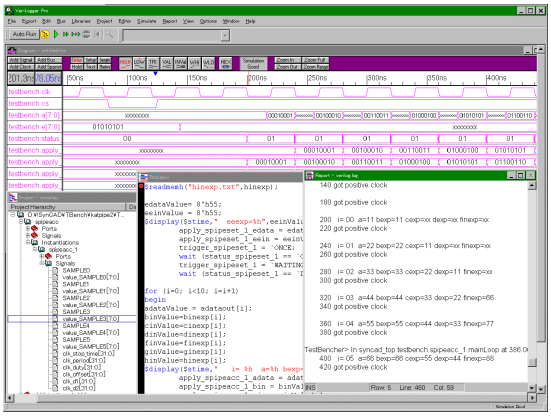
SynaptiCAD

VeriLogger Pro Verilog Simulator++

IEEE 1364準拠のVerilog-HDLシミュレータ
SDFバック・アノテーション，VCD出力，PLI(オプション)などをサポート
しかもWaveFormer Proの全機能も標準装備したハイパー・ツール

回路設計において，設計した回路のシミュレーションは必須の作業ですが，シミュレータに与えるテスト・スティミュラスの準備や，ソース・コードの管理，およびデバッグは面倒な作業です．VeriLogger Proはテスト・スティミュラスを作成しながらシミュレーションを行えるインタラクティブ・シミュレーション機能や，ソース・コードの管理を行うプロジェクト・マネージャ，ソース・レベル・デバッガなどを装備した使いやすいVerilog-HDLシミュレータです．しかも，WaveFormer Proのテスト・スティミュラス作成機能など，同商品の全機能も装備したハイパー・ツールです．

VeriLogger Proの概要



VeriLogger Proは，**IEEE1364に準拠するVerilog-HDLシミュレータ**です．シミュレーション性能に定評のあるWellspring Solutions社[VeriWell]のVerilog-HDLシミュレーション・エンジンを採用し，波形作成/編集/解析ツール[WaveFormer Pro]の使いやすいユーザ・インターフェースを装備しました．

VeriLogger Proは，作成したVerilog-HDLソース・コードをプロジェクトという単位で管理し，階層構造を自動的に認識してコンパイルを行うので，ユーザが階層構造に気を遣う必要はありません．またソース・コードに文法エラーがあれば，エラーのある行とエラー・メッセージを表示し，内蔵の**シンタックス・カラーリング・エディタ**にエラー行を表示するので，これを修正できます．

シミュレーションは，**Auto Runモード**と**Debug Runモード**の二種類があり，それぞれ連続実行，ステップ実行(ソース・レベル・デバッグ)が可能です．また，テスト・スティミュラスを波形ダイヤグラム上に作成しながらシミュレーションを行うこともできるので，作成したソース・コードのシミュレーションをテスト・スティミュラスなしですぐに行いたい時には非常に便利です．

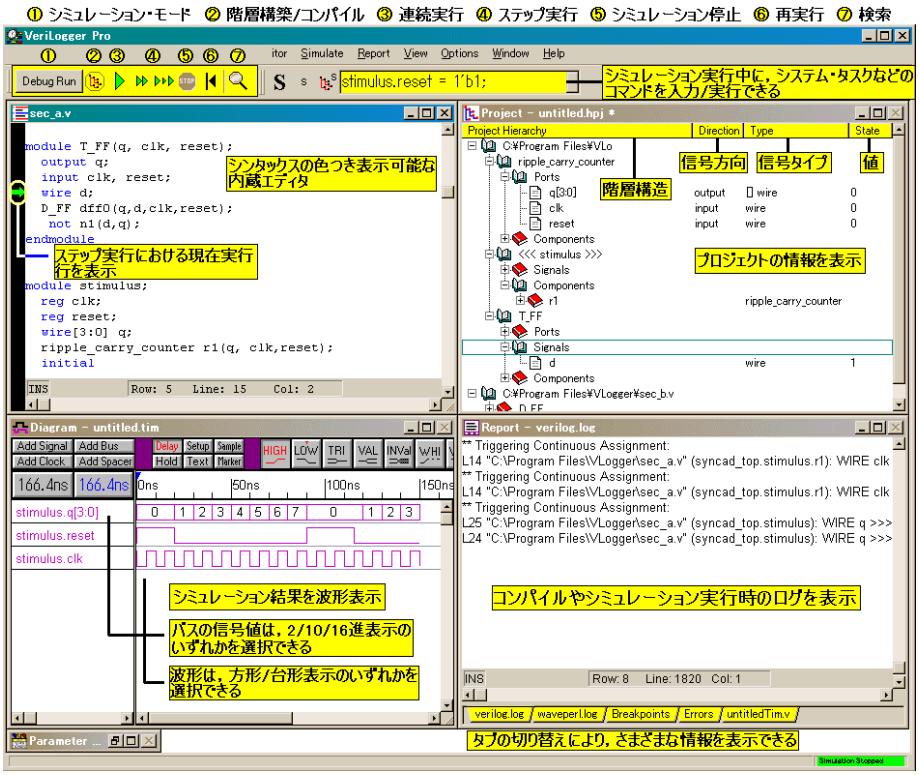
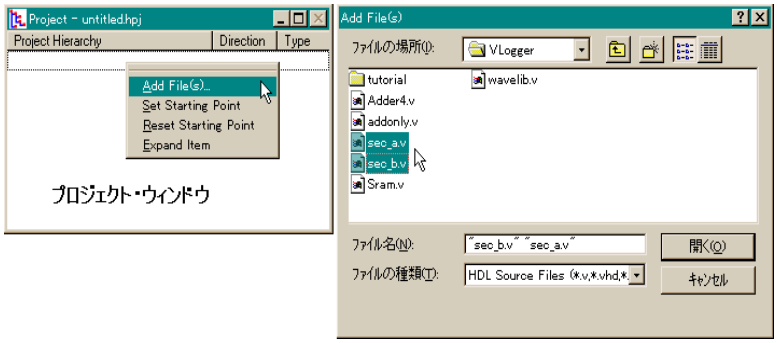


図1にVeriLogger Proの主要ウィンドウと操作ボタンのようすを示しました．ほとんどの処理は**図1**の①から⑦までのボタン操作で行えます．

また，ソース・コードの階層構造や信号名など，各種情報の表示は**図1**に示したProjectウィンドウから得ることができます．

図1 VeriLogger Proの主要ウィンドウとボタンの機能説明

プロジェクトの作成／ビルド ソース・コードを登録するだけで自動管理



新規プロジェクトの作成-ソース・コードの登録

VeriLogger Proを使い始めるにはまず**プロジェクト**を作成します．これはシミュレーションを行うVerilog-HDLソース・コードを登録する作業です．

Projectウィンドウでマウスの右ボタンをクリックしてAdd File(s)メニュを起動した後，必要なソース・コードを選択します(**図2**)．

なお，ソース・コードの登録は，ユーザが階層構造を意識しながら順に登録を行う必要はありません．また必要なファイルを一括して登録できるので，何度もファイル操作を行う必要がありません．

図2 新規プロジェクトの作成

プロジェクトのビルド-ソース・コードのコンパイル

ソース・コードを登録したら**プロジェクトのビルド**を行います．これは，登録したソース・コードをコンパイルし，各ファイル間の関係を認識する作業です．これは，**図1**に示す②ボタンをクリックするだけでVeriLogger Proが自動的に行います．

もし，ソース・コードに文法エラーがあれば**図3**に示すReportウィンドウにエラー・メッセージとエラーのある行番号が表示されます．これをダブル・クリックすれば内蔵のエディタにエラーのあるソース・コードがエラー行に矢印を付けて表示されるので，誤りのある個所をすぐに見つけられ，修正ができます．

正常にコンパイルが終了すると，**図3**に示すProjectウィンドウにソース・コードの階層構造が表示されます．

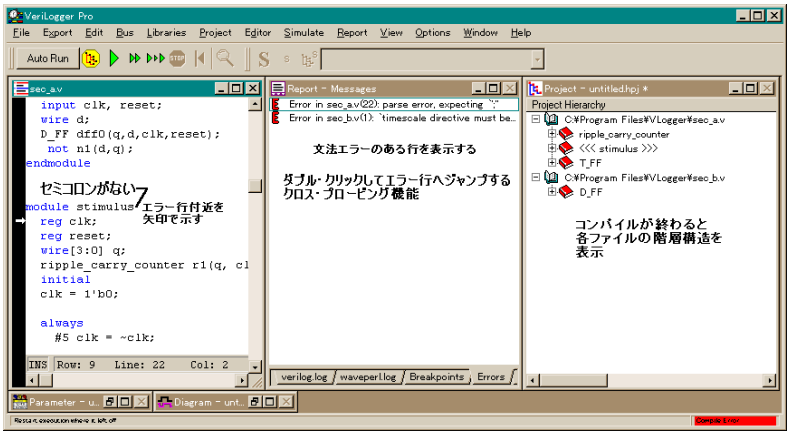
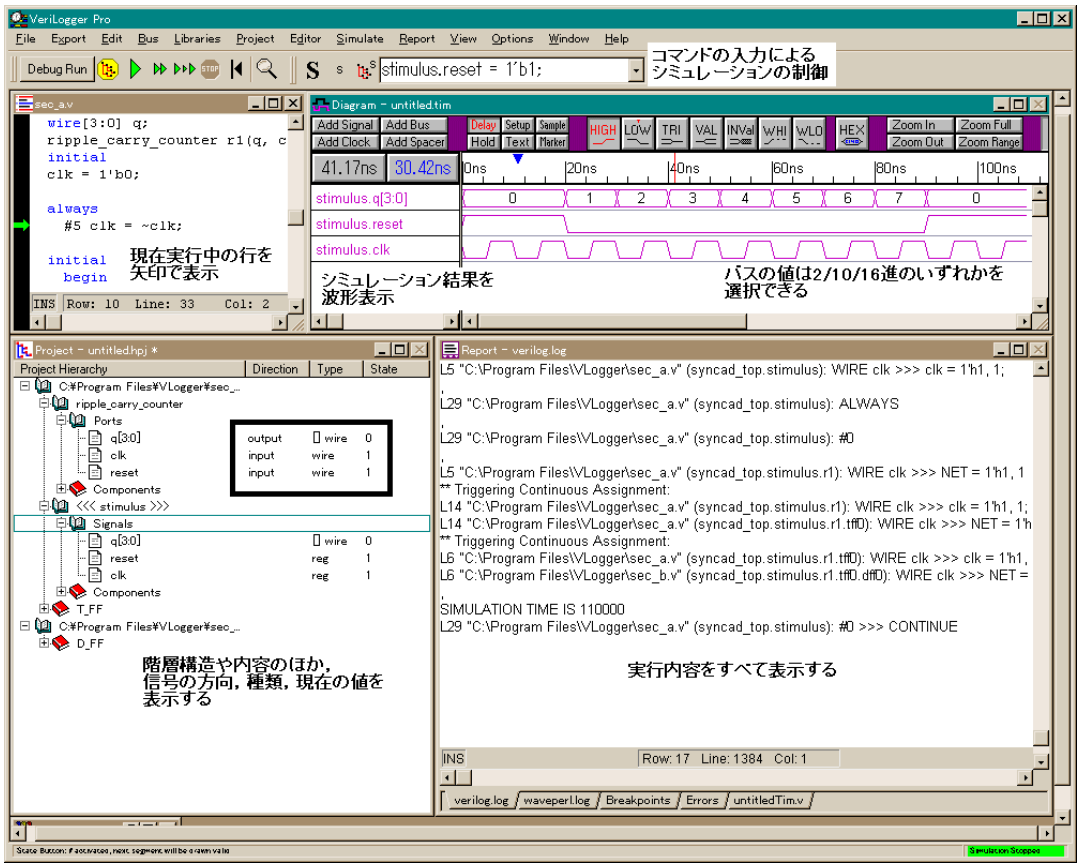


図3 プロジェクトのビルド

シミュレーション 連続実行とステップ実行（ソース・レベル・デバッグ），ブレーク・ポイントの設定が可能



プロジェクトのビルドが完了すればすぐにシミュレーションが始められます．

ビルドと同時に，Diagramウィンドウにはソース・コード中の最上位モジュールの信号が列挙されるので，ユーザが登録を行う必要はありません．

連続実行

これは，ソース・コードに記述されたテスト・スティミュラスにしたがってシミュレーションを行います．すばやくシミュレーション結果を見たい場合はこのモードでシミュレーションを実行します．

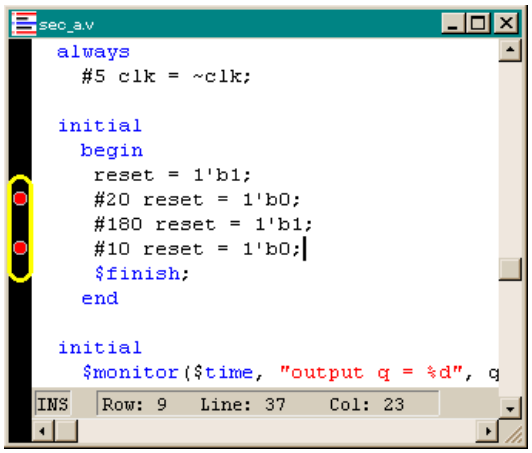
なお，**図1**に示す③ボタンをクリックすれば，連続シミュレーションが実行されます．

図4 シミュレーション例

ステップ実行(ソース・レベル・デバッグ)

連続実行で期待する結果が得られないときなどはステップ実行を行います．これはソース・コードを一つづつ実行するものです．まず**図1**に示す①ボタンをクリックして**Debug Run**を表示させます．つぎに**図1**に示す④ボタンをクリックするごとにソース・コードが一つづつ実行されます．このとき現在実行中の行が，矢印で示されて内蔵のエディタに表示されます(**図4**)．また，**Projectウィンドウ**には任意の信号について現在の値が表示されます(**図4**)．さらに，**Diagramウィンドウ**にはシミュレーション結果を波形で，**Reportウィンドウ**にはソース・コードの実行過程が表示されるので，これらを見ながら実行経過を詳細に追跡できます．

ブレーク・ポイントの設定



VeriLogger Proは，複数のブレーク・ポイントを設定できます．これは，設定した特定のソース・コードまで実行を行ったのち，シミュレーションを一時停止するというものです．これにより，ブレーク・ポイントまでは連続してシミュレーションを行い，この後はステップ実行で詳細な実行経過を追跡する，といった実行形式が実現できます．

図5に示すように，ブレーク・ポイントは，一時停止させたいソース・コードの行をクリックするだけで設定できます．また，設定されたブレーク・ポイントには赤い丸印が表示されます．ブレーク・ポイントを削除するには，設定した行を再びクリックするだけです．

なお，設定されたブレーク・ポイントは，連続実行あるいはステップ実行のいずれにおいても有効です．

図5 ブレーク・ポイントの設定

コマンド入力による対話的なデバッグ コマンド入力によるシミュレーションの制御

VerLogger Proは，ステップ実行時Verlog-HDLのステートメントを入力してシミュレーションを制御することができます．たとえばテスト・スティミュラスに記述されていない値をポートに与えて，これによるシミュレーション結果を見ることができます．**図6**では，テスト・スティミュラスにはない，stimulus.resetポートに’1’を与える(stimulus.reset = 1’b1;)というVerilog-HDLの代入文を途中で実行する例を示しています．

このように，VeriLogger Proはテスト・スティミュラスを新たに作成し直すことなくシミュレーションを継続できる対話的なデバッグ環境を装備しています．もちろん$displayなどのシステム・タスクの入力/実行も可能です．

なお，入力したコマンドは記憶されているので，以前に入力したコマンドは再びキー入力する必要がなく，メニュから選択するだけでコマンド入力が行えます(**図6**)．

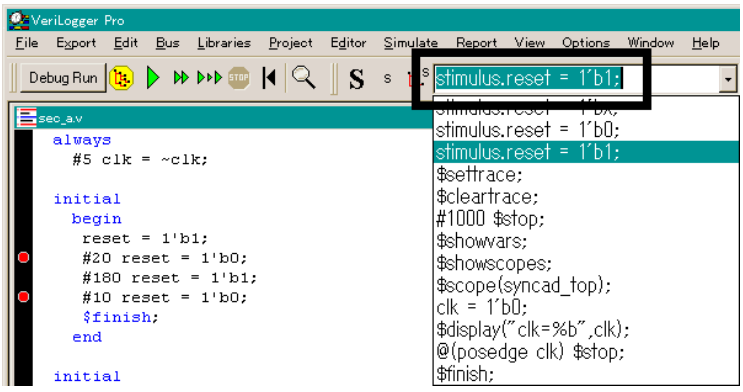


図6 Verilog-HDLステートメントの入力

インタラクティブ・シミュレーション テスト・スティミュラスを作りながらシミュレーションを実行

リスト1 テスト・スティミュラスのないソース・コード

```
`timescale 1ns / 1ps

module ripple_carry_counter(q,clk,reset);
   output[3:0] q;
   input clk, reset;
   T_FF tff0(q[0],clk, reset); T_FF tff1(q[1],q[0], reset);
   T_FF tff2(q[2],q[1], reset); T_FF tff3(q[3],q[2], reset);
endmodule

module T_FF(q, clk, reset);
   output q;
   input clk, reset;
   wire d;
   D_FF dff0(q,d,clk,reset);    not n1(d,q);
endmodule
```

デザインの機能記述とともにテスト・スティミュラスの作成も手間を要する作業です．小規模なモジュールの機能記述をすばやくシミュレーションしたいが，テスト・スティミュラスの作成がめんどうなために，シミュレーションをあきらめてしまう場合があります．このような時にはVeriLogger Proのインタラクティブ・シミュレーション機能が役立ちます．

これは，テスト・スティミュラスを含まないソース・コードを含むプロジェクトをビルドした場合，入力ポートに対して，Diagramウィンドウ上でスティミュラスに相当する波形を描き入れることができ，これを元にシミュレーションを行うことができるものです．

リスト1は，テスト・スティミュラスのないVerilog-HDLソース・コードの例です．これを含むプロジェクトをビルドする

と，ダイヤグラム・ウィンドウにはclk，resetおよびq[3:0]が表示されますが，入力信号のclkとresetは入力信号を表す黒色で，出力信号のq[3:0]は出力信号を表す桃色で表示されます(**図7**)．ここで，clkおよびresetに波形を描き入れると，描かれた波形にしたがってq[3:0]に出力が現れます(**図7**)．

つまり，描き入れた波形がVeriLogger ProによってVerilog-HDLのスティミュラスに自動変換され，シミュレーションが実行されるのです．さらに，入力した波形はエッジ位置の移動などの修正が可能で，修正されるごとにシミュレーションが実行され，結果がアップデートされます．

このように，VeriLogger Proは，テスト・スティミュラスを作成/編集しながらシミュレーションが行える，対話的(インタラクティブ)なシミュレーション機能も装備しています．

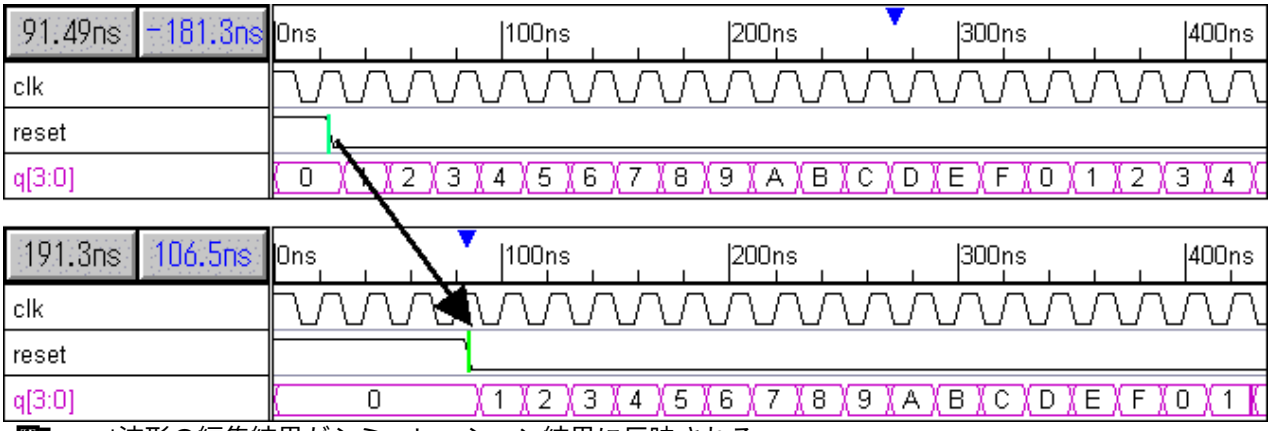


図7 reset波形の編集結果がシミュレーション結果に反映される

VeriLogger Proのおもな機能　WaveFormer Proのすべての機能も標準装備

リスト2　VeriLogger Proのおもな機能一覧

- ✓ IEEE1364準拠
- ✓ ビヘイビア，RTLおよびストラクチャのいずれのシミュレーションもサポート
- ✓ SDF(Standard Delay Format)によるタイミング解析が可能
- ✓ Programming-Language Interface 1.0(PLI)[オプション]
- ✓ 高速な起動とコンパイルを実現
- ✓ 階層ブラウザによるソース・コードの管理/ステータス表示が可能
- ✓ ステップ実行(ソース・レベル・デバッグ)やブレーク・ポイントの設定が可能
- ✓ シンタックス・カラーリング・テキスト・エディタを装備
- ✓ GUIモードと，Verilog-XLのコンソール・モードの両方をサポート
- ✓ インタラクティブ・シミュレーション機能を装備
- ✓ シミュレーション結果のVCDフォーマットでの出力が可能
- ✓ WaveFormer Proの持つすべての機能も標準装備

上記の機能以外にも，SDFファイルによるタイミング解析や，シミュレーション結果のVCDフォーマットでの出力，およびオプションでのPLIの装備などVerilog-HDLシミュレータとして必要な基本機能も装備しています．

VeriLogger Proが装備するおもな機能一覧については**リスト2**を参照してください．

なお，VeriLogger Proは波形作成/編集/解析ツールの**WaveFormer Proの全機能も標準で装備**しています．WaveFormer Proの機能については別に用意した資料をご覧ください．

開発元　SynaptiCAD, Inc.
販売元　CQ出版株式会社

最新情報はCQ出版社のホームページをご覧ください．
規模限定プログラムやチュートリアルを入手できます．
http://www.cqpub.co.jp/eda/edamenu.htm