

SILOS III Tutorial

SILOS III Tutorial

Rev 97.1xx



SILOS での FPGA と ASIC デザインのデバッグ

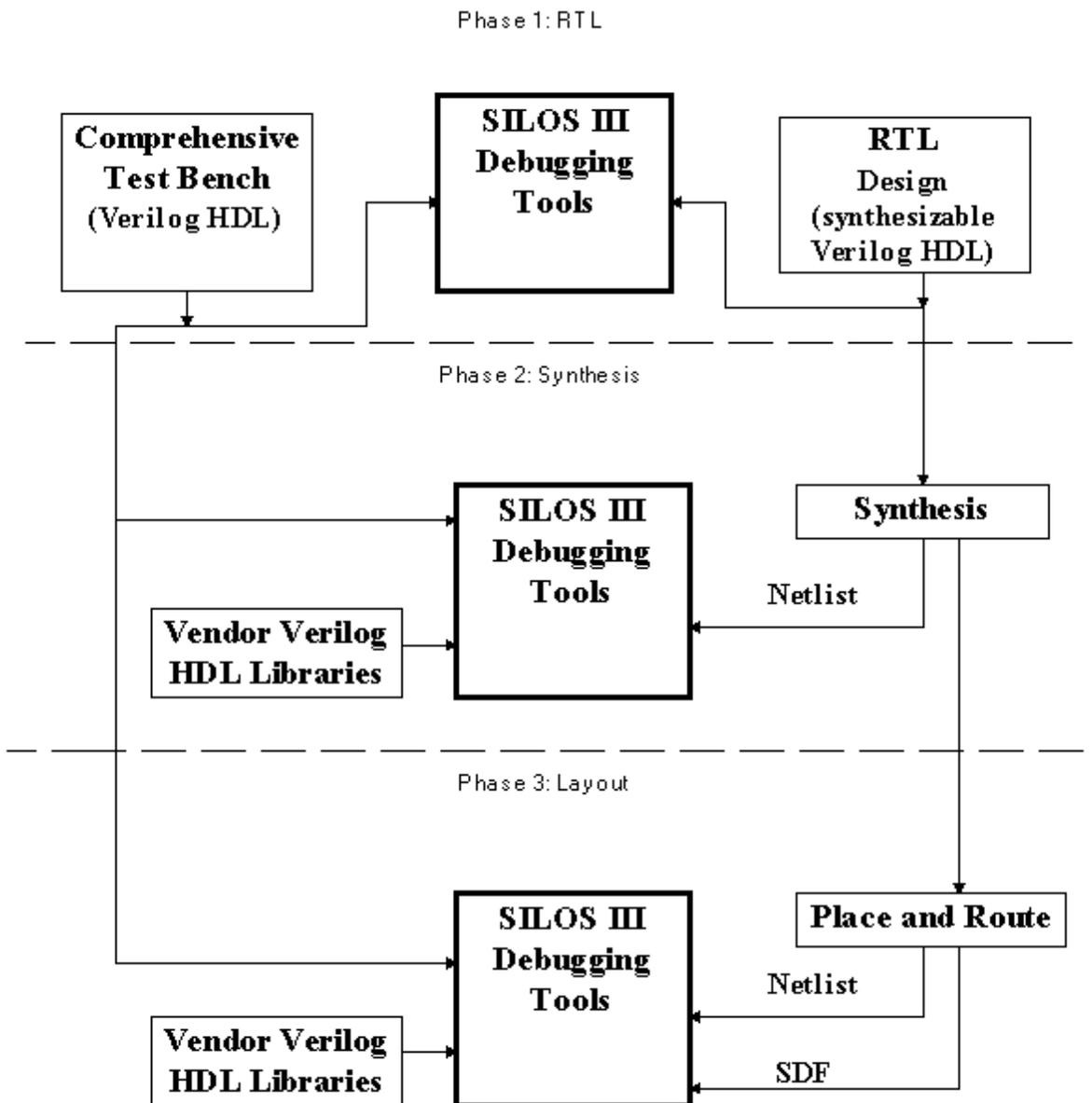
SILOS 論理シミュレーション・パッケージは、FPGA や ASIC の設計に低価格のデバッグ方法を提供します。SILOS は、Windows95 と WindowsNT、UNIX で使用可能です。

FPGA と ASIC をデザインする際、Register Transfer Level (RTL) で設計する場合は、合成ツールが必要になり、小規模のデザイン (3000 から 5000 ゲート) で回路図を使用する場合は、スキマティック・エントリー・ツールが必要になります。また、FPGA、ASIC ベンダーの開発キットと、SILOS シミュレーション・パッケージが必要になります。FPGA、ASIC デザインの Verilog HDL ライブラリーは、各ベンダーから入手することができます。

このチュートリアルでは、FPGA、ASIC デザインのデバッグに使用可能な SILOS の機能を説明します。

FPGA、ASIC を設計する際の一般的なデザインフローを以下に示します。

デザイン・フローの例



処理 1

・ Verilog HDL を使用して Register Transfer Level (RTL) 動作記述を作成し、デザインの構造をモデル化します。デザイン全体の複雑度を考慮すると、最初に Verilog HDL 動作デザインを作成して、デザインの一般的な機能性をモデル化したい場合が考えられます。(optional)

・ デザインとトータル・システムのインターフェースとなるテスト・ベンチを作成します。

・SILOS を使用してテストを実行し、RTL デザインが正しいかどうかをデバッグして検証します。RTL 記述とテストベンチは、一般的にテキスト・エディター、またはハイレベル・ツールを使用して作成します。多くの設計者の間で、プロジェクトのトータル時間の 60% から 70% がテストベンチの開発と RTL デザインのデバッグに費やされています。

処理 2

・RTL デザインを Verilog ネットリストに合成します。また、合成ツールでは、エリア、タイミングなどの制約ファイルが必要となります。

・合成ツールから Verilog HDL ネットリストを生成します。

・SILOS を使用してネットリスト、テストベンチ、ベンダーの Verilog HDL ライブラリーのシミュレーションを実行します。SILOS が供給するデバッグ・ツールを使用して、ネットリストをデバッグします。合成されたデザインが RTL デザインと同じ結果になることを確認します。

処理 3

・合成ツールからの出力をベンダーのツール・キットに入力し、デザインの配置配線を実行します。ベンダーのツール・キットでは、Verilog HDL ネットリストと SDF ファイルを生成し、SILOS で配置配線後のシミュレーションを行うための遅延をバックアノテートします。

・SILOS で RTL デザインのテストベンチを使用して、配線後のゲートレベル・デザインのシミュレーションとデバッグを行います。合成後、または配線後のデザインが期待通りに動作しない場合は、SILOS のトレース・バック機能を使用し、トポロジー（位相幾何学）を使用したトレース・バックを行って問題の原因を見つけることができます。

・ASIC デザインに SILOS を使用して、ASIC のシミュレーション作成に使用する foundry で、テスターの故障グレードのテスト・ベクターに対する故障シミュレーションを実行したい場合は、SIMUCAD USER'S MANUAL の 2-52 頁 "Fault Simulation to Increase Foundry Yields" を参照してください。

チュートリアル の操作について

チュートリアルで実行するフローを使用する場合は、トピックに示される順序にしたがってください。いくつかのトピックをスキップすると、各トピックに必要な処理が実行されない場合があります。

チュートリアル の例に使用されるファイルは、SILOS のインストール・ディレクトリーのサブディレクトリー "examples" に格納されています。"examples" 中のチュートリアル例以外のものについては、インストール・ディレクトリーの README ファイルを参照して下さい。

SILOS でデバッグを行う方法

このトピックでは、以下の内容について説明します。

- プロジェクトの設定
- SILOS の起動とオンライン・ヘルプのアクセス
- シミュレーションの実行と、Data Analyzer ウィンドウの波形表示
- ベクターのビット表示と、シンボル名を使用したベクター値の表示
- ASCII ベクターを設定して、シミュレーション進捗のテキスト時間行を作成する
- 信号をグループにまとめる
- Verilog HDL 式を使用した条件サーチと、サーチ条件の波形表示

1-1. SILOS シミュレーション環境 (SSE) の起動

Windows のスタート・メニューから SILOS を起動します。"SSE.EXE" のショートカットを作成した場合は、Silos Simulation Environment アイコンをダブルクリックして SILOS を起動します。

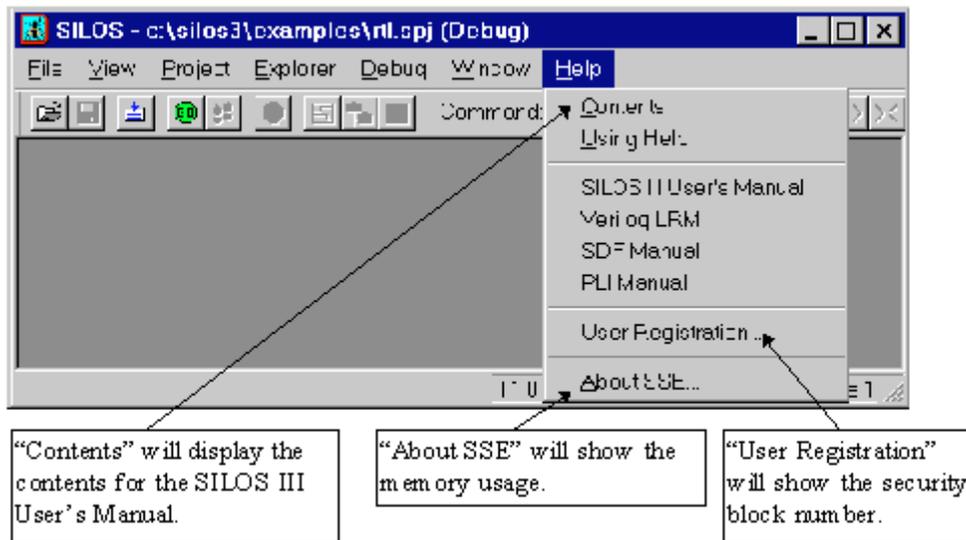
1-2. オンライン・ヘルプのアクセス

SILOS のオンライン・ヘルプをアクセスするには、以下の操作を行います。

- SILOS の Help メニューを選択します。

以下のマニュアルがオンライン・ヘルプ・ファイルとして供給されています。

- SILOS User's Manual (Help/Contents から選択できます。)
- Open Verilog International(OVI)Verilog Language Reference Manual version 1.0
- OVI Standard Delay Format (SDF) Manual version 2.0
- OVI Programming Language Interface (PLI) Manual version 1.0



Help で Project/New の使用法をサーチする例を以下に示します。

- "Help" メニューをハイライトして "Contents" メニューを選択します。
- on-help ウィンドウが表示されたら、"Menus Chapter" までスクロールします。

"Project" メニュー - を選択し、次に "New Menu Selection" を選択します。

SILOS のメモリー使用について確認するには、以下の操作を行います。

- "Help/About SSE" メニューを選択し、"About SILOS Simulation Environment" のダイアログ・ボックスを開きます。このボックスは、SSE のバージョンと "Silos Memory Allocated" を表示します。

1-3. プロジェクトの設定

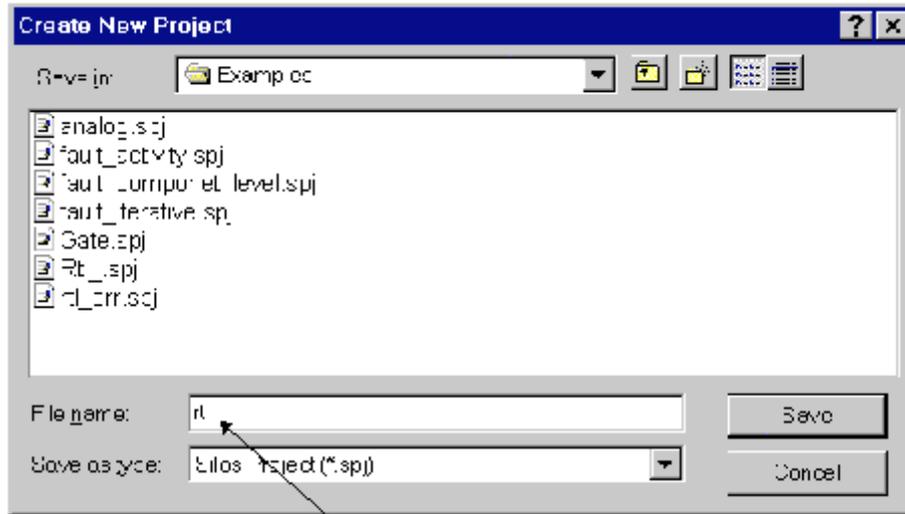
SILOS でデザインをシミュレーションするには、各デザインのプロジェクトを作成します。プロジェクトは以下の機能を提供します。

- 入力ファイルとライブラリー・ファイルを体系化する
- キーワード定義などのプロジェクト設定を行って、機能を保存する。
- Data Analyzer のグループ表示など、インタラクティブな設定を保存する。

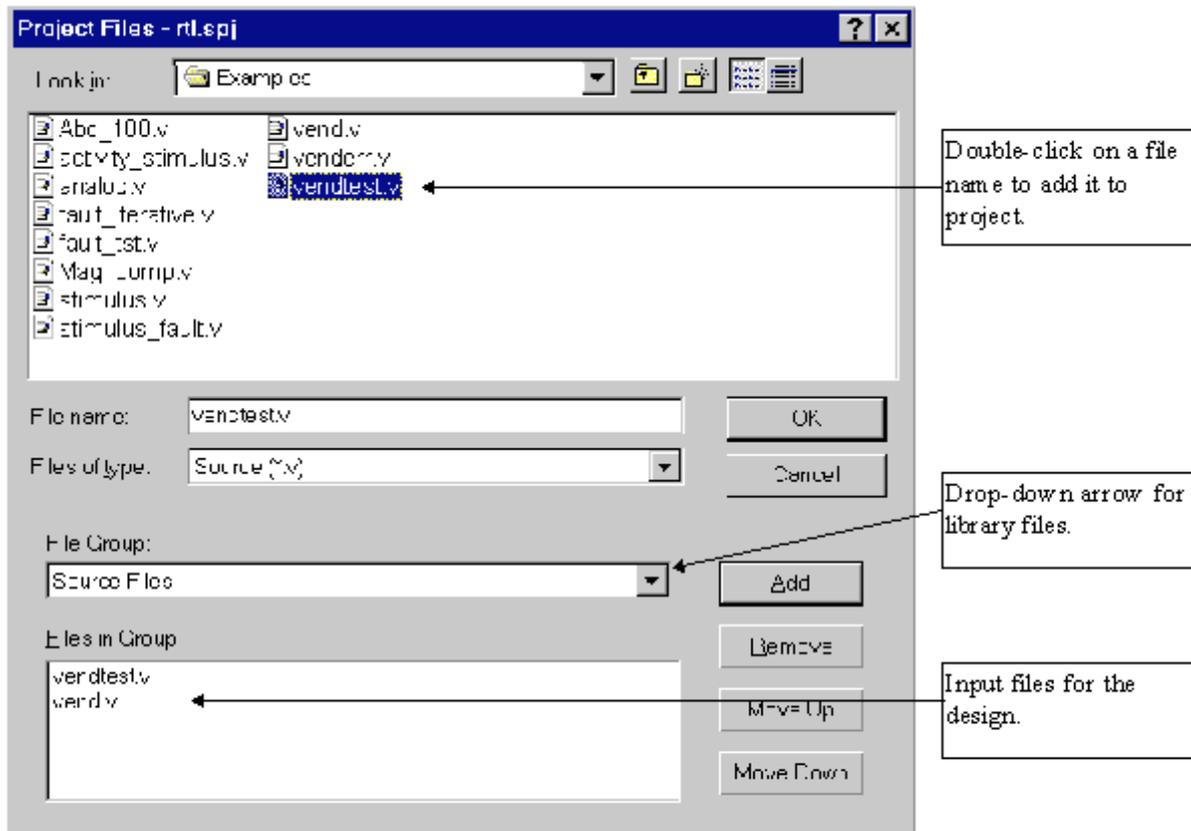
以下のサンプル回路は、新聞の自動販売機の RTL 記述です。

この例で使用される回路は、README ファイルに示されています。

この章では、新しいプロジェクトを設定して、ソース・ファイルを入力する方法を説明します。ライブラリー・ファイルを定義する方法は、SIMUCAD USER'S MANUAL の 3-11 頁 "File Menu Selection" を参照してください。



- "Project/New" メニューを選択し、"Create New Project" ダイアログを開きます。
- "Directories" のリストボックスで、インストール・ディレクトリーのサブディレクトリーにある "examples" をクリックします。
- "File Name" ボックスに "rtl" と入力し、" 保存 " ボタンをクリックして、ダイアログ・ボックスを閉じます。プロジェクト名に拡張子を指定しない場合、SSE はプロジェクト名に自動的に拡張子 .spj を追加します。"Project File" ダイアログ・ボックスが自動的に開き、そのプロジェクトの入力ファイルを設定することができます。

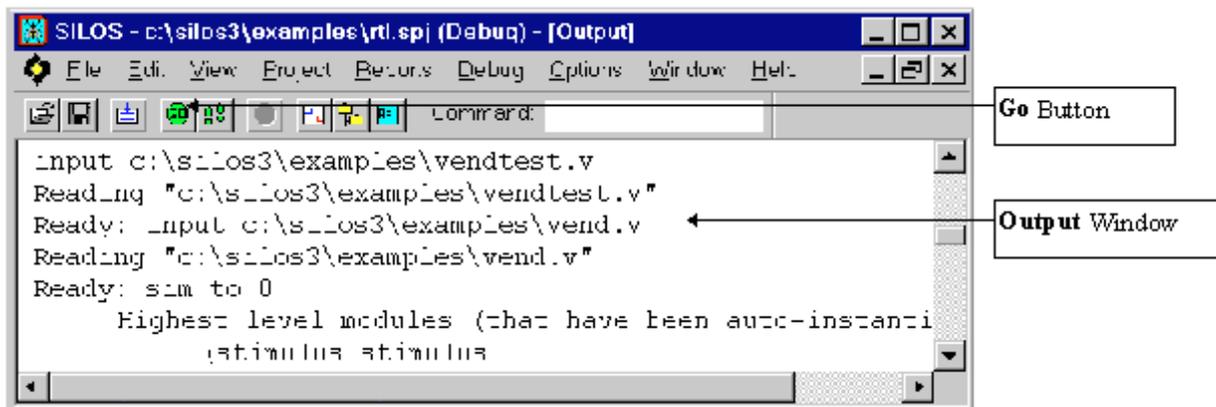


プロジェクト "rtl.spj" に入力ファイルを指定するには、以下の操作を行います。

- "File Name" リストボックスで "vendtest.v" をダブルクリックして、"File in Group" リストボックスにこのファイルを追加します。
- 次に "vend.v" を選択し、"Add" ボタンをクリックして、"File in Group" リストボックスにこのファイルを追加して、"OK" ボタンをクリックします。コンパイラ指示の include を使用して、ソース・ファイルを指定することもできます。(Verilog HDL Reference のオンライン・ヘルプ参照)

1-4. シミュレーションの実行

- "Project" メニューを選択し "Debug" モードがアクティブになっていることを確認してください。("Debug Mode" の左側に黒いドットが表示されている状態)
- SSE ツールバーの各ボタンの上にマウス・カーソルを合わせると、ボタンのテキスト・ラベルが表示されます。



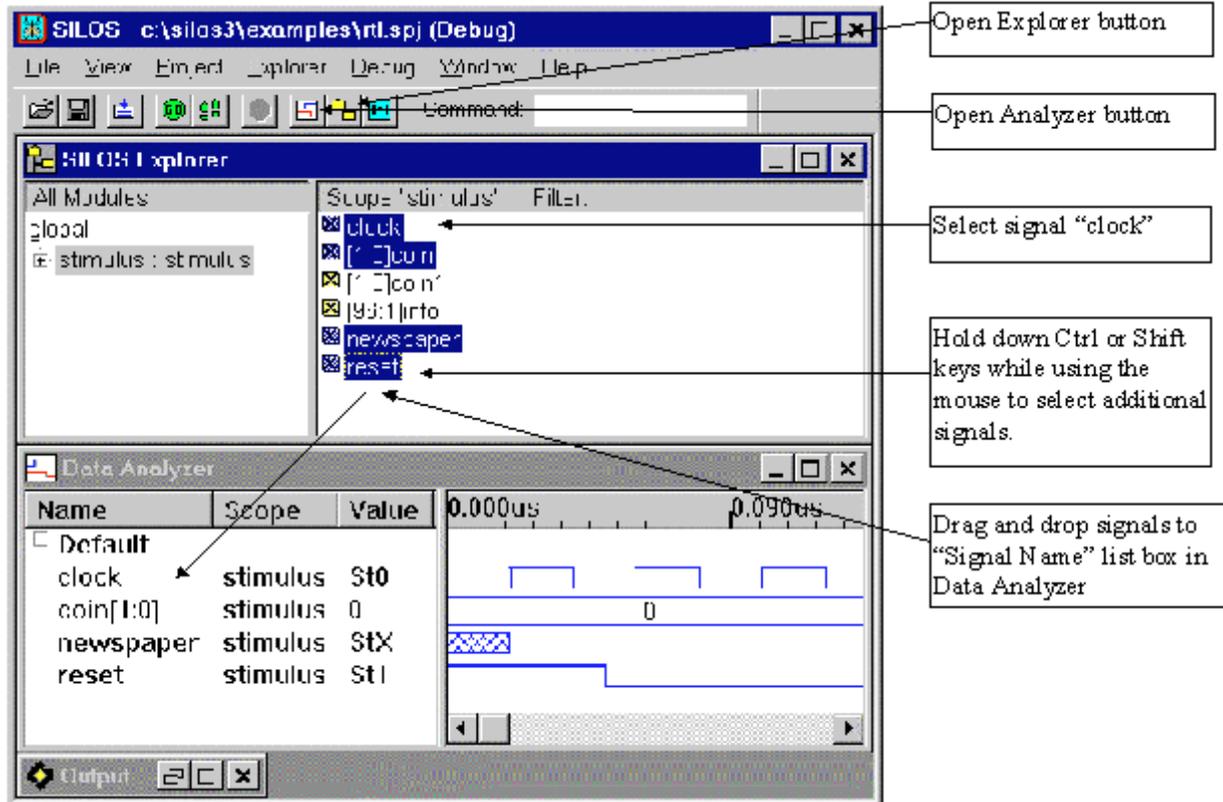
- ツールバーの "Go" ボタンをクリックして、入力ファイルをロードし、論理シミュレーションを実行します。論理シミュレーションは、ファイル "vendtest.v" のシステム・タスク \$finish の記述部分まで実行されます。また、"Debug Go" メニューを使用して論理シミュレーションを実行することもできます。詳細については、SIMICAD USER'S MANUAL の 3-27 頁 "Go Menu Selection" を参照してください。

1-5. Explorer ウィンドウの使用法

SSE は、Explorer ウィンドウ (と HDL ソース・ウィンドウ) から Data Analyzer ウィンドウに変数名をドラッグ&ドロップする機能を持っています。この操作を以下に説明します。

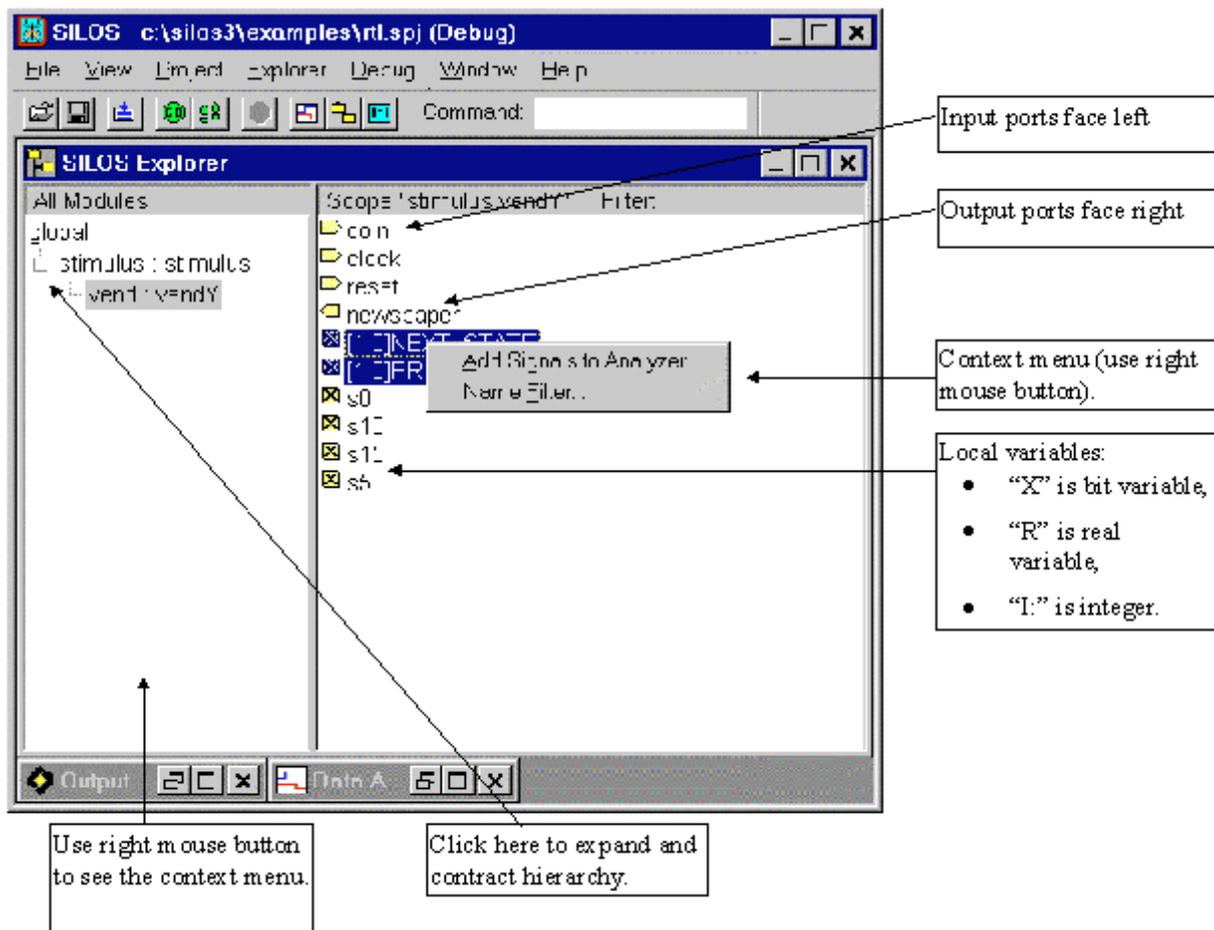
- "Open Analyzer" ボタンをクリックして Data Analyzer ウィンドウを開きます。Data Analyzer ウィンドウに関する詳細は、SIMUCAD USER'S MANUAL の 3-35 頁 "Data Analyzer Menu Selection" を参照してください。
- 次に、ツールバーの "Open Explorer" ボタンをクリックして Explorer ウィンドウを開きます。Window/Tile メニューを使用して 2 つのウィンドウを見やすいように整列させます。最初に Output ウィンドウを最小化しておいたり、SILOS の View/List メニューを選択して変数を表示することもできます。Explorer ウィンドウで View/Detail を選択すると、Explorer ウィンドウが表示する信号は、シミュレーション中 Explorer ウィンドウの "Keep" 列に保存されます。

Explorer はマルチスレッドになっているため、Explorer の起動中でも Explorer や、シミュレーションを使用することができます。



Explorer ウィンドウの左側のボックスのインスタンス名 "stimulus" は、デザインのトップ・レベル・モジュールです。インスタンス "stimulus" の信号名を選択するには、以下の操作を行います。

- マウスでインスタンス "stimulus" を選択します。
- 最初の名前 "clock" をクリックします。次に Ctrl キーを押しながら、その他の信号名 "coin[1:0]", "newspaper", "reset" をクリックして、インスタンス "stimulus" の変数名をハイライトします。
- 信号名をドラッグ&ドロップするには、"Explorer" ウィンドウでハイライトされた変数名をマウスでドラッグし、Data Analyzer ウィンドウのリストボックスにドロップします。各信号の波形が表示されます。
- Data Analyzer に信号を追加するには、Explorer の文脈メニュー "Add Signals to Analyzer" を使用することもできます。

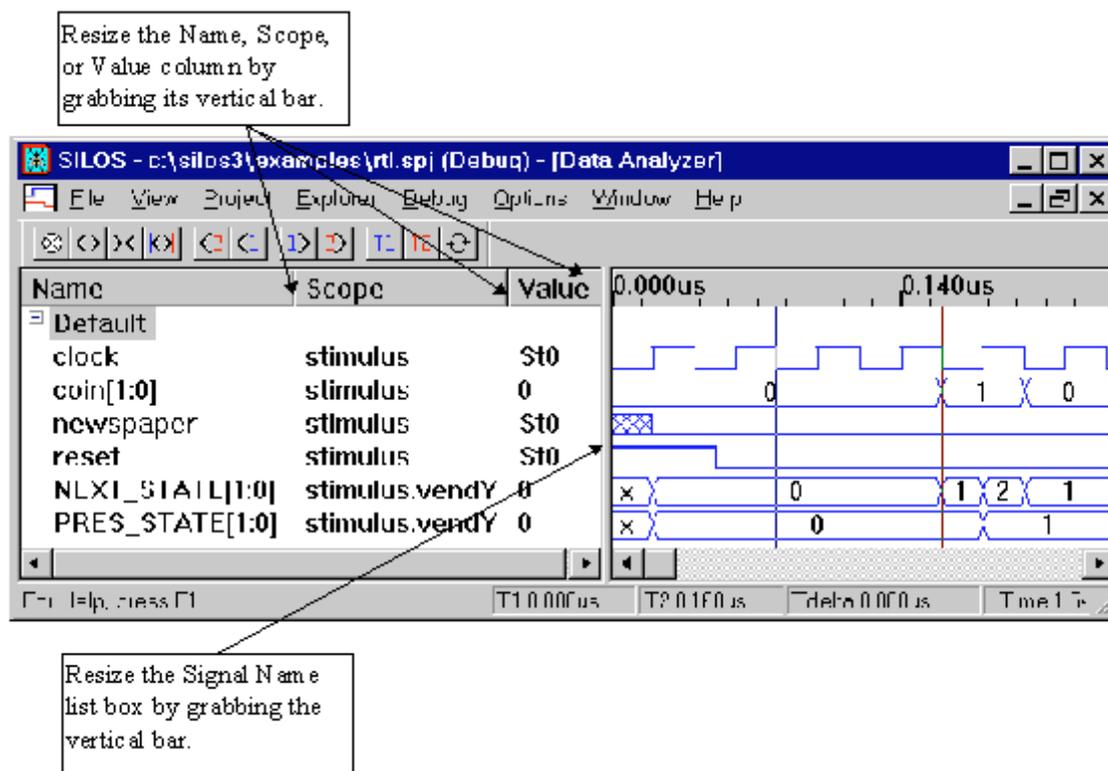


その階層のサブモジュールから信号名をドラッグ&ドロップするには、以下の操作を行います。

- インスタンス "stimulus" のとなりにある "+" 記号をクリックして、その階層のサブモジュールを表示します。
- 次にインスタンス "vendY" を選択すると、Explorer ウィンドウの右側に変数名が表示されます。変数の左に示されるシンボルは、ローカル変数からインスタンス（論理変数は×印のボックス、実数の変数は "R"、整数の変数は "I"）までのポート変数の種類（パッド・シンボル）を識別しています。入力ポートは右方向のパッド・シンボルで、出力ポートは左方向のパッド・シンボル、インアウト・ポートは、双方向のパッド・シンボルになっています。
- インスタンス "vendY" の信号 "NEXT_STATE" と "PRES_STATE" を選択し、Data Analyzer ウィンドウの "Signal Name" リストボックスにドラッグ&ドロップします。

1-6. Signal Name リストボックスのサイズ変更

Signal Name リストボックスのサイズを変更するには、Signal Name リストボックスの垂直エッジをマウスでドラッグして左右にスライドさせます。Name, Scope, Value の列も同様に各列間の区切りをスライドさせてサイズを変更できます。

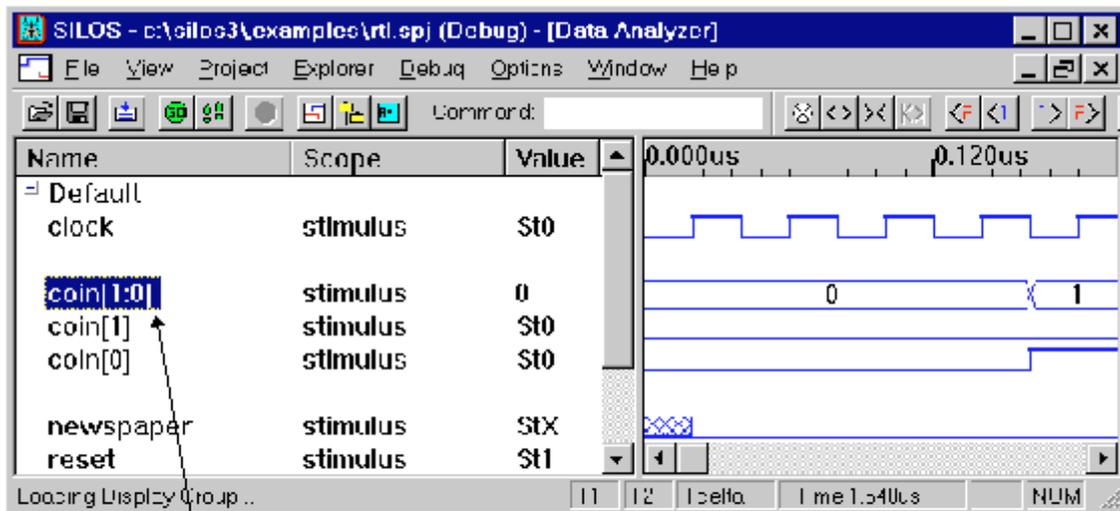


1-7. ベクターの展開 / 各ビットの非表示

SILOS は、デザインを再度シミュレーションすることなく、ベクターを展開して各ビットを参照することができます。したがって、デバッグ中にシミュレーションの実行を待つ必要がなく、時間を節約できます。ベクターを各ビットに展開するには、以下の操作を行います。

- ベクター名 "coin[1:0]" をダブルクリックします。
- 再度そのベクター名をダブルクリックすると、ベクター信号のビットは非表示になります。

信号を見やすく表示するには、Signal Name リストボックスの文脈メニューの "Add Blank Line" メニューを使用して空白行を追加します。Data Analyzer の文脈メニューにアクセスする方法に関する詳細は、SIMICAD USER'S MANUAL の 3-35 頁 "Data Analyzer Menu Selection" と 3-53 頁 "Data Analyzer Signal Name List Box" を参照してください。

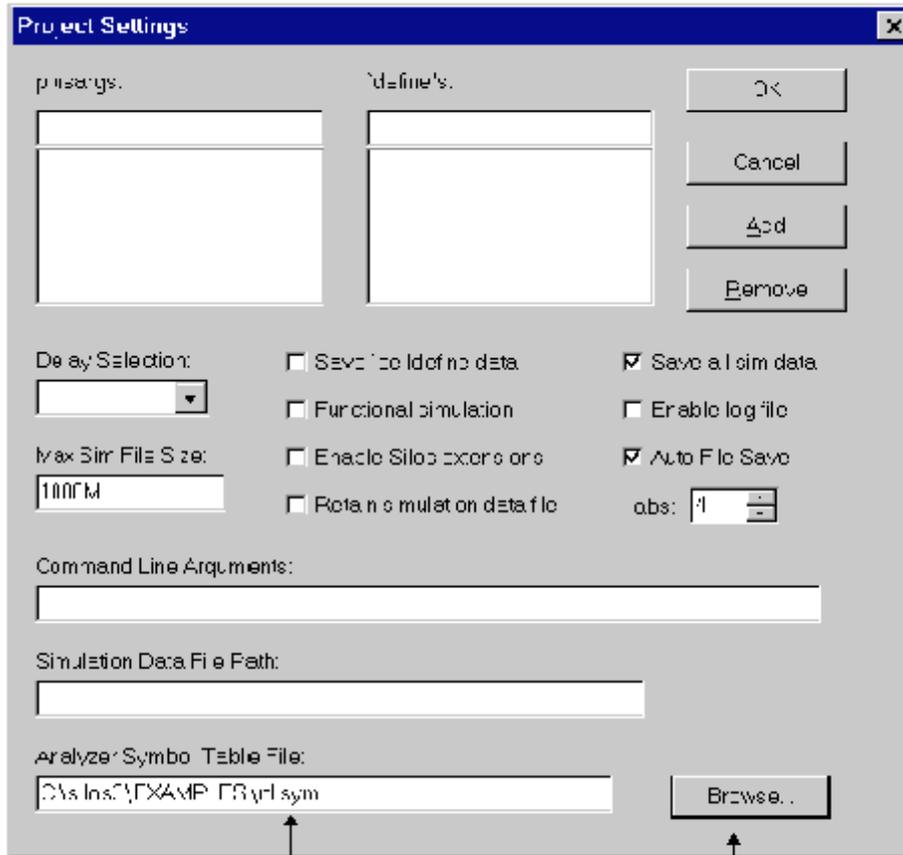


Double click on vector
"coin[1:0]" to expand and
hide the bits.

1-8. ベクター値をシンボル名で表示する

ステートマシンでは、ベクター値を意味を成すシンボル名で表示する便利な機能を使用できます。例えば、ベクター coin[1:0]、NEXT_STATE[1:0]、PRES_STATE[1:0] のステートマシン値を表示するには以下の操作を行います。

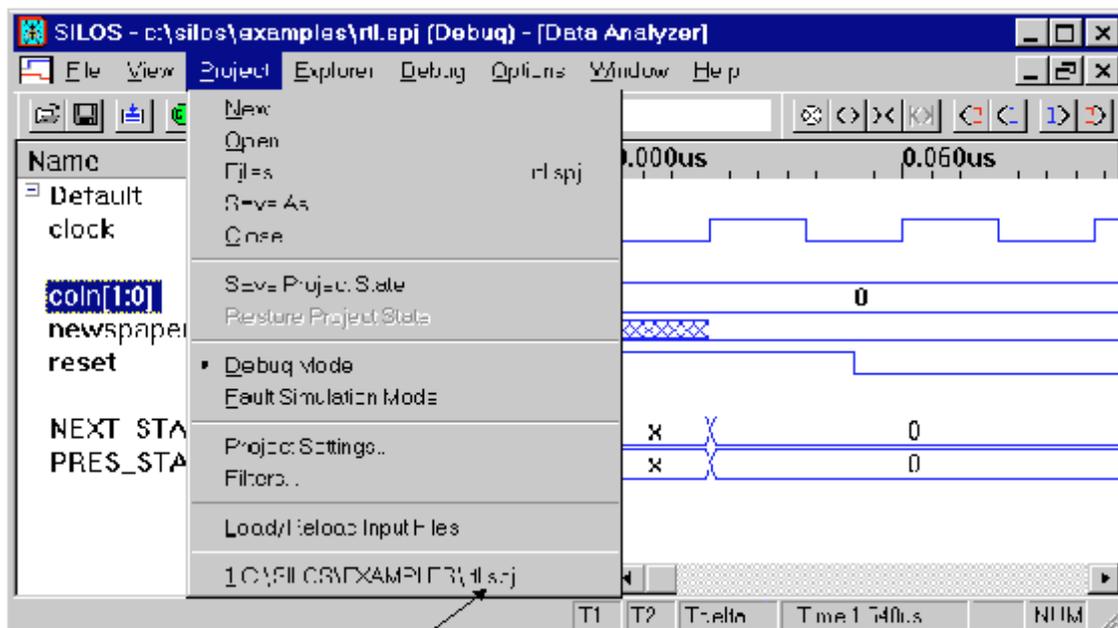
- "Project/Project Settings" メニューを選択し、"Project Settings" ダイアログ・ボックスを開きます。"Project Settings" ダイアログ・ボックスの Browse ボタンを使用して "Analyzer Symbol Table File" ボックスにファイル "silos3\examples\rtl.sym" を設定します。



File "rtl.sym" contains symbolic names for the state values for vectors.

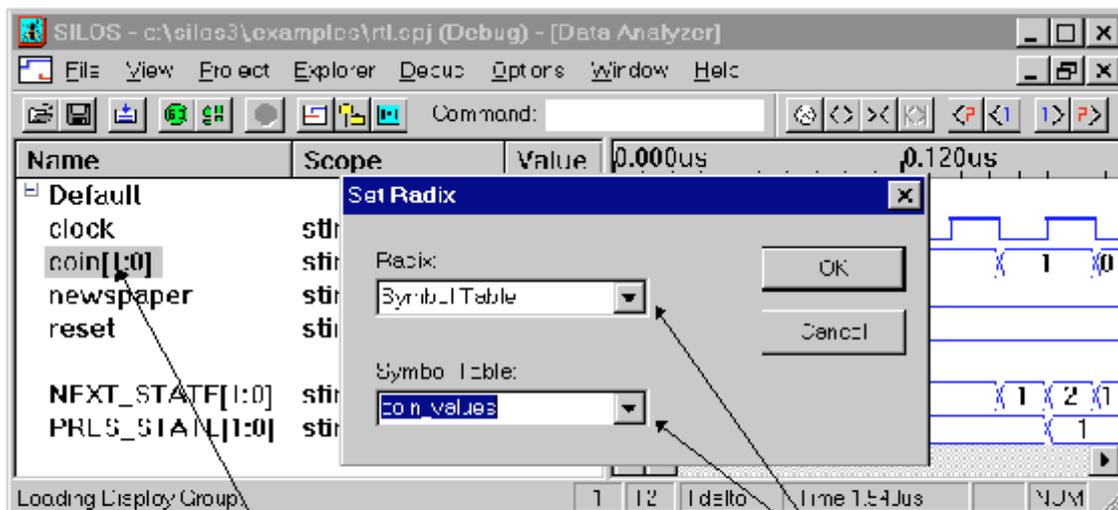
Browse button.

- "Project Settings" ダイアログ・ボックスで "OK" ボタンをクリックすると、変更を再度ロードする必要があるというメッセージが表示されます。今回は "Project Settings" ダイアログ・ボックスを開いたときに、メッセージを再度表示させたいので、"No" を選択します。
- "Project" メニューから "rtl.spj" プロジェクトを選択します。これにより、"rtl.spj" プロジェクトが再びロードされます。メッセージが表示されたら、Default 表示グループの変更を保存するため、"Yes" を選択します。



Select project "rtl.spj" so that the project will reload and the project settings will take effect.

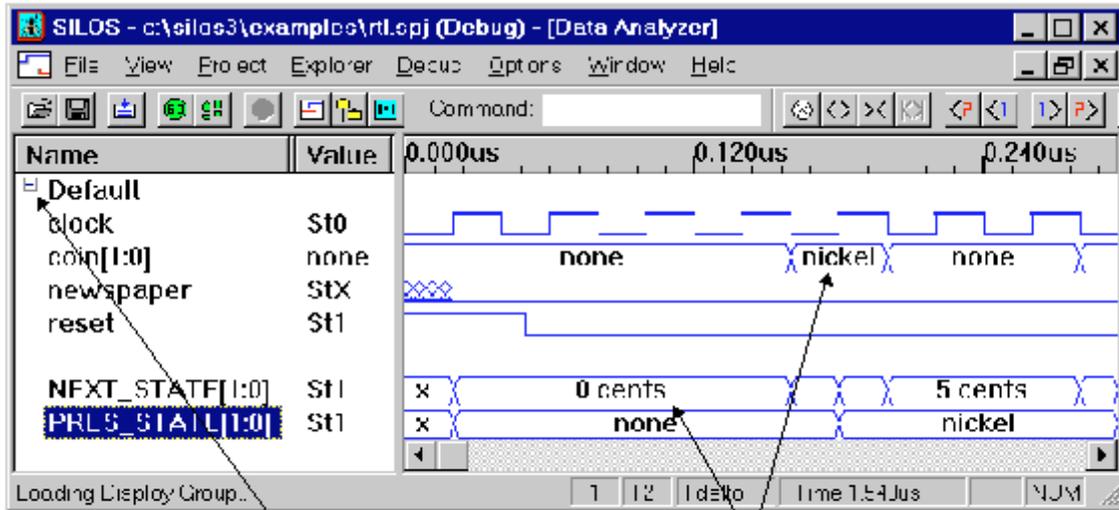
- "Go" ボタンをクリックすると、デザインのシミュレーションを実行し、Data Analyzer ウィンドウが再び開かれます。
- Signal Name リストボックスの文脈メニューを開くには、Data Analyzer の Signal Name リストボックスで信号 "coin[1:0]" の上でマウス右ボタンをクリックします。次に "Set Radix" をクリックし、"Set Radix" ダイアログ・ボックスを開きます。



Select "coin[1:0]". Click with right mouse button to see context menu. Then select "Set Radix".

Use the drop-down arrows to display Symbol Table and coin_values.

- Set Radix" ダイアログ・ボックスで "Radix" の下矢印をクリックして、"Symbol Table" を選択します。Set Radix" ダイアログ・ボックスの "Symbol Table" ボックスで下矢印をクリックし、"coin_value" を選択して "OK" ボタンをクリックします。



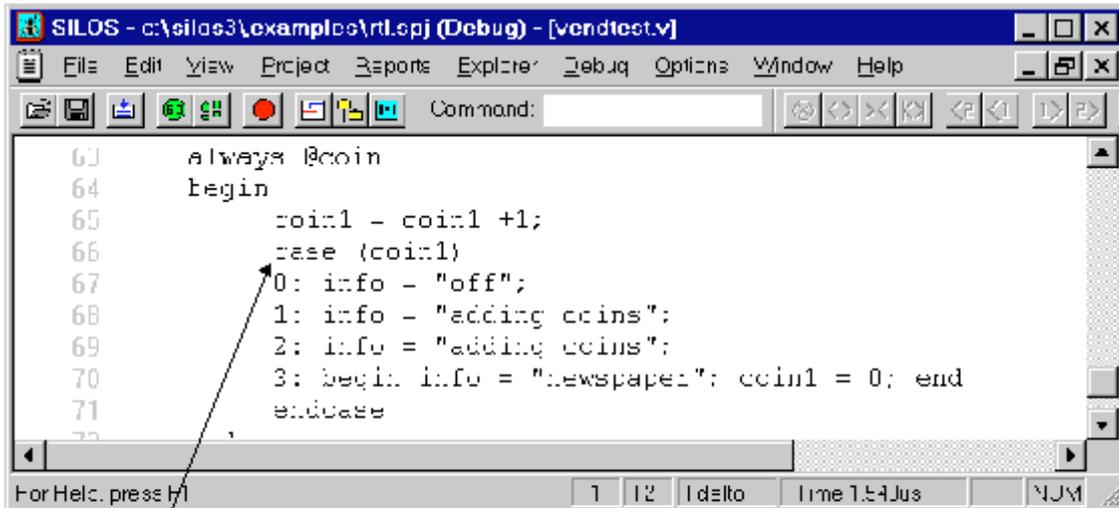
Make sure you click on the minus sign "-" to save the new radices.

Symbolic names for state values for vectors.

"Set Radix" メニューを選択して、NEXT_STATE[1:0] の基数に cents, PRES_STATE[1:0] の基数に coin_values を選択します。Default グループの "-" 記号をクリックして、新しい基数を保存します。

1-9. ASCII ベクターを使用して結果に注釈を付ける

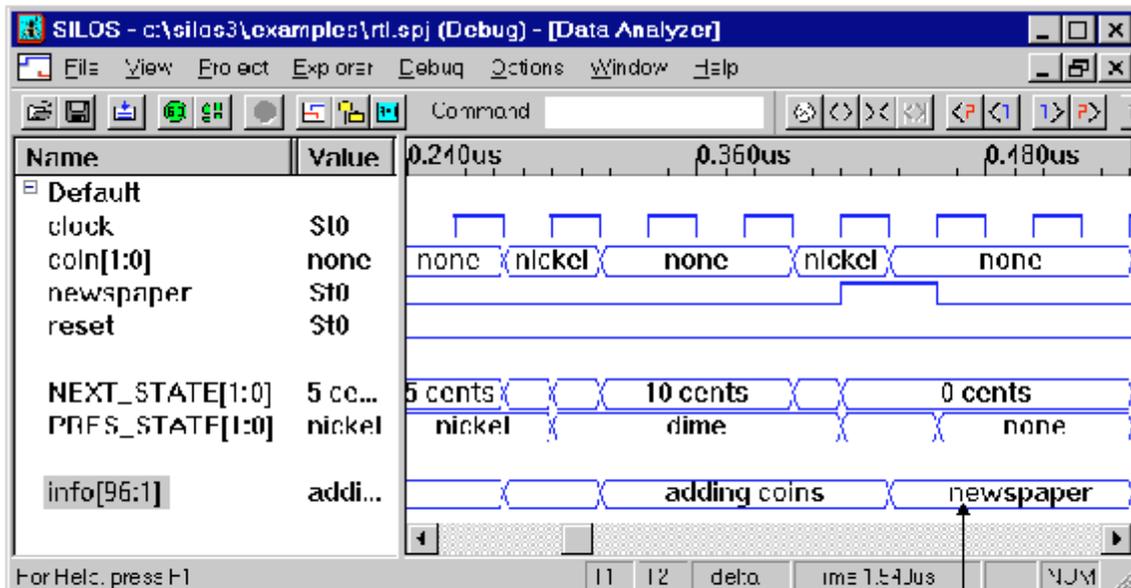
デザインをデバッグする際、ASCII テキストのベクターを表示して、シミュレーションがどのような処理を行っているかを説明すると大変便利な場合があります。この操作を以下に説明します。



```
63 always @coin
64 begin
65     coin1 = coin1 + 1;
66     case (coin1)
67     0: info = "off";
68     1: info = "adding coins";
69     2: info = "adding coins";
70     3: begin info = "newspaper"; coin1 = 0; end
71     endcase
72
```

case statement for ASCII vector "info".

- メイン・ツールバーの "Open File" ボタンを使用してファイル "vendtest.v" を開きます。
- ファイル "vendtest.v" を下へスクロールして "case" 文が使用されている部分 (66行目) を表示し、ベクター "info" に ASCII 文字列を設定します。



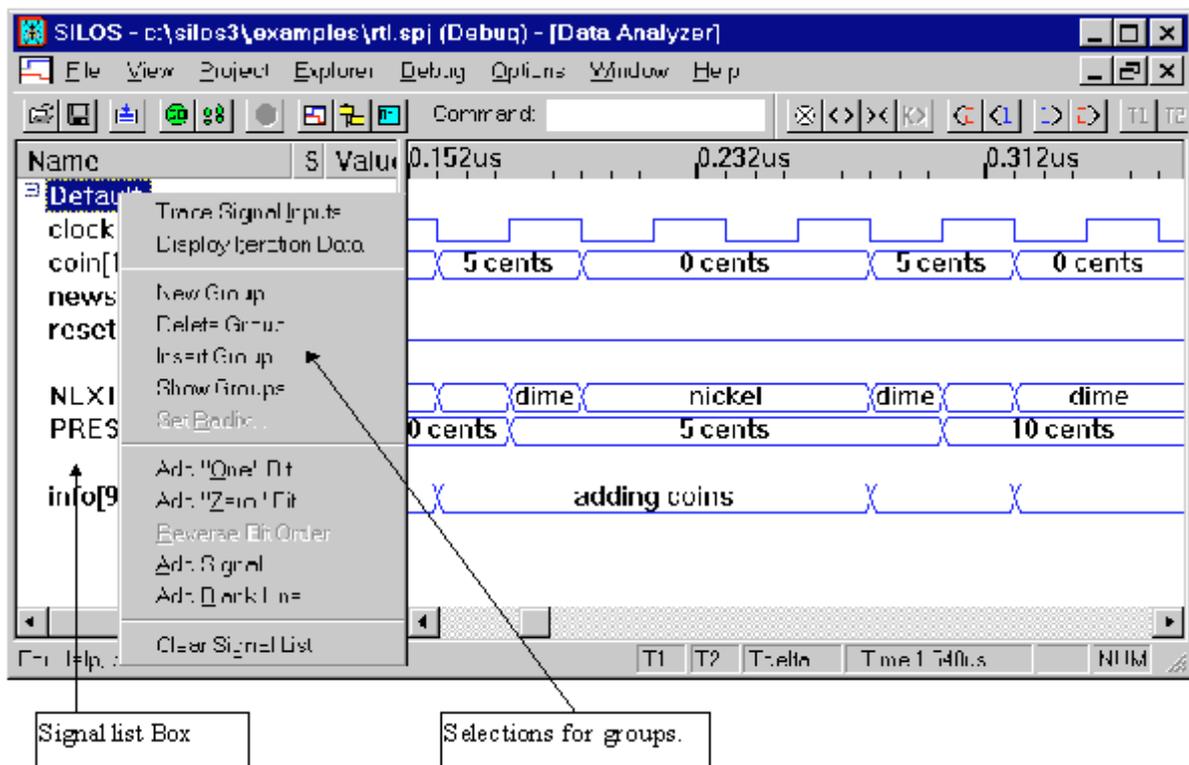
ASCII text describes the simulation.

- Explorer ウィンドウで、信号 "info" をマウス右ボタンでクリックし、文脈メニューを開きます。文脈メニューの "Add Signals to Analyzer" を選択し、Data Analyzer ウィンドウで Signal Name リストボックスの Default グループの一番下に信号 "stimulus.info" を追加します。
- 信号名 "info" の基数を ASCII に変更するには、Data Analyzer ウィンドウで Signal Name リストボックスの信号 "info[96:1]" でマウス右ボタンをクリックします。次に "Set Radix" を選択し、"Set Radix" ダイアログ・ボックスを開きます。
- "Set Radix" ダイアログ・ボックスで基数を ASCII に変更し、"OK" ボタンをクリックします。 "-" 記号をクリックして Default グループに新しい基数を保存します。

1-10 信号のグループ

グループを使用すると、Data Analyzer ウィンドウの信号を体系化するのに便利です。この機能は、信号の表示を体系化するだけでなく、Default グループが不注意に変更されたり、失われてしまった場合でも、信号リストは失われません。また、グループの表示は、デザインに精通していない技術者をサポートしたり、デザインを再利用する場合の記録保存にも便利です。

Signal Name リストボックスの文脈メニューではグループに対して以下の機能を選択できます。



- New Group : Signal Name リストボックスに新しいグループを追加します。
- Delete Group : グループを削除します。

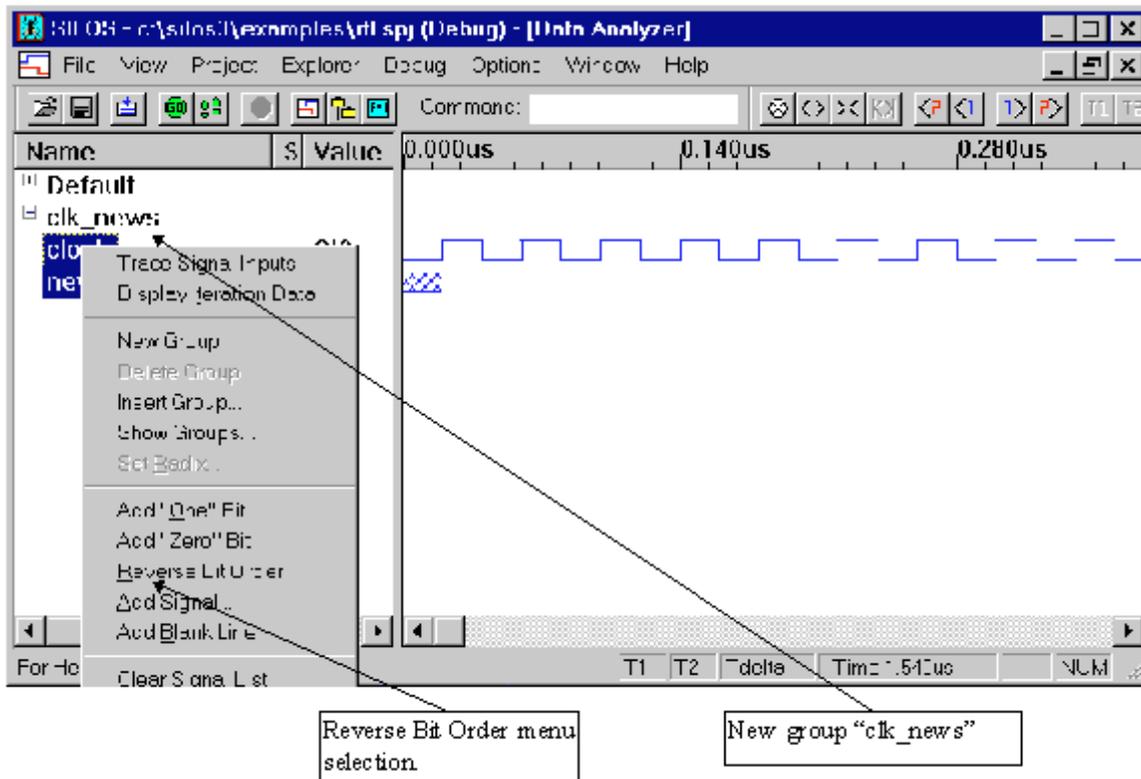
- Insert Group : "Add Group" ダイアログ・ボックスを開きます。このダイアログ・ボックスは、あるグループ内にグループを挿入します。挿入されたグループは、バスで表示され、ダブルクリックで展開したり、非表示にしたりすることができます。
- Show Group : "Select Signal Group" ダイアログ・ボックスを開きます。このダイアログ・ボックスを使用して、Data Analyzer に表示されたグループを選択します。

Data Analyzer ウィンドウを開くと、"Default" グループが表示されます。Default グループに追加された信号を保存するには、以下の操作を行います。

- Signal Name リストボックスで Default グループの左側の "-" 記号をクリックします。
- SILOS は、Default グループの変更を保存するかどうかを聞いてきます。(まだ保存していない場合) "Yes" ボタンをクリックします。(信号がまだ保存されていない場合)

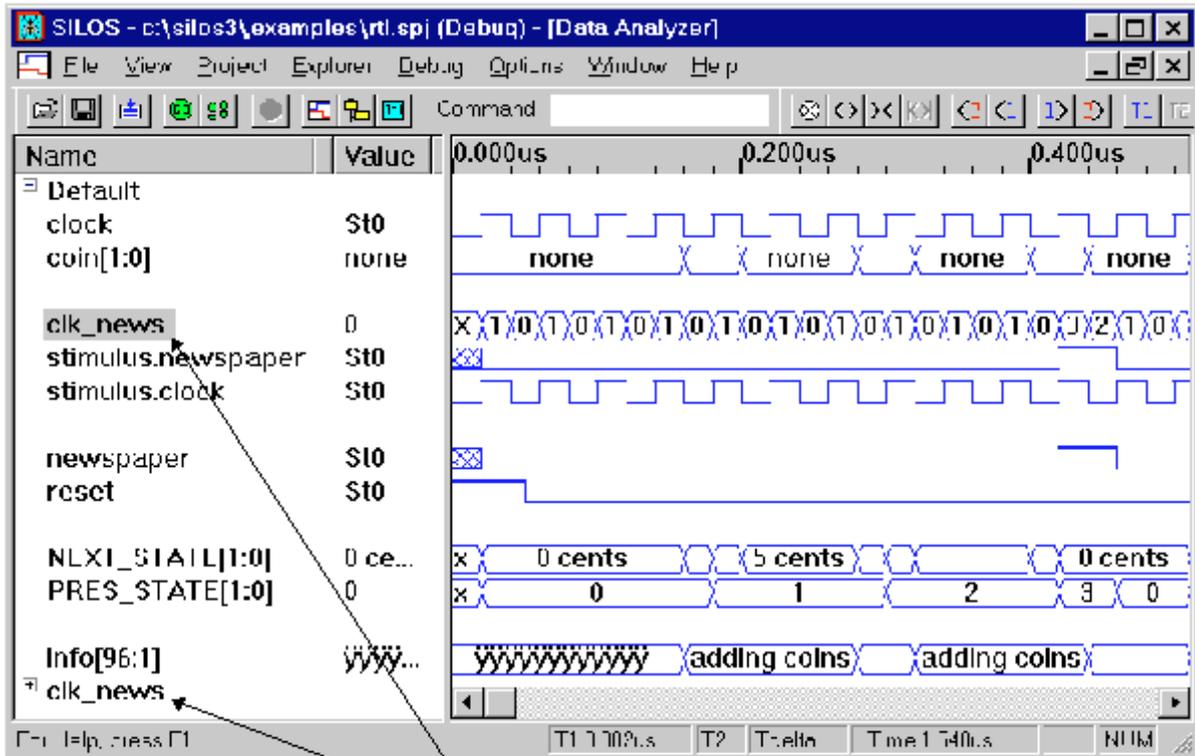
1-11. バス

グループを使用して、バスを表現することができます。信号 clock と newspaper を使用してバスを作成する場合、以下の操作を行います。



- Signal Name リストボックスでマウス右ボタンをクリックし、文脈メニューを開きます。
- 文脈メニューで "New Group" を選択します。Signal Name リストボックスに "New Group" というラベルのグループが追加されます。
- 新しいグループの名前を変更するには、"New Group" を再度クリックします。ラベルの入力状態にならない場合は "New Group" をもう一度クリックします。ラベル "New Group" を "clk_news" に変更します。

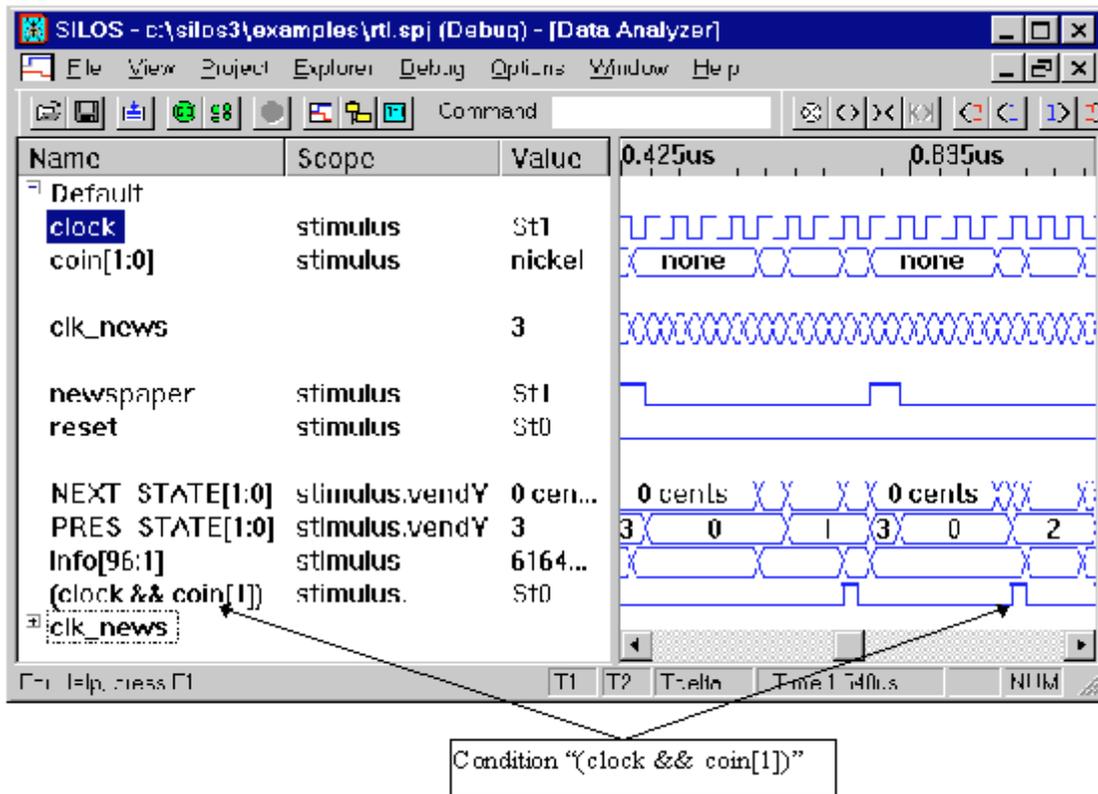
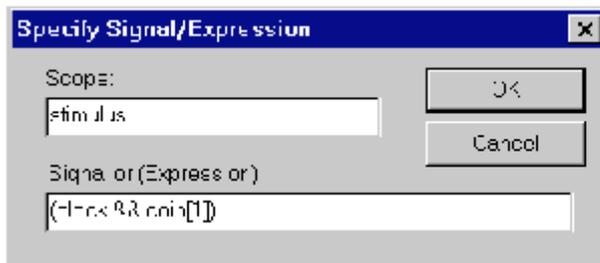
- Ctrl キーを押しながら、"clock" と "newspaper" を選択し、Ctrl キーを押しながら新しいグループ "clk_news" にマウスでドラッグ&ドロップします。
- グループ "clk_news" のビット配列を入れ替えるには、グループ "clk_news" の信号 "clock" と "newspaper" をハイライトします。Signal Name リストボックスの文脈メニューを開き、"Reverse Bit Order" を選択します。
- グループ "clk_news" を保存するには、グループ "clk_news" の左にある "-" 記号をクリックし、"Yes" を選択します。
- 次にグループ "clk_news" をグループ "Default" にドラッグ&ドロップします。また、"clk_news" の前後に空白行を追加することもできます。



Drag group "clk_news" into group "Default" so you can view "clk_news" as a user defined bus.

1-12. 条件サーチ

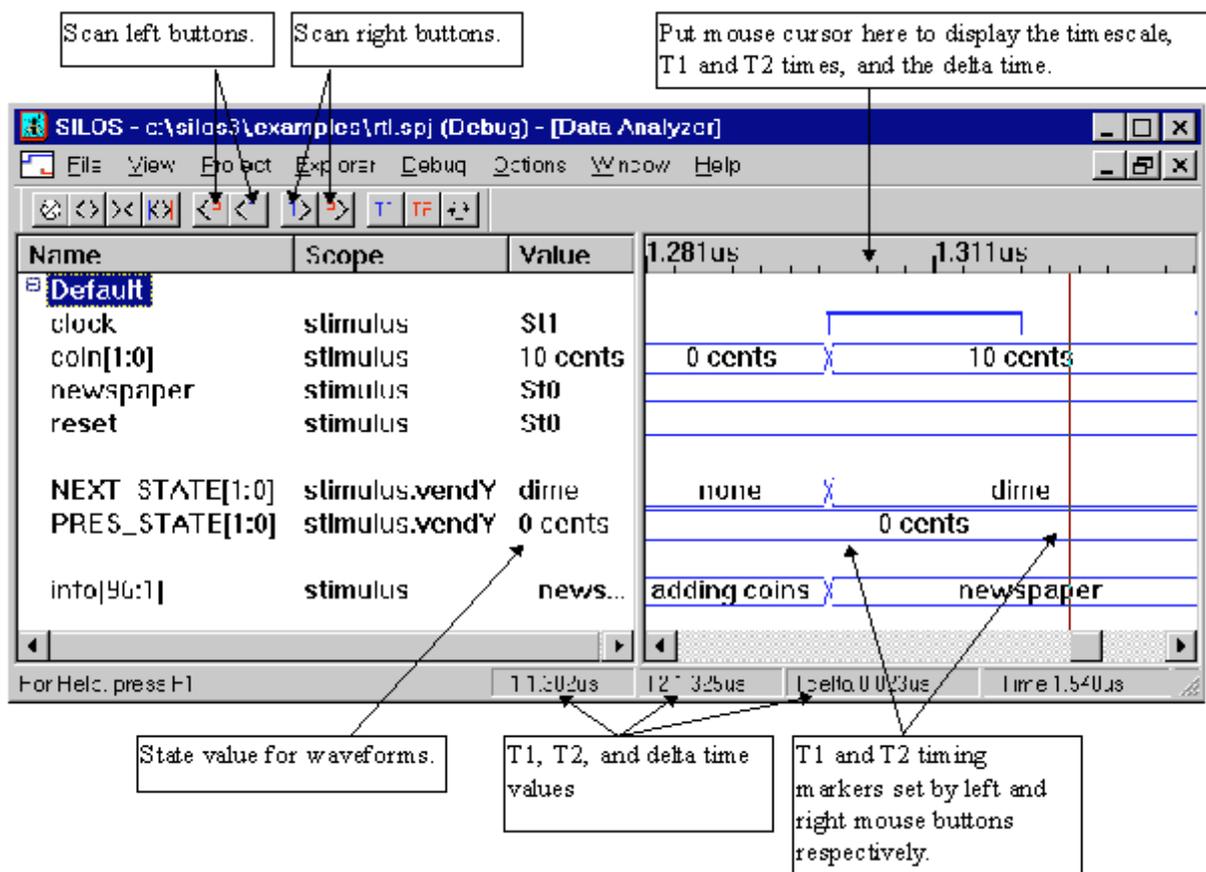
SILOS では、サーチ条件を波形で表示する特殊な機能を持っています。有効な Verilog HDL 式を使用してサーチ条件を作成し、波形で表示することができます。



例えば、"stimulus.clock" と "stimulus.coin[1]" の両方が真である場合のサーチ条件を作成するには、以下のように操作します。

- Signal Name リストボックスの文脈メニューを開きます。
- "Add Signal" メニューを選択し、"Specify Signal/Expression" ダイアログ・ボックスを開きます。
- "Scope" 編集ボックスで範囲を "stimulus" に設定します。
- "Signal/Expression" 編集ボックスに式 (clock&&coin[1]) を入力し、"OK" ボタンをクリックします。式の波形が Data Analyzer に表示されます。

1-13. エッジのスキャンとパンニング



波形表示ウィンドウでは、T1、T2の2つのタイミング・マーカーを設定することができます。T1、T2を使用して、エッジが生じた時間と、エッジ間のデルタ時間を表示することができます。タイミング・マーカーを設定するには、以下の操作を行います。

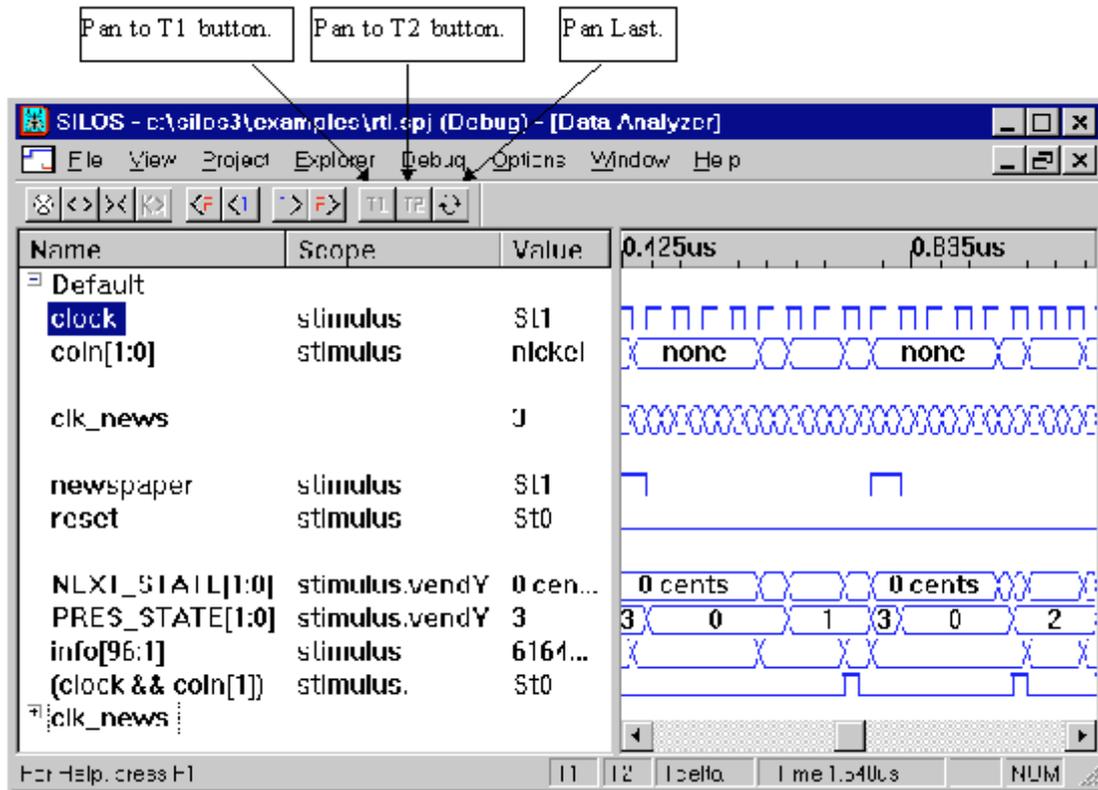
・波形表示エリアにマウス・カーソルを置きます。

・マウス左ボタンでT1（青い垂直ライン）、マウス右ボタンでT2（赤い垂直ライン）のタイミング・マーカーを設定します。マウス・カーソルをタイミング・マーカーのとなりの時間に置くと、マーカーの時間を示すテキスト・ボックスが表示されます。グレイの時間行エリアにマウス・カーソルを置くと、現在のタイムスケール、T1、T2の時間値、T1とT2間のデルタ時間を表示するテキスト・ボックスが表示されます。各波形のT1タイミング・マーカーのステート値は、Data AnalyzerでSignal Nameリストボックスの"Value"列に表示されます。

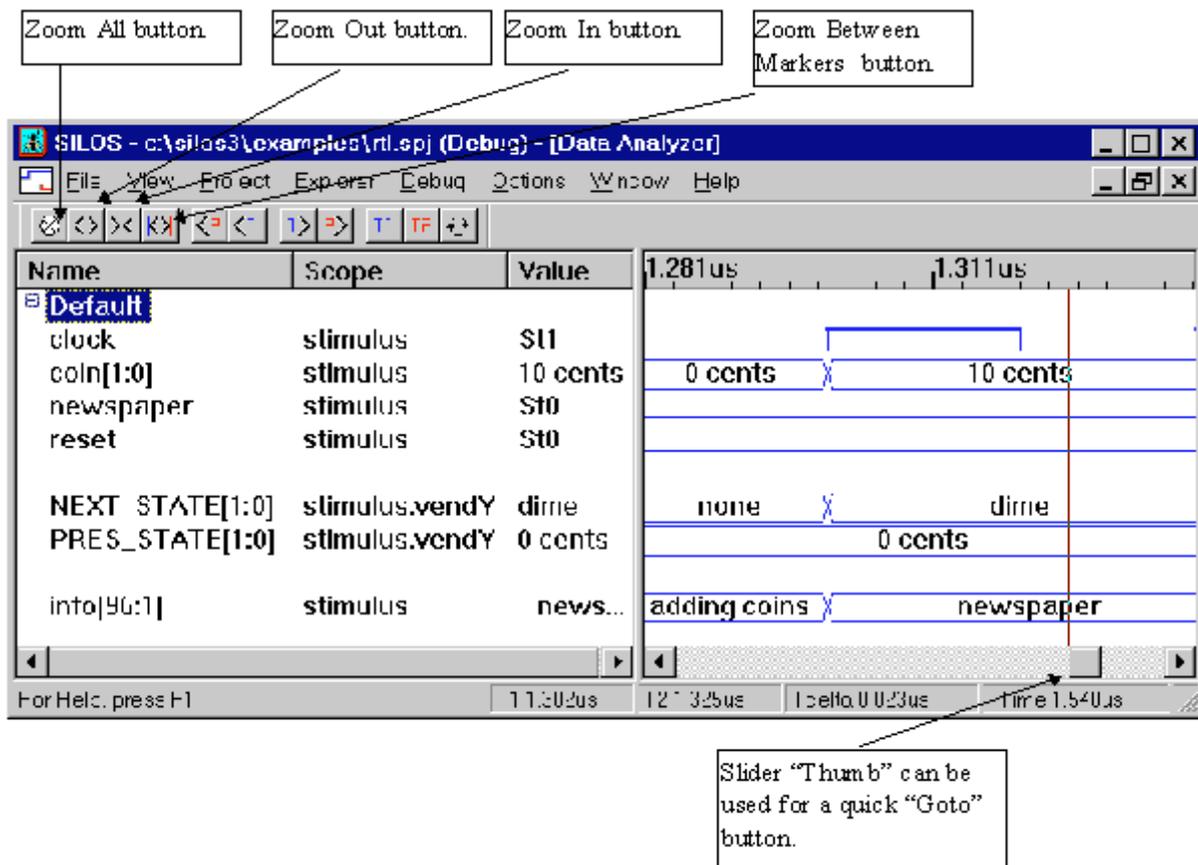
Data AnalyzerのT1、T2タイミング・マーカーを設定している場合、"Option/Snap to Edge"メニューの選択がアクティブになっていると、T1、T2は最も近いエッジにスナップします。タイミング・マーカーを設定している場合に、Shiftキーを押しながら、"Snap to Edge"を一時的にトグルすると、解除されます。

信号エッジの時間値とステート値は、エッジをスキャンして表示することができます。この操作を行うには、Signal Nameリストボックスの条件式"(clock&&coin[1])"をハイライトしてから、ツールバーの"1>"ボタンをクリックしてT1マーカーを右方向にスキャンします。"Value"列の各波形のステート値は、T1マーカーをエッジ毎にスキャンするたびに変わります。T2マーカーは、Value列のステート値では機能しません。

Analyzer ツールバーの "Pan to T1"、"Pan to T2"、"Pan Last" ボタンは、T1、T2 タイミング・マーカを中心に表示を切り替えたり、前の表示に戻したりします。



1-14. Zoom ボタン

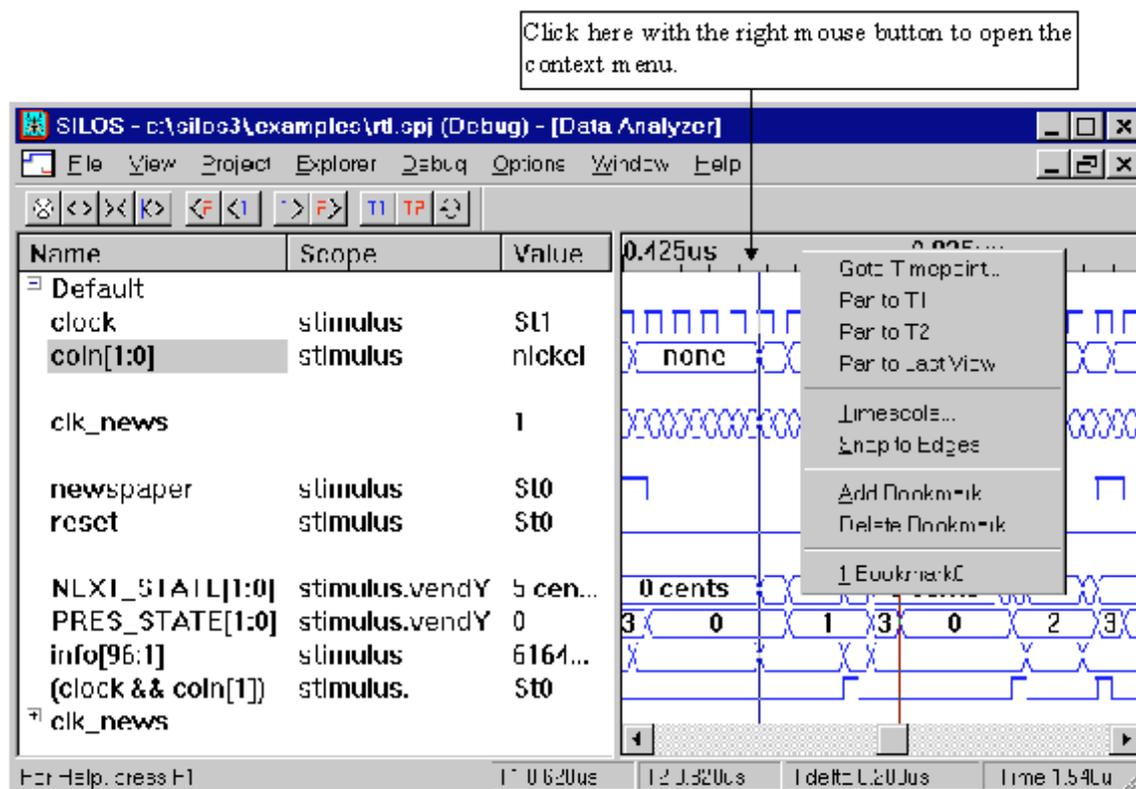


ツールバーの zoom ボタンを使用して、拡大表示 (zoom-in)、縮小表示 (zoom-out)、全体表示 (zoom-all)、マーカー間の表示ができます。

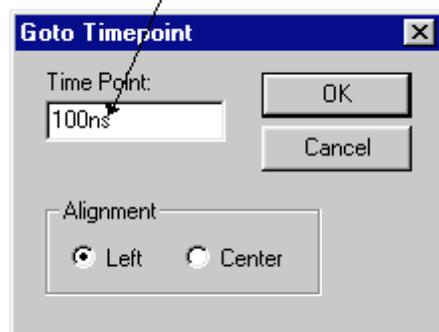
zoom 機能について以下に説明します。

- ツールバーの zoom-all ボタンをクリックすると、シミュレーション範囲全体を表示します。キーボードから "Esc" キーを押すと、zoom-all の表示を止めることができます。
- 次に、T1 と T2 のタイミング・マーカーをシミュレーション範囲の終わり付近に設定します。
- ツールバーの "+" ボタンを使用すると、タイミング・マーカー間を拡大表示します。
- ツールバーの "-" ボタンを使用すると、2つのマーカーを基準に縮小表示を行い、("P") ボタンは2つのマーカーを基準に拡大表示を行いません。
- 波形表示のスライダー "thumb" ボタンは、"go to" ボタンとして使用することができます。マウスを使用してスライダー "thumb" を 1us (マイクロ秒) まで移動します。スライダー "thumb" を移動する際、新しい時間値が水平の時間軸に沿って表示されます。

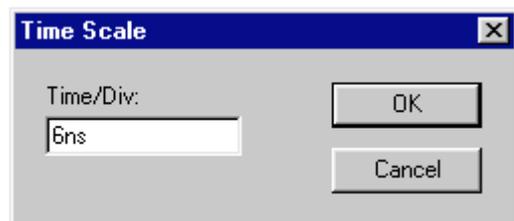
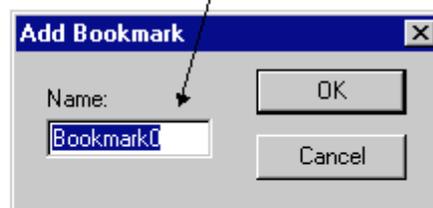
1-15. Bookmark, Timescale, Goto Timepoint



Time value can be specified in units such as nano seconds (ns).



Bookmarks allow you to jump between views that display different timescales and time values.



Data Analyzer ウィンドウの文脈メニューには、"Goto Timepoint"、"Timescale"、"Add Bookmark" のオプションがあります。文脈メニューを実行するには、Waveform Display ウィンドウのグレイの時間行エリア部分をマウス右ボタンでクリックします。文脈メニューのアイテムを選択するには、マウス左ボタンを使用します。

"Goto Timepoint" メニューを選択すると、"Goto Timepoint" ダイアログ・ボックスが開き、指定したタイム・ポイントに移動します。指定したタイム・ポイントは、Waveform Display ウィンドウの左、または中心位置に表示されます。

"Timescale" メニューを選択して、Time Scale ダイアログ・ボックスを開きます。現在の時間は、SSE 表示の最下行に表示されます。タイム・スケールを定義すると、600ns (ナノセカンド)、1000ps (ピコセカンド) など、すべての標準単位を使用することができます。

"Add Bookmark" メニューを選択すると、現在時間とタイム・スケールの解析を行なうために、Waveform Display ウィンドウの中心に垂直マーカをおきます。"Add Bookmark" ダイアログ・ボックスをオープンした後、"Bookmark1"、"Bookmark2" などのデフォルトブックマーカーが表示されます。ブックマーカーには、あらゆる文字列を定義することができます。設定したブックマークは、文脈メニューの最下行に表示されます。

文脈メニューの "Add Bookmark" メニューを選択して、"Add Bookmark" ダイアログ・ボックスを開きます。

"Bookmark0" を設定するには、"OK" ボタンをクリックします。

次に、文脈メニューを開き、"Bookmark0" を選択すると前の時間位置に移動します。

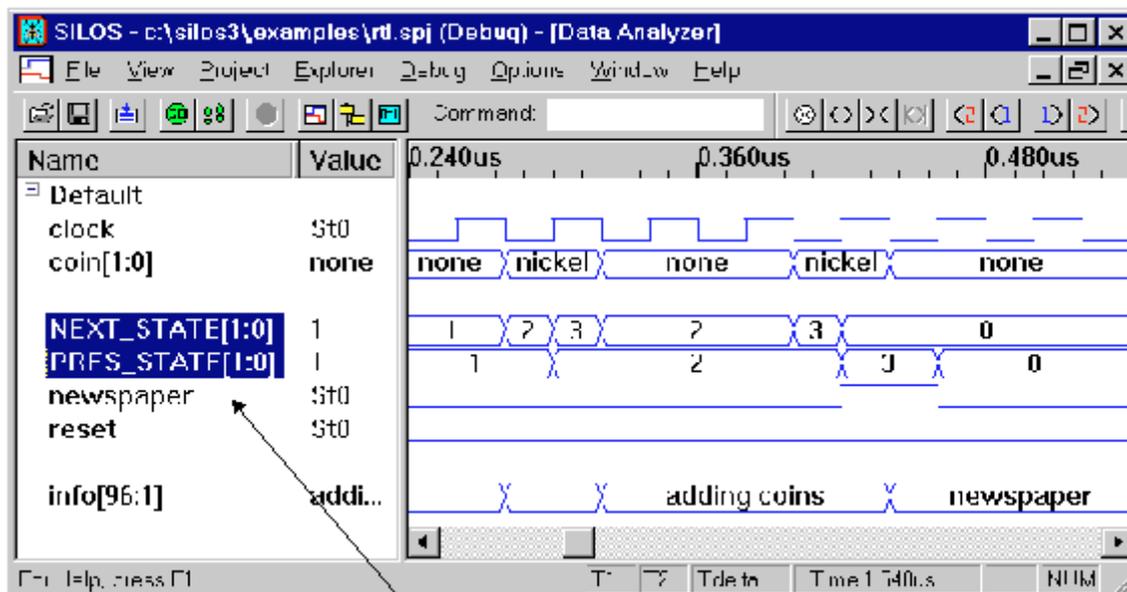
1-16. 信号名の再配列

信号名は、マウスで信号名をドラッグ&ドロップしてからマウスカーソルの矢印で指定して、Signal Name リストボックス内で再配列することができます。以下に例を示します。: 例

マウスで "NEXT_STATE[1:0]" を選択し、"Ctrl" キーを押しながら "PRES_STATE[1:0]" を選択します。

Signal Name リスト・ボックスの信号 newspaper の上に "NEXT_STATE[1:0]" と "PRES_STATE[1:0]" をドラッグ&ドロップします。

信号名は、ドラッグする際、"Ctrl" キーを押しながらコピーすることができます。信号名を削除するには、ハイライトしてキーボードの "Delete" キーを押します。



2. RTL (動作) デバッキング

SILOS は、様々な種類の高度な統合ツールを持ち、動作デザインのデバッグを支援します。

- シングル・ステッピング
- ドラッグ&ドロップ
- ブ레이크ポイント
- Watch ウィンドウ
- Data Analyzer ウィンドウ
- Explorer ウィンドウ

このトピックで説明する機能を以下に示します。

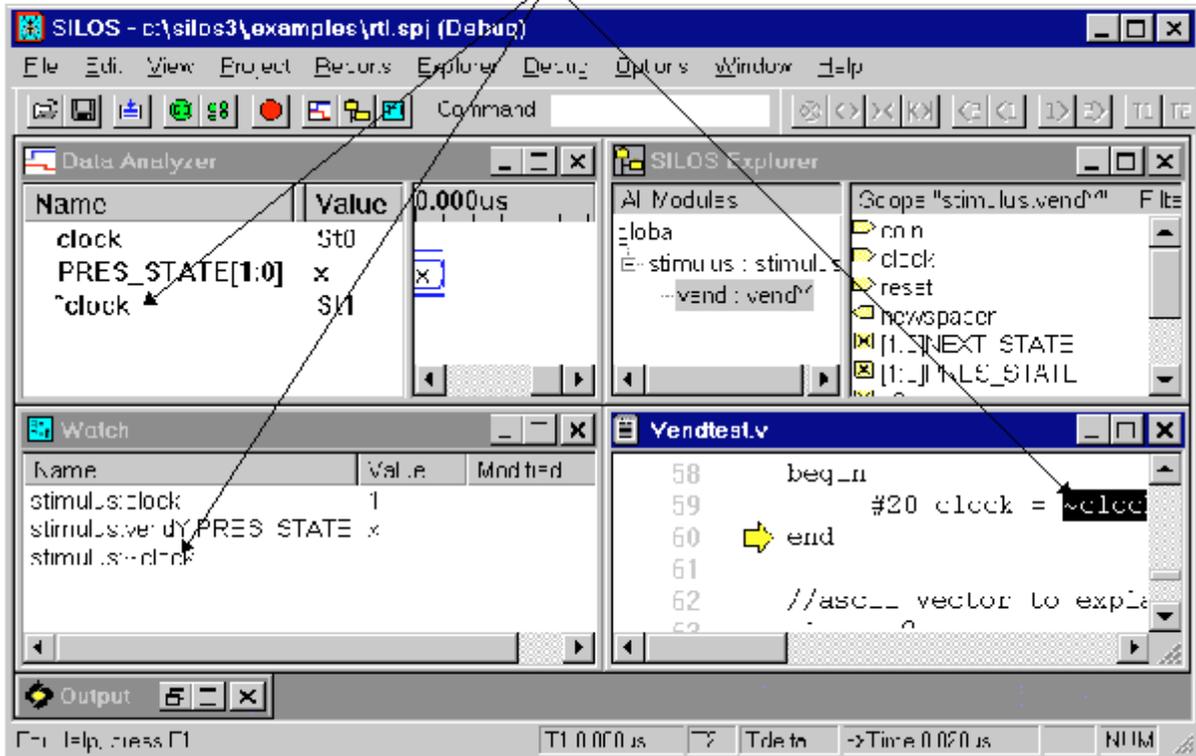
- デザインを1ステップずつ処理する。
- ソース・コードから変数と式を Data Analyzer ウィンドウと Watch ウィンドウにドラッグ&ドロップする。
- Data Analyzer ウィンドウと Watch ウィンドウで変数の値を変更した場合に、それらを参照する。
- 2ビットのレジスターの値を設定する。
- ブ레이크ポイントを使用して、特にチェックする必要のない動作コードをスキップする。

2-1 シングル・ステッピング

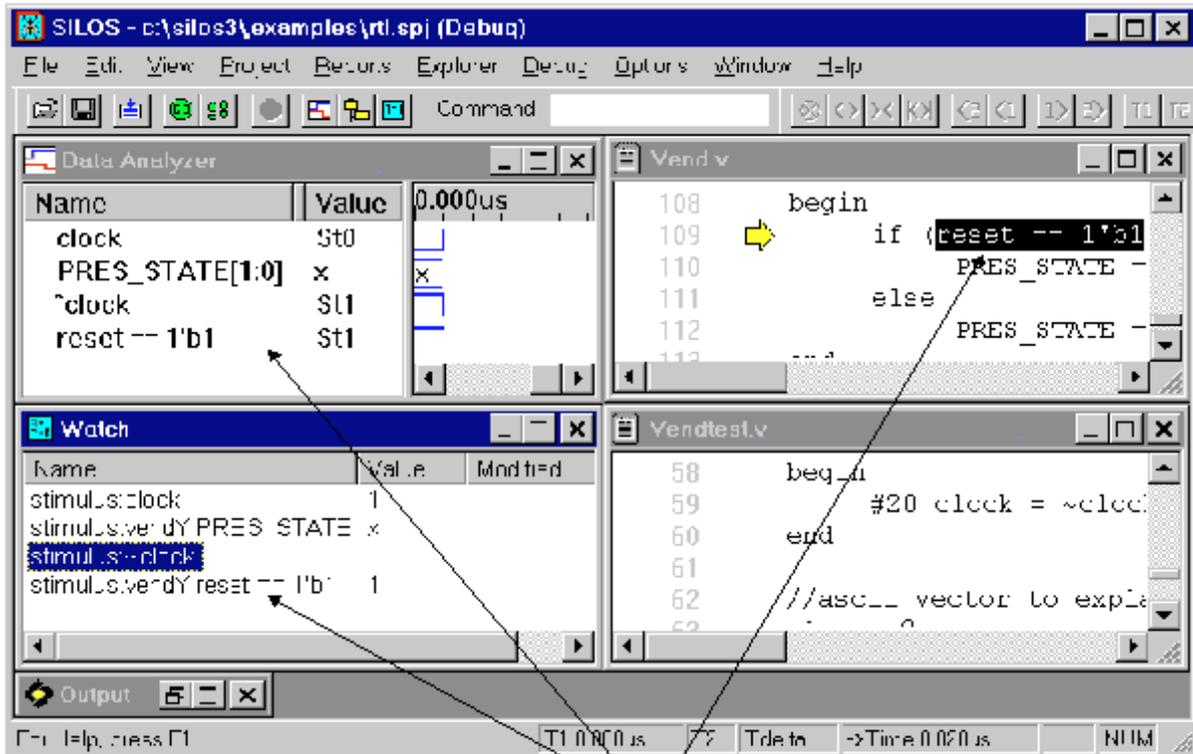
ビヘイビア・コードのシングル・ステッピング機能について、以下に説明します。

- "Debug/Restart Simulation" メニューをクリックすると、時間0でシミュレーションを再開します。
- ツールバーの "Step" ボタンをクリックすると、SSEは "vendtest.v" ファイルのソース・ウィンドウが自動的に開き、実行されたソースコード・ラインの左側に黄色の矢印が表示されます。SILOS の内部エディタは、ノートパッド・エディタと同様で、ソース・ウィンドウで基本的な編集を行なうことができます。(SIMUCAD USER'S MANUAL 3-6 頁 "Edit Menu" を参照)
- 次にツールバーの "Open Watch ウィンドウ" ボタンをクリックし、Watch ウィンドウをオープンします。Output ウィンドウを縮小するために、左上の縮小ボタンをクリックします。

Drag and drop expression “~clock” from file “vendtest.v” to the Data Analyzer and Watch Window. Use the Explorer to drag and drop “stimulus.clock” and “stimulus.vendY.PRES_STATE[1:0]” into the Data Analyzer and Watch Window.



- Window/Tile メニューを選択して、SILOS で現在オープンしているウィンドウを整列 (tile) させます。Data Analyze ウィンドウにスペースを作成するには、Data Analyzer の Signal Name リスト・ボックスの文脈メニューから "Clear Signal List" を選択します。Explorer から Watch ウィンドウと Data Analyzer ウィンドウへ変数 "stimulus.clock" と "stimulus.vendY.PRES_STATE[1:0]" をドラッグ&ドロップします。ウィンドウを表示する追加スペースを作成するには、Explorer ウィンドウを閉じます。
- SILOS は、ソースコードの変数を Data Analyzer ウィンドウと Watch ウィンドウに直接ドラッグ&ドロップする特別な機能を持っています。また、ソースから式をドラッグすることもできます。例えば、ソースコードの式を表示するには、"vendtest.v" の "~clock" 式をマウスでハイライトして、 "~clock" を Data Analyzer ウィンドウと Watch ウィンドウにドラッグ&ドロップします。
- "vendtest.v" のソース・ウィンドウが開くまでツールバーの "Step" ボタンを続けてクリックします。
- Window/Tile メニューを選択して SILOS のウィンドウをオープンします。

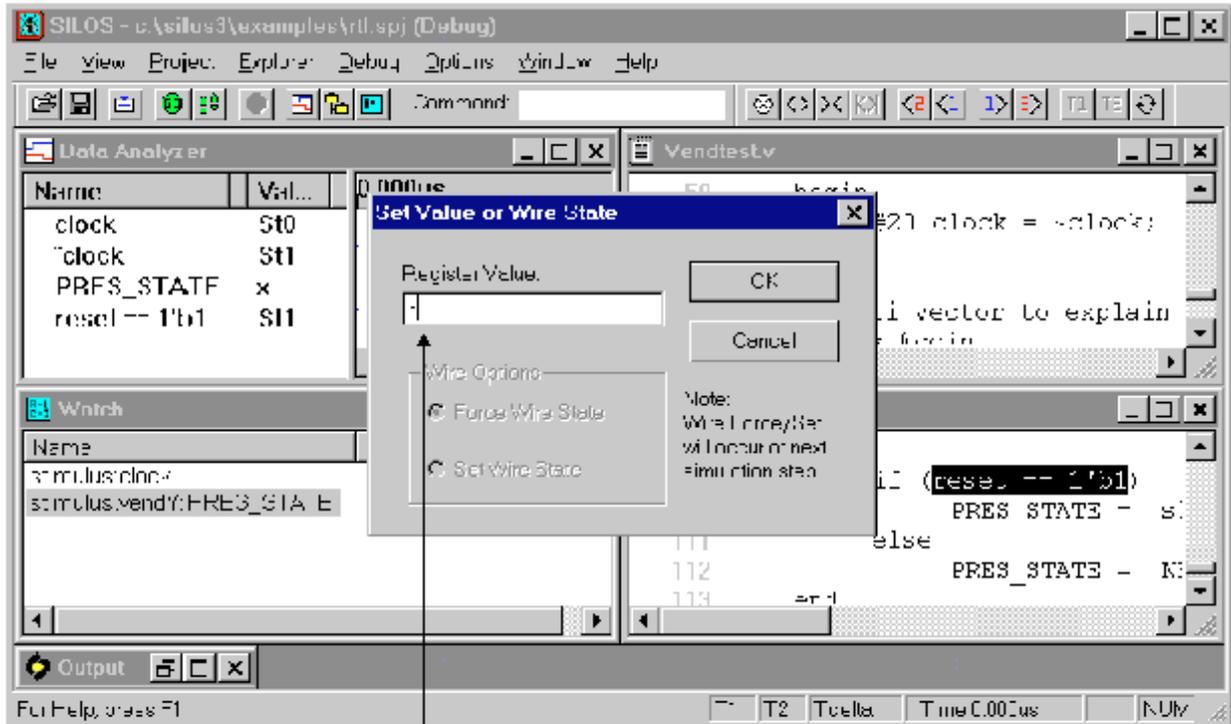


Drag and drop expression "reset == 1'b1" into the Data Analyzer and Watch Windows.

- 次に "vendtest.v" ファイルの "reset == 1'b1" の式をマウスでハイライトし、この式を Data Analyzer ウィンドウにドラッグ&ドロップします。
- "Step" ボタンを続けてクリックします。Watch ウィンドウと Data Analyzer ウィンドウの値は、1ステップ進むごとにアップデートされます。

2-2. 値の設定と値の強制定義

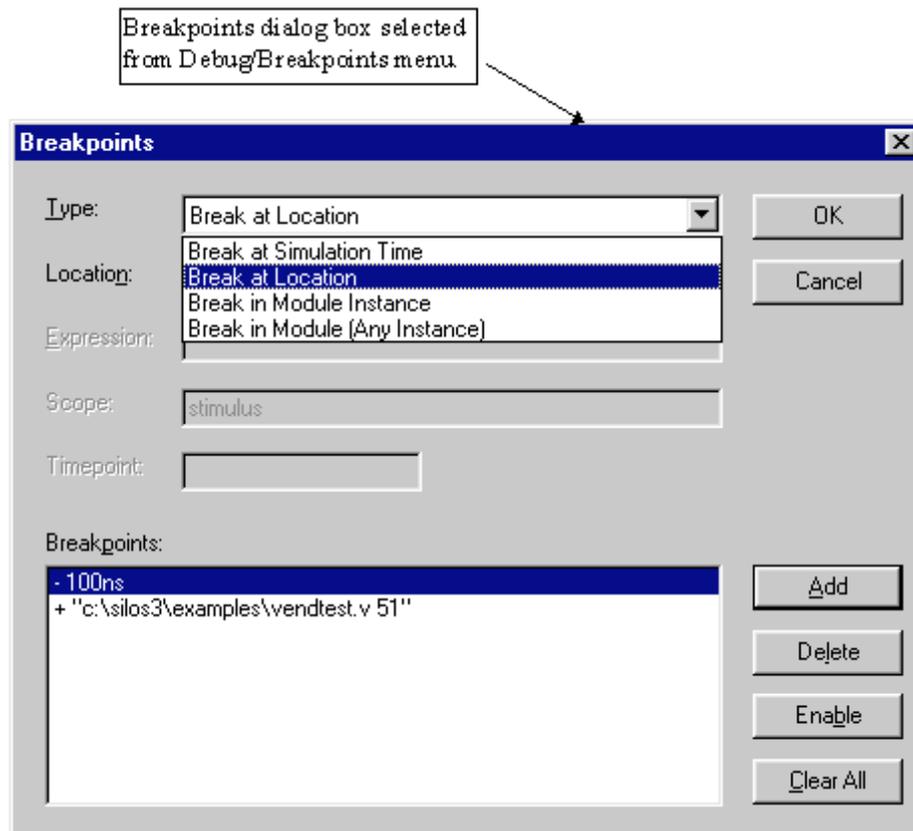
デバッグ中は、変数に値を設定したり、強制的な値を使用したりすることが多くあります。以下の操作を行うと、信号 PRES_STATE を時間 0 で "3" に設定され、プログラムのスタート時点で無料で新聞が排出されるようになります。



Set the value for register PRES_STATE to "3".

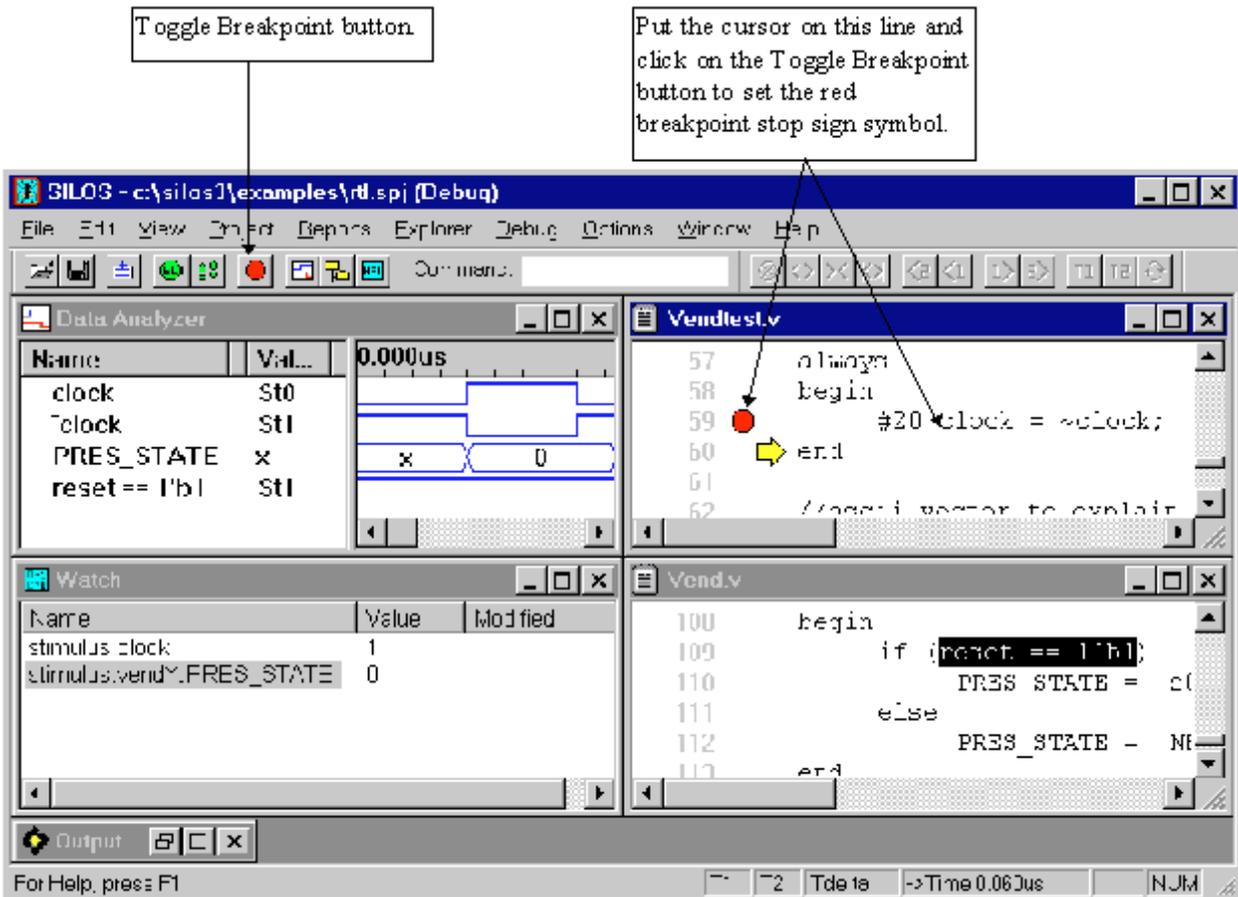
- シミュレーションを時間 0 にリセットするには、"Debug/Restart Simulation" メニューを選択します。
- マウスの右ボタンを使用して Watch ウィンドウの信号 "stimulus:vendY: PRES_STATE" をクリックします。
- "Set Value or Wire State" ダイアログ・ボックスで "X" のレジスター値を "3" に変更して "OK" ボタンをクリックします。
- Watch ウィンドウの "Modified" の列に "Set Value 3" が表示されます。シングル・ステップ・ボタンを押すと、変数 "PRES_STATE" が 3 に設定され、"Set Value 3" の表示が消えます。シングル・ステップを続けると Data Analyzer の信号 "PRES_STATE" は、"X" に代わって "3"（またはシンボル名を使用すると 15 セント）で始まります。

2-3. ブレイクポイント



- ブレイクポイントは、"Debug/Breakpoints" メニュー、またはツールバーの "Toggle Breakpoint" ボタンを使用して設定することができます。SSE は動作コードをデバッグするために以下のブレイクポイントを使用することができます。
- シミュレーション時間でブレイクする。: 選択した時間の前まで論理シミュレーションを実行し、ストップします。
- ロケーションでブレイクする。: 選択したソース行の前まで論理シミュレーションを実行し、ストップします。
- モジュール・インスタンスでブレイクする。: モジュール・インスタンスを選択すると、そのモジュール・インスタンス内の全ソース行の論理シミュレーションを実行して、ストップします。
- モジュール (全てのインスタンス) でブレイクする。: モジュール・インスタンスを選択すると、そのモジュールのすべてのインスタンス内のソース行の論理シミュレーションを実行して、ストップします。
- ブレイクポイントは、確認したいコードでシングル・ステップを続けることができるように、必要のないコードをスキップする場合に使用することもできます。例えば、サブディレクトリー "examples" 内の "vendtest.v" ファイルでシングル・ステップを続ける場合は、以下の操作を行いません。
- 以下のラインにマウス・カーソルをおきます。 ("vendtest.v" ファイルの最下行近く):
- #20 clock = ~clock;
- ツールバーの "Toggle Breakpoint" ボタンをクリックします。赤いストップ・サイン・シンボルが行番号の左隣に表示されます。

- 次にファイル "vendtest.v" が終了するまでツールバーの "Step" ボタンを繰り返しクリックします。
- 次にツールバーの "Go" ボタンをクリックして、タイムポイントを終了します。"Go" ボタンを2回クリックすると、ブレイクポイントで論理シミュレーションがストップします。
- モジュール "vendtest.v" のシングル・ステップの実行を続けることができます。
-



2-5. ゲートレベル・デバッキング

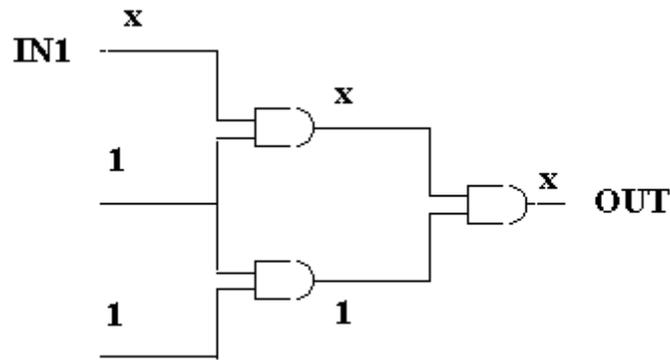
このトピックでは、以下の機能について説明します。

- ゲートレベル・デザインをトレースバックして、不定レベルの原因を見つけ出します。

2-5-1. 信号入力へのトレース

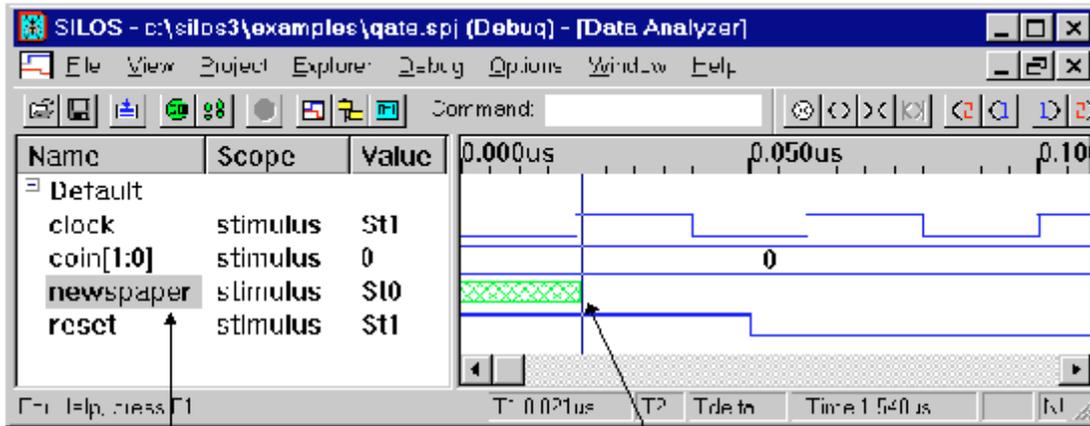
RTL デザインを合成する場合、ゲート・デザインの実行結果が設計者の期待どおりにならず、シミュレーション結果が RTL デザインと異なる場合があります。ゲートレベルのデバッキングは回路図がないため、非常に難しくなる可能性があります。また、合成ツールと配置配線ツールは、既存の名前を変更し、新しい名前を作成する場合があります。

SILOS はトポロジー（位相幾何学）を使用して、信号を逆方向にトレースし、期待値と異なる状態値の原因を見つけ出します。以下のような単純な回路図では、信号 "OUT" が不定 ("X") レベルの場合、"Trace Signal Input" 機能を使用して逆方向にトレースし、入力 "IN1" が不定レベルの原因であることをを見つけ出します。



このチュートリアルでは、ゲートレベルに合成された新聞自動販売機のデザインで "Trace Signal Inputs" メニューを使用して、信号 "newspaper" が時間 0.021 マイクロ秒 (us) で不定レベルからロー・レベルに変化した原因を見つけ出します。

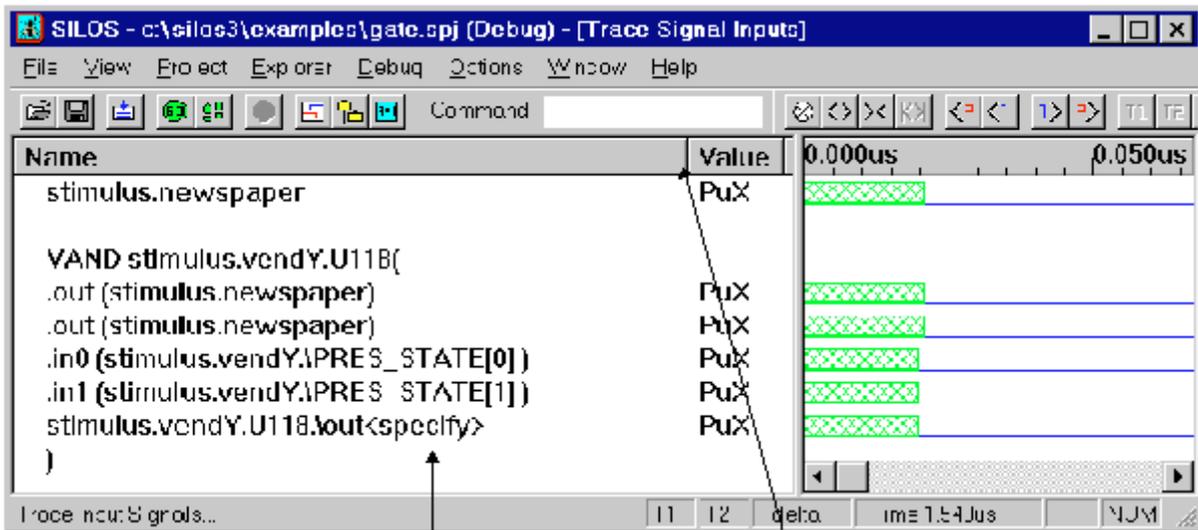
- Project/Open メニューを選択し、"Open Project" ダイアログ・ボックスを開きます。
- プロジェクト・ファイル "Gate.spj"（新聞自動販売機 FSM のゲートレベル・デザイン）をハイライトして、"Open Project" ダイアログ・ボックスの "開く" ボタンをクリックします。
- "Go" ボタンをクリックして、シミュレーションを実行します。
- "Open Analyzer" ボタンをクリックして、Data Analyzer を開きます。（Data Analyzer が自動的に開かない場合）



Highlight signal "newspaper" and click with the right mouse button in the Signal Name list box. Select "Trace Signal Inputs" when the context menu opens.

At time=0.021 us, "newspaper" changes from Unknown to Low.

- "Signal Name" リストボックスで信号 "newspaper" をマウス右ボタンでクリックし、文脈メニューを開きます。
- "Trace Signal Inputs" メニューを選択し、"Trace Signal Inputs" ウィンドウを開きます。



The "<specify>" appended to the end of a signal name means this is the signal for the specify block delays from the input to output pins. **Double click** on this signal to display the gates that are driving the module output.

To view the full signal name, use the mouse to drag the vertical bar and increase the size of the Name column.

- 信号 "newspaper" はモジュール・ポートになっているため、"Trace Signal Inputs" ウィンドウにはモジュール "vend" の各変数が表示されます。信号名の最後に "<specify>" (入力ピンから出力ピンまでの遅延を定義するブロック) という記述が追加されている信号をダブルクリックすると、信号 "newspaper" を駆動しているゲートを見つけることができます。この例では、"stimulus.vendY.U118%out<specify>" をダブルクリックします。

注意： 波形ではなく "No Saved Data" が表示された場合は、変数のシミュレーション履歴が保存されていないことを意味します。シミュレーション履歴が保存されない原因は、信号が celldefine 範囲の内部に存在する可能性が考えられます。 celldefine 範囲内のデータの保存については、SIMUCAD USER'S MANUAL の 3-14 頁 "Project Setting Menu Selection" を参照してください。

Two input "and" gate.

Gate instance name.

Name	Value	0.000us	0.050us
stimulus.vendY.U118.out<specify>	PuX	[Waveform]	[Waveform]
.and stimulus.vendY.U118(.out(stimulus.vendY.U118.out<specify>)).in(stimulus.vendY.PRES_STATE[0])	PuX	[Waveform]	[Waveform]
.in(stimulus.vendY.PRES_STATE[0])	PuX	[Waveform]	[Waveform]
.in(stimulus.vendY.PRES_STATE[1])	PuX	[Waveform]	[Waveform]

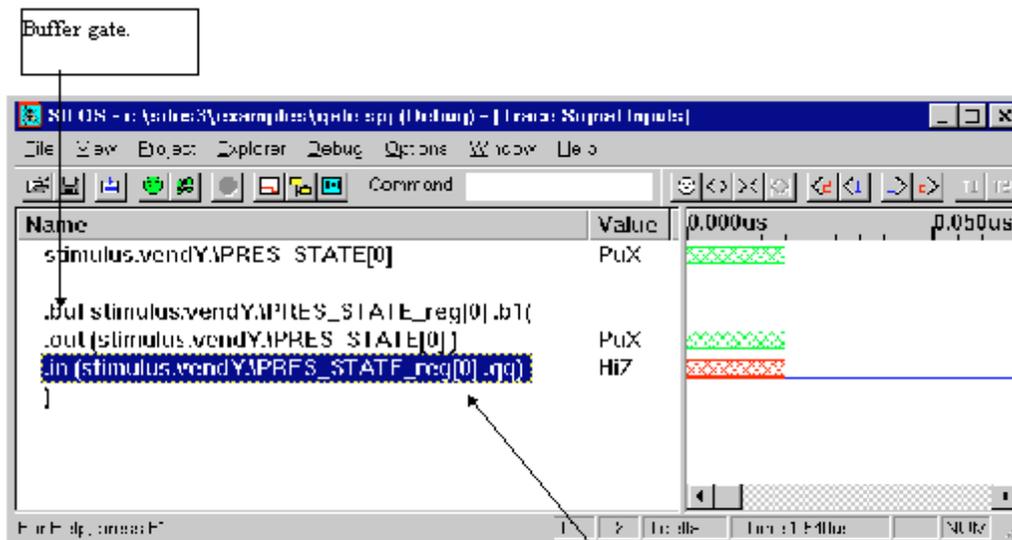
"in" is the local name for the input pin.

Instance name for the input pin.

Since both inputs change from Unknown to Low at time=0.020us for this example, **double click** on input "stimulus.vendY.PRES_STATE[0]".

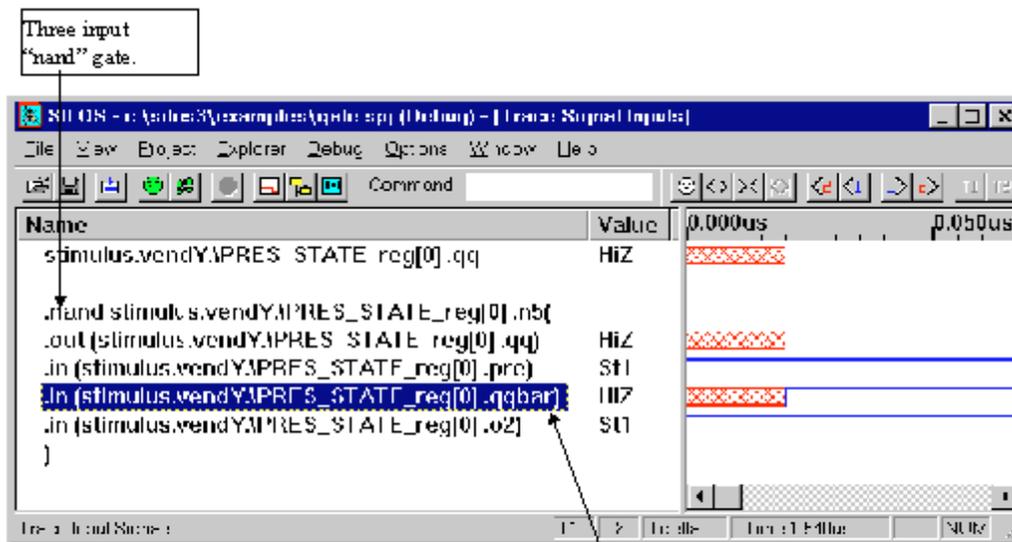
- 信号 "stimulus.vendY.U118%out<specify>" が、Trace Signal Inputs ウィンドウの先頭に表示されます。"stimulus.vendY.U118%out<specify>" の下にリストされた信号は、2 入力の "and" ゲートで、出力がこの "stimulus.vendY.U118%out<specify>" 信号を駆動しています。"and" ゲートの双方の入力が時間 0.021 マイクロ秒 (us) で不定レベルからロー・レベルに変化しているため、いずれかの入力を逆方向にトレースすることができます。原因を見つけるために、"stimulus.vendY.PRES_STATE[0]" をダブルクリックします。

注意： 誤って別の信号をダブルクリックしてトレースを実行してしまった場合は、トレースされた信号名 (Signal Name リストボックスの先頭の信号名) をダブルクリックすると、信号のトレースを元に戻すことができます。



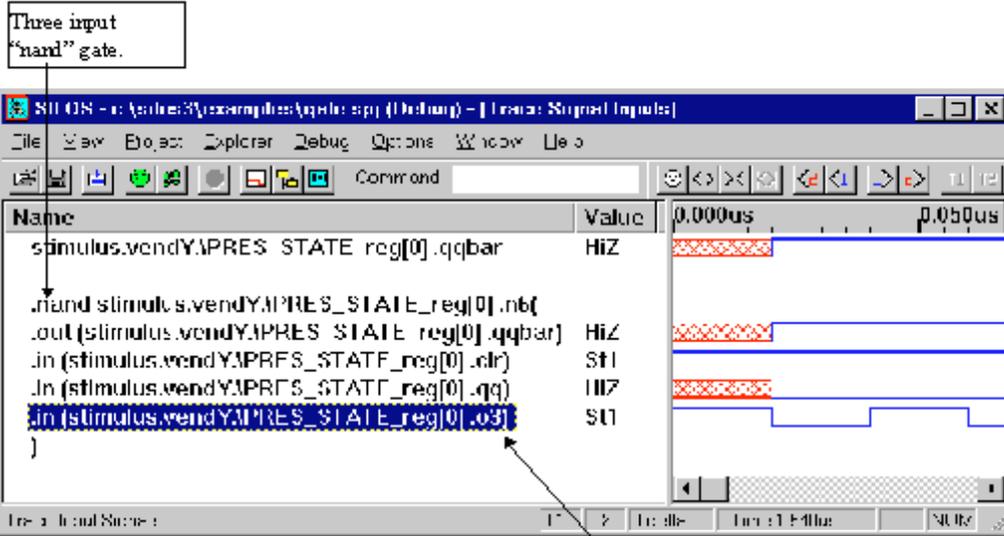
Double-click on the input to the buffer to continue tracing back through the topology.

- 信号 "stimulus.vendY.PRES_STATE[0]" はバッファ・ゲートで駆動されているので、バッファ入力 "stimulus.vendY.¥PRES_STATE_reg{0}.qq をダブルクリックして、トポロジーを用いた逆方向のトレースを続けます。



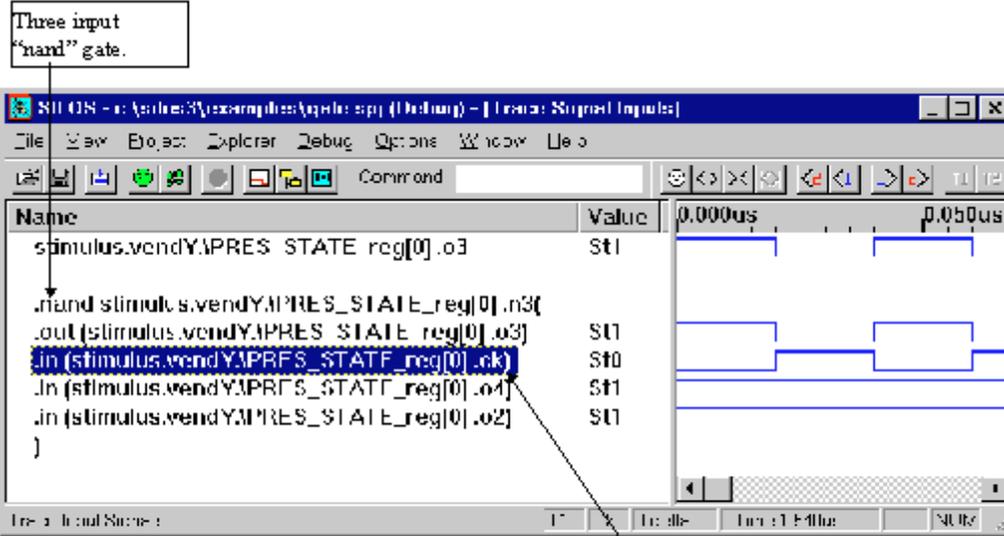
Double-click on the "qqbar" input because the Unknown to High transition on this input caused the output to change.

- 信号 "stimulus.vendY.¥PRES_STATE_reg[0].qq" は、3入力の "nand" ゲートで駆動されていますが、信号 "stimulus.vendY.¥PRES_STATE_reg[0].qqbar" のみが不定レベルからロー・レベルに変化しています。信号 "stimulus.vendY.¥PRES_STATE_reg[0].qqbar" をダブルクリックしてトレース・バックを続けます。



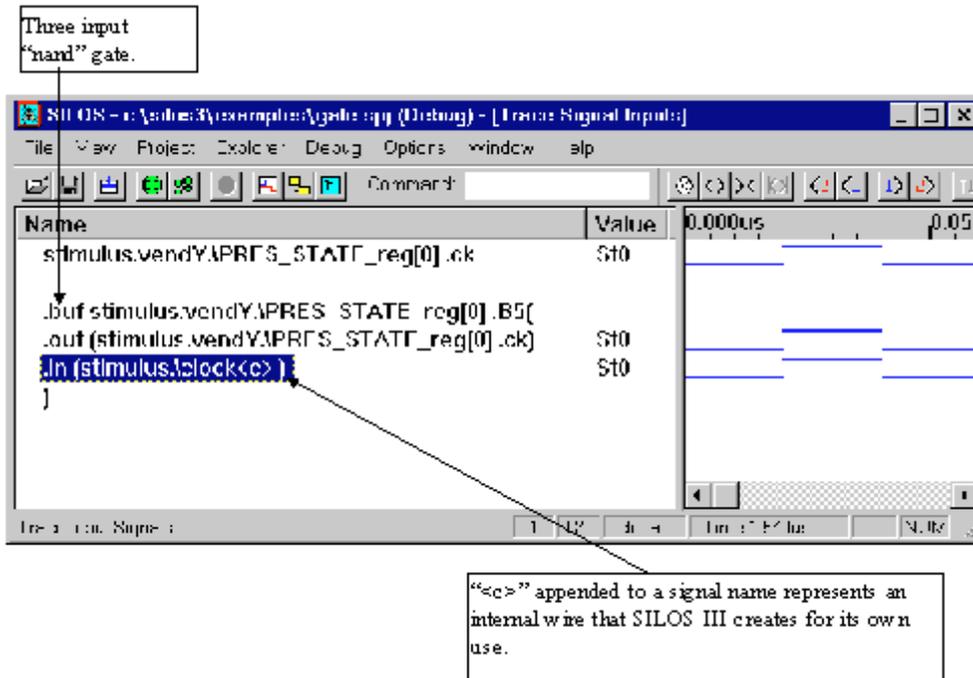
Double-click on the "o3" input to the "nand" gate because the High to Low transition on this input caused the output to change.

- 信号 "stimulus.vendY.%PRES_STATE_reg[0].qqbar" は、別の 3 入力 nand ゲートで駆動されています。信号 "qq" と "qqbar" は、nand ゲートのフィードバックでラッチを形成しているため、"qq" 入力を更にトレース・バックするノードがありません。"qq" の代わりに、信号 "stimulus.vendY.%PRES_STATE_reg[0].o3" をダブルクリックしてトレース・バックを続けます。この信号 "stimulus.vendY.%PRES_STATE_reg[0].o3" は、時間 0.020us でローに変化し、出力 "qqvar" が変化する原因となっています。



Double-click on the "ck" input to the "nand" gate. The Low to High change on "ck" caused the nand gate to change.

- 信号 "stimulus.vendY.¥PRES_STATE_reg[0].o3" は、3 入力の nand ゲートで駆動されています。信号 "stimulus.vendY.¥PRES_STATE_reg[0].ck" は、時間 0.020us で nand ゲート出力が変化することになっているため、この信号をダブルクリックします。



- 信号 "stimulus.vendY.¥PRES_STATE_reg[0].ck" は、バッファで駆動されており、このバッファの入力は、"stimulus.clock<c>" になっています。信号名 "clock" に追加された "<c>" は、SILOS プログラムが内部で使用するために作成されたワイヤです。この内部信号は、モジュール "stimulus" からモジュール "vend" へのポートの境界にまたがる信号表現に使用されています。

入力刺激信号 "clock" をトレース・バックすると、このクロック信号は時間 0.020us でハイになるように設計で指示されており、信号 "stimulus.newspaper" が時間 0.021us で不定レベルからロー・レベルになる原因となっていることが確認できます。これは設計に問題があるため、ゲートレベル・設計、あるいは RTL レベルの設計を修正し、再度合成します。

