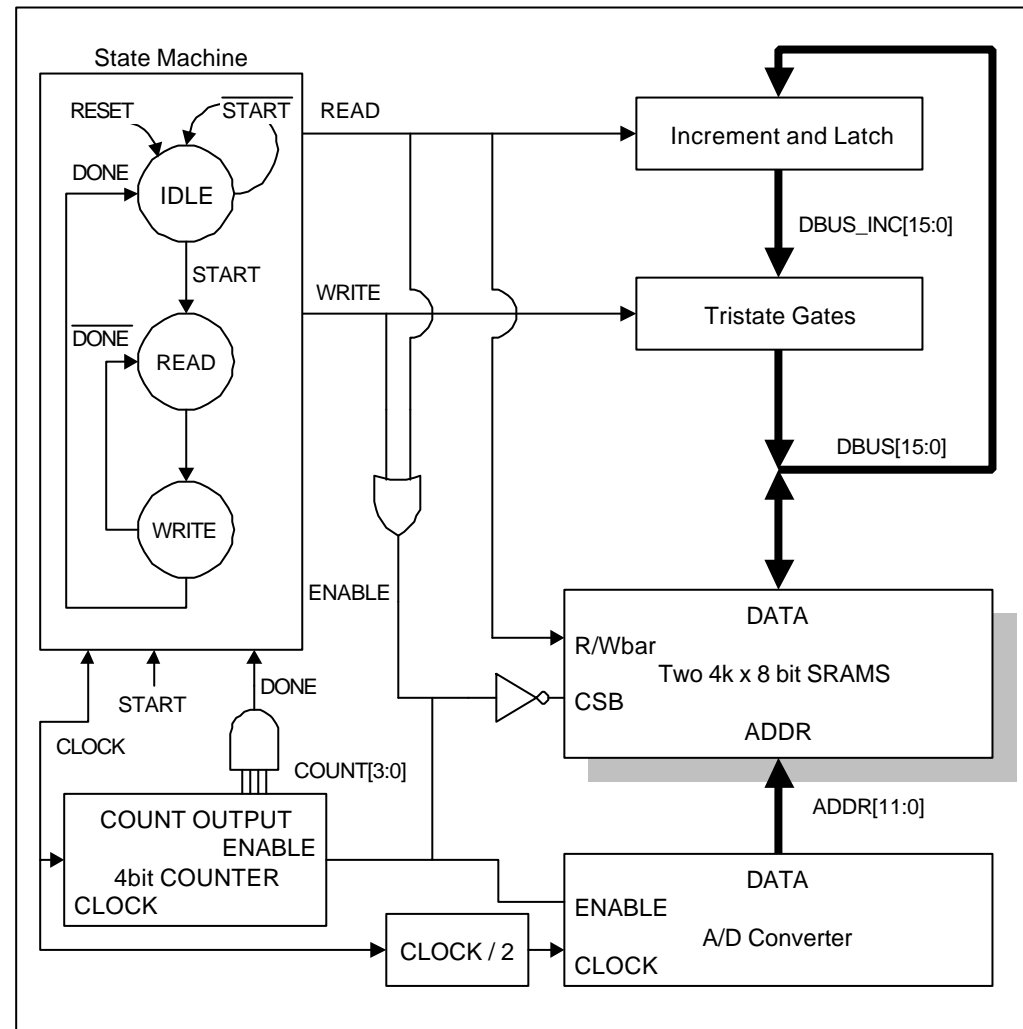
  SIG0 <= '0';
  wait for 29.12 ns;

  SIG2 <= GREEN;
  wait for 0.88 ns;

  FakeBus <= "ZZZZZZZZ";
  wait for 0.144 ns;
```

# WaveFormer Pro インターラクティブ シミュレータ (1)

この程度の中規模の設計であれば RTL入力機能で素早くモデル化可能



# WaveFormer Pro インターラクティブ シミュレータ (2)

前ページのモデル = シミュレーション波形とVerilogコードの同時出力

