

Verilog-Aデバッガ入門

はじめに

近年、アナログ/ミックスド・シグナル設計において、Verilog-AMS ハードウェア記述言語 (Verilog-A) が広く用いられるようになりました。これに伴い、多くの EDA ベンダが Verilog-A に対応したシミュレーション・ツールを提供しており、SIMUCAD の SmartSpice も、数年前に Verilog-A をサポートしました。その後、Verilog-A デバッガ (VA デバッガ) が開発され、SmartSpice のコンポーネントとして追加されました。VA デバッガは、シミュレーション中に Verilog-A のソース・コードをデバッグするための便利でパワフルなツールです。VA デバッガを使用することにより、設計上の問題点を素早く突き止め、設計効率を大きく向上できます。本稿では、VA デバッガの重要な特徴をいくつか紹介します。

VA デバッガの起動手順

VA デバッガは、SmartSpice のコンポーネントであるため、直接起動できません。VA デバッガを起動するには、次のステップを行う必要があります。

- 1) Verilog-A オプション `-debug` を、Verilog-A ファイルに設定する必要があります。設定方法は次の 3 通りあります。
 - a) ネットリスト内で `.options` コマンドを使用
`.options veriloga-args= "-debug"`
 - b) ネットリスト内で `.verilog` コマンドを使用
`.verilog "<file-name-string>" -debug`
 - c) 環境変数 `SIMUCAD_VERILOGA_ARGS` を `-debug` に設定。UNIX での例を次に示します。
`%setenv SIMUCAD_VERILOGA_ARGS -debug`

このオプションを設定すると、Verilog-A ソース・ファイルは最適化なしで処理され、必要なデバッグ情報が生成されます。

- 2) SmartSpice のウィンドウのメニューバーから [編集] → [環境設定] を選択し、[デバッグ情報の生成を有効にする] チェックボックスをオンにします (図 1 を参照)。

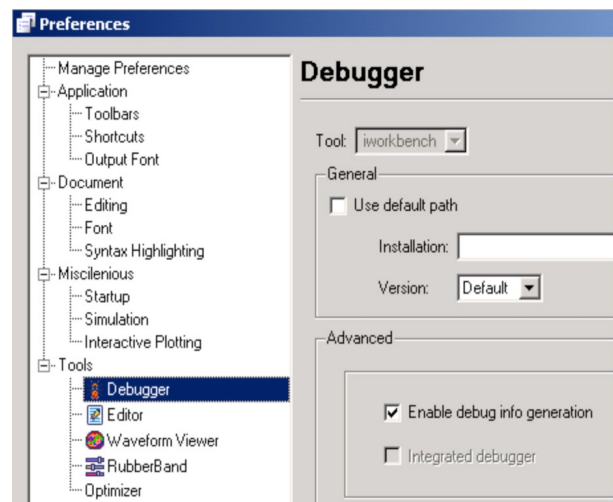


図1: VAデバッガの環境設定

この環境設定は、SmartSpice を終了するときに保存され、次の SmartSpice 実行時に復元されます。

- 3) 入力デッキを読み込んだ後、SmartSpice のメニューバーから [解析] → [Debug] をクリックすると、VA デバッガでシミュレーションが実行されます。

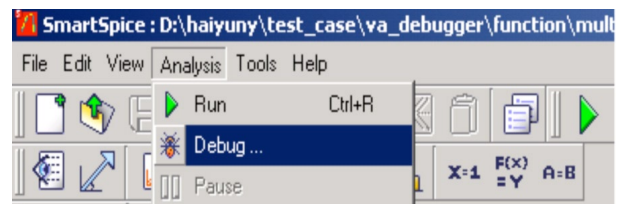


図2: VAデバッガを起動するメニュー

ただし、読み込んだ入力デッキに、ステップ 1) で説明した `-debug` モードで処理される Verilog-A ソース・ファイルが存在しない場合、VA デバッガは起動されません。また、Verilog-A ソース・ファイルのデバッグ情報が生成されないため、シミュレーションは最適化モードで実行されます。

ステータス情報の説明

VA デバッガの下部には、いくつかのステータスバーがあります。このうち3つのステータスバーには、特別な意味があります。

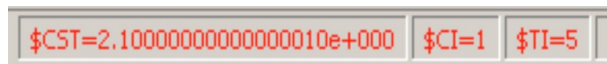


図3: ステータスバー

\$CST: 現在のシミュレーション時間

\$CI: 現在のシミュレーション時間での反復回数

\$TI: 総反復回数

これらの変数の値が変化すると、変化した値が赤色でハイライト表示されます。変化がない場合は黒色で表示されます。この仕組みにより、実際の数値に最小限の注意を払うだけで、シミュレーションの進捗状況をモニタできます。

ここで注意したいのは、VA デバッガにおける \$TI の更新方法です。この変数は、有効なシミュレーション反復だけをカウントするもので、1つのタイム・スポットにおいてシミュレーションが終了した後に1回だけ更新されます。つまり、あるタイム・スポットで何回か反復してもシミュレーションが収束できない場合、その反復回数は \$TI にはカウントされません。\$CI の値は、シミュレーションが収束するかどうかにかかわらず、反復のたびに更新されます。そのため、\$TI の値は \$CI と同期しませんが、それが正常な動作です。

ブレークポイントの設定

他のデバッガと同様、VA デバッガでブレークポイントを設定または設定変更するには、いくつかの方法があります。ブレークポイントを設定するのに最もよく使用される便利な方法は、ソース・コード行の左でマウスを左クリックすることです。ブレークポイントが正しく設定されると、[STOP] マークがその行の左に表示されます。ポート宣言や変数宣言など、デバッグ情報を持たない行には、ブレークポイントを設定できません。

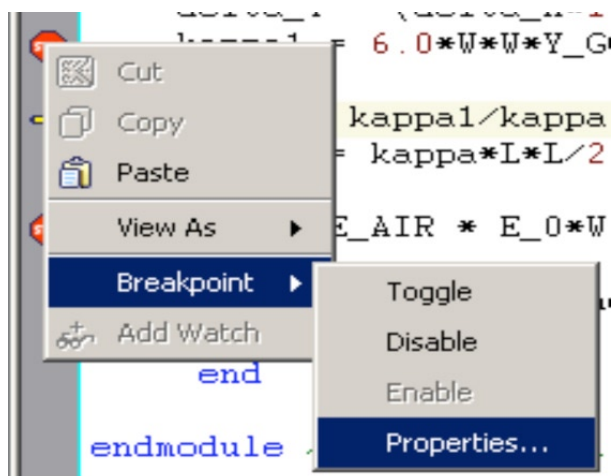


図4: ブレークポイント設定用ダイアログを開く1つの方法

VA デバッガは、デバッグ時に大変便利な条件付きブレークポイントもサポートしています。条件付きブレークポイントを設定するためのダイアログを開くには、2つの方法があります。1つは、任意の既存のブレークポイント上で右クリックし、[Breakpoint] → [Properties] を選択する方法です。

もう1つは、[Breakpoints] ダイアログのブレークポイント・リストから任意のブレークポイントを選択し、[Properties] ボタンをクリックする方法です。

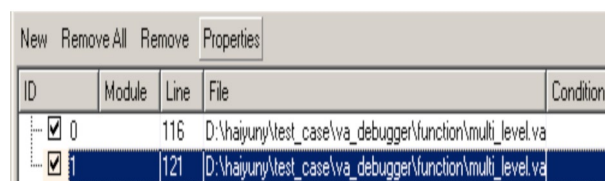


図5: ブレークポイント設定用ダイアログを開くもう1つの方法

ブレークポイント設定用ダイアログには [Condition] フィールドがあり、そこに条件式を入力できます。条件式は、ブール値として評価できる論理式である必要があります。条件式が正しく評価できない場合（式の種類が正しくない場合や、式に未知のシンボルが含まれる場合など）、そのブレークポイントは削除され、[Input/Output] ダイアログにメッセージが表示されます。

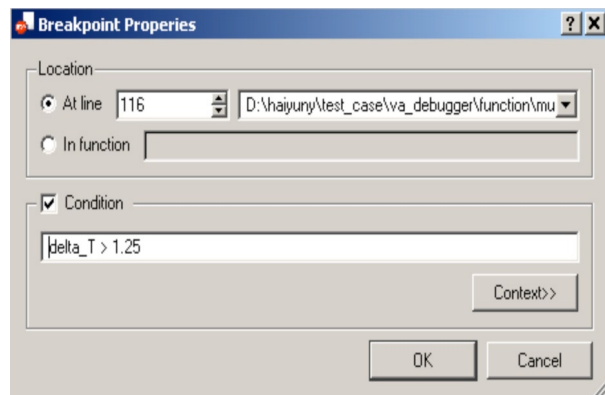


図6: ブレークポイント設定用ダイアログ

条件式の作成を簡単にするため、ブレークポイント設定用ダイアログには [Context] セクションが用意されています。ブレークポイント設定用ダイアログの [Context] ボタンをクリックすると、[Context] セクションの表示 / 非表示を切り替えられます。

[Context] セクションには、変数とその値のリストが表示されます。変数を右クリックして、コンテキスト・メニューから [Insert Name] または [Insert Value] をクリックすると、変数の名前または値を条件式に簡単に挿入できます。この仕組みにより、コピー・ペーストしたりウィンドウを切り替えたりする手間を省くことができます。

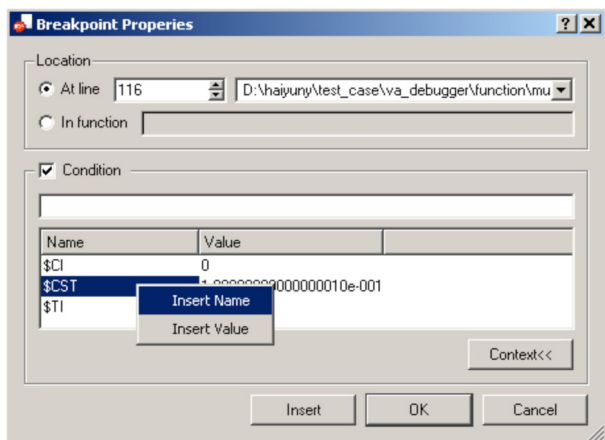


図7: [Context]セクション

まとめ

本稿では、いくつかの機能に焦点を当てて VA デバッガを紹介しました。1 つ目の項では、SmartSpice から VA デバッガを正しく実行するために必要な 3 つのステップを紹介しました。2 つ目の項では、ステータスバーの 3 つの特別な変数と、その値の更新方法について説明しました。最後に、ブレークポイントの設定方法、特に条件付きブレークポイントの設定方法を紹介しました。