

WaveFormer Pro / VeriLogger Pro

と HPパターンジェネレータの波形データ生成

HPパターンジェネレータは、そのシステム上に豊富な波形データの編集機能を持っていますが、設計環境のコンピュータ上で他のデータと一緒に、パターンジェネレータに与えるデータも管理できるのが望ましいと言えます。HPパターンジェネレータとのインターフェースを持つ強力な波形編集ツールである SynaptiCAD社の WaveFormer Pro / VeriLogger Pro はWindowsベースのソフトウェアであり、HPパターンジェネレータへのデータ生成とコンピュータ上での波形データの管理に最適なツールです。

(WaveFormer Pro / VeriLogger Proの編集機能)

- 1) 周期とデューティ比を指定してのクロックの生成
- 2) マウスクリックでの信号波形(バスも)の入力
- 3) 時間式での波形入力
- 4) バス値の生成式での自動入力
- 5) Boolean式 (VHDL/Verilog記述) で波形の演算として
 - * 単純 組み合わせ回路
 - * レジスター代入式
- 6) Verilogコード記述での波形生成
- 7) Perlスクリプトでの波形定義

1. クロックの生成

周波数(または周期)、デューティ比、初期値、オフセットを指定してのクロックの生成

2次クロックの周期は1次クロックの式で生成

(例) `'2*clk-'30` `'0.5*clk+'20`



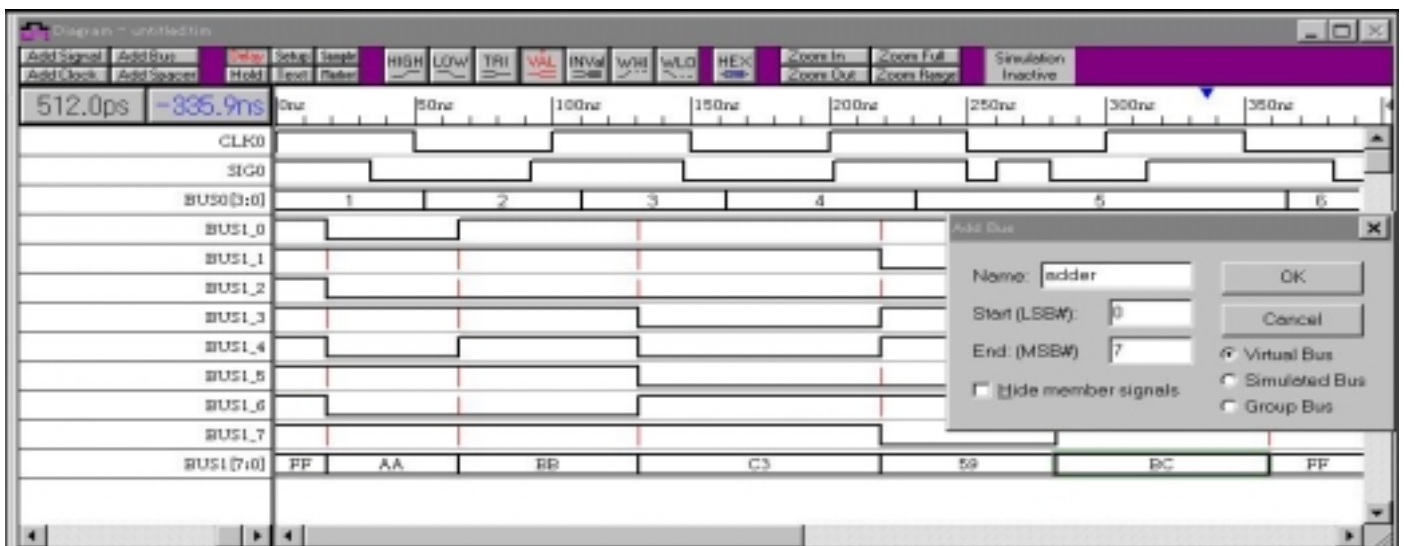
2. マウスクリックで信号波形/バスの入力

マウスで信号エッジをクリックすると、設定した2つの信号値間(Validは連続OK)をトグルします。

信号値: L, H, Hi-Z, Valid, Invalid, Weak-L, Weak-H

バスは個々の構成する信号の編集で定義することも、バス一括として定義することもできます。一括バスはValid, Invalidを使い、エッジ定義をして、各Validステートに後から値を振り付けします。値の振り付けは手動でも、また連番など規則性があれば自動化(Label Equation: 3項参照)できます。バスを構成信号に分解したり、信号をまとめて新しくバスを定義することも可能です。

豊富な編集機能 - 信号エッジのマウスドラッグによる移動、信号波形のCopy&Paste、ダイアログでのエッジ時間入力、グリッドによるエッジの設定時間単位の規定、Undo/Redo など。。。。。



3. 時間式での波形入力/バス値のラベル生成式での自動入力

複雑なパターンを持つ波形の生成を、すばやく行う手段を提供します。

(例) 時間式: 170=X (200=V)*20

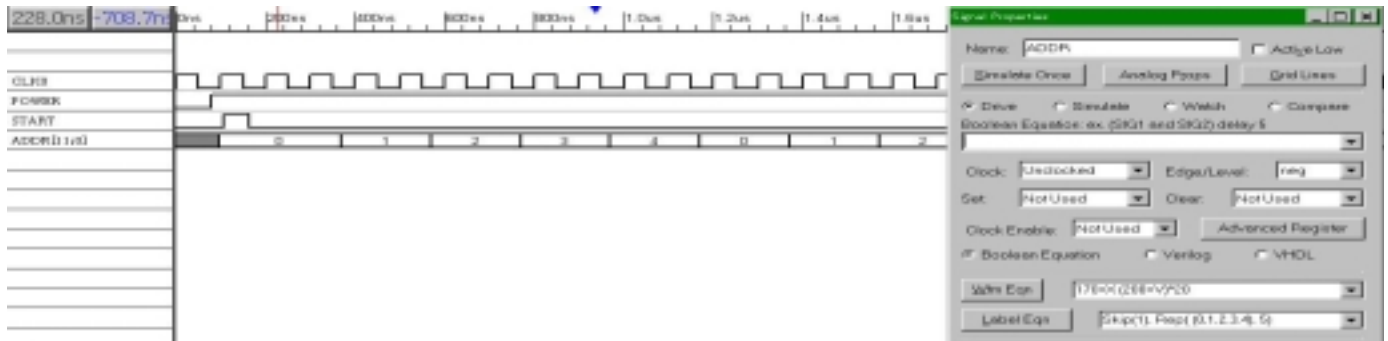
既存の信号でない場合、この名前の信号が生成され、存在する場合は、波形の生成は継ぎ足しとなります。

170=Xは、170[ns]間不定値Xを持つことを意味します。

(200=V)は200nsほどある有効な信号が続くことをあらわします。

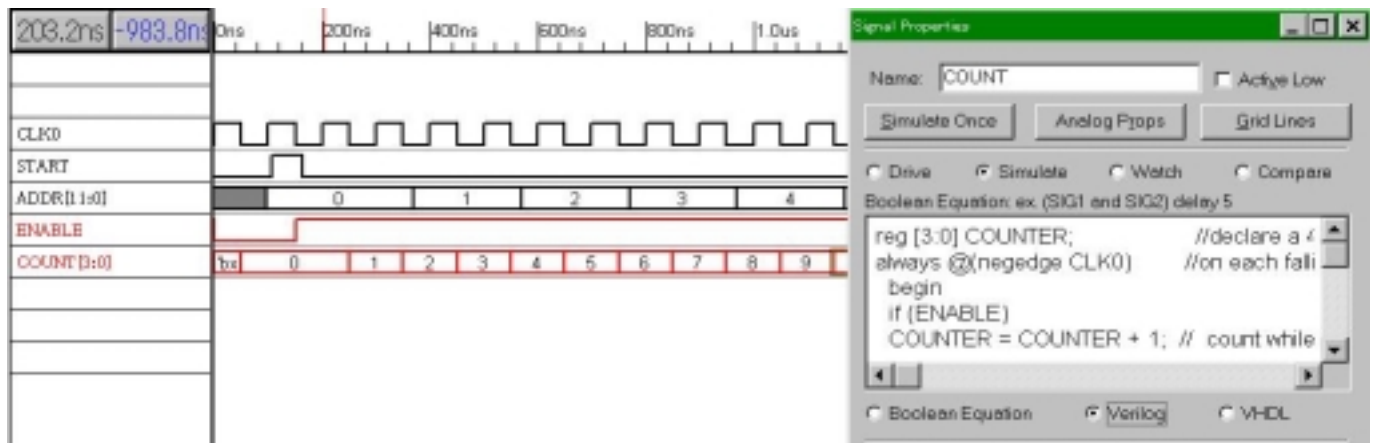
つぎの *20はカッコ内のシーケンスが20回繰り返されることをあらわします。

(例) ラベル式: Skip(1), Rep((0,1,2,3,4), 4) ~ Skip(1)は最初の一つのステートにラベリングをしないことを意味し、次は0-4を4回繰り返す意味です。



4. Boolean式で波形の演算として組み合わせ回路波形やレジスタ回路波形の生成 Verilogコード直接記述での波形生成も

Waveform Pro/VeriLogger Proには、Verilogシミュレーション・コアが組み込まれています。波形の組み合わせ演算(Boolean式- VHDLまたVerilog演算子で SIG2: SIG0 and not SIG1)または、そのレジスタへの代入はWaveform Pro / VeriLogger Pro 内部でVerilogコードに変換して、Verilogシミュレーション・コアに渡しシミュレーションされ、結果が波形表示されます。Verilogでの直接入力も許され、Verilogの持つ強力なシンタックスを使ってカウンターやシフターなどの複雑な波形も描画できます。



5. Perlスクリプトで波形生成

Waveform Pro/VeriLogger Proには、Verilogシミュレーション・コアだけでなくPerl言語プロセッサも組み込まれています。PerlスクリプトでWaveform Pro/VeriLogger ProのAPIにアクセスでき、マウスクリックでは実現できない複雑な波形も簡単なスクリプトで描画できます。

```
for ($i = 0; $i < $numClockPeriods; $i++)  
{  
    $halfPeriod = $halfPeriod + 2 * $i;  
    wfm("DRIFT_CLK $halfPeriod=0 $halfPeriod=1");  
}
```

