

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。

QI153014-7.2.0

はじめに

アルテラ FPGA の容量と複雑さは進化し続けており、IP (Intellectual Property) の必要性がますます重要になっています。IP メガファンクションを使用すると、デザインおよび検証時間が短縮されるため、デザインのカスタマイズに集中することができます。アルテラおよび AMPPSM (Altera Megafunction Partners Program) は、アルテラ FPGA 用に最適化された IP メガファンクションの幅広いポートフォリオを提供しています。パラメータ化によって、これらの再利用可能な IP ブロックはデザイン要件に合わせてカスタマイズ可能です。

IP ソース・コードが暗号化などにより機密となっても、アルテラの Quartus[®] II ソフトウェアにより、アルテラ IP を含むデザインを簡単にシミュレートできます。Quartus II ソフトウェアを使用すると、IP デザインをカスタム・コンフィギュレーションし、好みのシミュレーションで使用できる VHDL または Verilog HDL の機能シミュレーション・モデルを生成できます。

この章では、デザインの IP メガファンクションをインスタンス化し、そのファンクション・モデルをアルテラがサポートするサードパーティ・シミュレーション・ツールでシミュレートするためのプロセスの概要を示します。この章では、IP メガファンクションは、アルテラ・メガファンクション、IP MegaCore[®] ファンクション、および IP AMPP メガファンクションを指します。すべての IP MegaCore ファンクションには、IP 機能シミュレーション (IPFS) モデルが付属しており、機能シミュレーションをサポートします。アルテラ・メガファンクションと AMPP メガファンクションには、機能シミュレーションに IPFS モデルを必要とするものがあります。

IP 機能シミュレーション・フロー

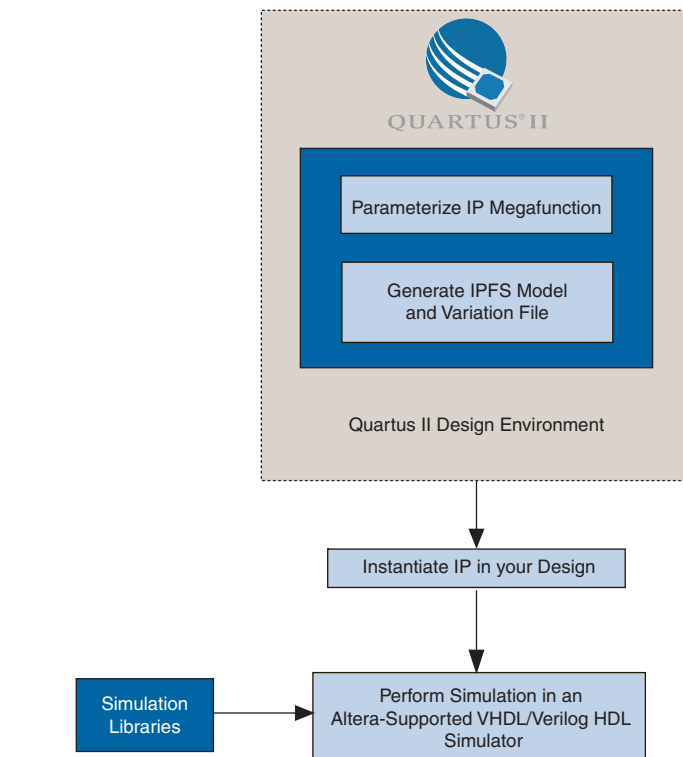
IP メガファンクションの MegaWizard[®] インタフェースを使用すると、ドキュメントの表示、パラメータの指定、IP 機能シミュレーション (IPFS) モデルの生成、パラメータ化された IP メガファンクションのデザインに統合するのに必要なファイルの出力をすばやく簡単に行うことができます。Quartus II ソフトウェア内で、MegaWizard Plug-In Manager を使用して、IP メガファンクションを選択し、パラメータ化することができます。Quartus II ソフトウェアは、Quartus II プロジェクトに含まれる IP メガファンクションのバリエーション・ファイルを生成します。IPFS モデルを必要とする IP メガファンクションの場合、ユーザがメガファンクションをパラメータ化した後で、Quartus II ソフトウェアは、レジス

タ転送レベル (RTL) IPFS モデルを含む Verilog 出力ファイル (.vo) または VHDL 出力ファイル (.vho) を生成することもできます。IPFS モデルは Quartus II プロジェクト・ディレクトリに書き込まれます。

ほとんどのアルテラ・メガファンクションと IP MegaCore ファンクションは、アルテラがサポートするすべてのサードパーティ・シミュレータに対して、Verilog および VHDL での機能シミュレーションをサポートしています。IP メガファンクションをシミュレートするには、シミュレーション・ライブラリが必要です。Quartus II ソフトウェアに付属するシミュレーション・ライブラリのサブセットについては、12 ページの表 5-2 を参照してください。

図 5-1 に、サードパーティ・シミュレータによるアルテラ IP の標準的なシミュレーション・フローを示します。

図 5-1. IP 機能シミュレーション (IPFS) モデルのデザイン・フロー



Verilog および VHDL IP 機能シミュレーション (IPFS) モデル

IP メガファンクションによっては、機能シミュレーションをサポートするために IPFS モデルを必要とします。これらの IPFS モデルは、高レベルのレジスタ転送レベル (RTL) HDL で記述されています。これらの Verilog または VHDL フォーマットの高レベル RTL モデルは、Quartus II ソフトウェアによって合成後または配置配線後のシミュレーション用に生成される、Verilog または VHDL フォーマットの低レベルの合成ネットリストとは異なります。Quartus II ソフトウェアによって生成される IPFS モデルは、加算器、乗算器、マルチプレクサなど、さらに高いレベルのプリミティブにマップされるため、デザインの低レベルの合成後または配置配線後ネットリストより、はるかに高速です。これらの IPFS モデルは、デザインの残りの部分とともに、アルテラがサポートする任意のシミュレータでシミュレートできます。IPFS モデルは、IP メガファンクションのバリエーション・ファイルのハードウェア言語と同じハードウェア言語で生成することを推奨します。



IPFS モデルは、シミュレーションにのみ使用でき、合成やその他の目的には使用できません。IPFS モデルを合成すると、機能しないデザインが作成されます。



アルテラ MegaCore ファンクション用の IPFS モデルの生成にはライセンスは必要ありません。しかし、AMPP メガファンクション用に IPFS モデルを生成する場合は、ライセンスが必要となります。ライセンスの必要性については、IP メガファンクションのベンダにお問い合わせください。

IP のパラメータ化および生成方法については、該当する IP ユーザ・ガイドを参照してください。

デザイン での IP の インスタンス 化

デザイン内の各 IP メガファンクションについて、デザイン内の対応するエンティティまたはモジュールをインスタンス化する必要があります。各 IP メガファンクション・エンティティまたはモジュール名は、Quartus II によって生成されるメガファンクションのバリエーション・ファイルに定義されます。デザイン内の IP メガファンクションをインスタンス化した後は、合成またはシミュレーションのためにデザインを編集する必要はありません。

Quartus II ソフトウェアを使用してデザインを合成するには、Quartus II で生成された Verilog HDL または VHDL バリエーション・ファイルを Quartus II プロジェクトに追加します。Quartus II プロジェクト用の新

しいバリエーション・ファイルを作成すると、それらのファイルはメガファンクションの生成時に、現在開かれているプロジェクトに追加されます。

サードパーティ製 EDA ツールを使用してデザインを合成するには、VHDL デザインでは Quartus II で生成された CMP ファイル (<メガファンクション・バリエーション >.cmp) を、Verilog HDL デザインでは Verilog HDL ブラック・ボックス・ファイル (<メガファンクション・バリエーション>_bb.v) をサードパーティ合成プロジェクトに追加します。

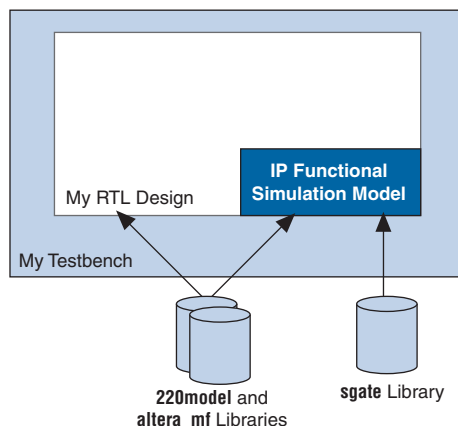


Quartus II ソフトウェアによる合成およびコンパイルについて詳しくは、「Quartus II ハンドブック Volume 1」の該当する章を参照してください。

シミュレーションの実行

シミュレーションを実行するには、デザイン・ファイルとテストベンチ・ファイルを追加するのに加えて、IP メガファンクションのバリエーション・ファイルまたは IPFS モデルをシミュレーション・プロジェクトに追加する必要があります。IP メガファンクションにシミュレーション用の IPFS モデルが必要ない場合は、メガファンクションのバリエーション・ファイルをシミュレーション・プロジェクトに追加します。シミュレートする IP メガファンクションに IPFS モデルが必要な場合は、IPFS モデルをシミュレーション・プロジェクトに追加します。シミュレーション・プロジェクトでシミュレーションを正常に行うには、アルテラが提供するライブラリも必要です。図 5-2 に、IP 機能シミュレーションでアルテラのライブラリがどのように使用されるかを示します。

図 5-2. IP 機能シミュレーション・ライブラリの用途



Quartus II ソフトウェアには、アルテラ IP の正常なシミュレーションを設定および実行するのに必要なライブラリがすべて含まれています。使用する IP メガファンクションが Quartus II NativeLink をサポートしている場合は、Quartus II NativeLink 機能を使用してシミュレーションを設定できます。IP メガファンクションが Quartus II ソフトウェアの NativeLink 機能をサポートしているかどうかを判断するには、該当する IP メガファンクションのユーザ・ガイドを参照してください。あるいは、サードパーティ・シミュレータを使用してアルテラ IP を直接シミュレートすることもできます。

Quartus II NativeLink 機能を使用したアルテラ IP のシミュレーション

Quartus II NativeLink 機能により、シミュレーションの設定および実行作業が容易になります。NativeLink 機能を使用すると、Quartus II ソフトウェアからサードパーティ・シミュレータを起動して、シミュレーションを実行できます。NativeLink 機能により、テストベンチのコンパイルおよびシミュレーションが自動化されます。

次に、Quartus II NativeLink 機能を使用して、サードパーティ・シミュレータで IP メガファンクションをシミュレートするステップを簡単に説明します。これらの各ステップについては、以降の項で詳しく説明します。

1. Quartus II プロジェクトの設定
2. サードパーティ・シミュレーション・ツールの選択
3. サードパーティ・シミュレータのパスの指定
4. テストベンチ設定の指定
5. Quartus II プロジェクトの分析および詳細設定
6. RTL 機能シミュレーションの実行

Quartus II プロジェクトの設定

Quartus II NativeLink 機能を使用して IP メガファンクションをシミュレートするには、Quartus II ソフトウェアで既存のプロジェクトを開くか、新しいプロジェクトを作成する必要があります。Quartus II ソフトウェアで MegaWizard Plug-In Manager を使用して、デザインで使用する IP を作成してパラメータ化できます。IP を作成してパラメータ化すると、Quartus II プロジェクトにアルテラ IP メガファンクションのバリ

エーション・ファイルが追加されます。Quartus II プロジェクトには、必要な他のデザイン・ファイルも追加できます。Quartus II NativeLink 機能を使用する場合、Quartus II プロジェクトにシミュレーション用の IPFS モデルを必要とする IP メガファンクションが含まれていても、それらの IP メガファンクションのために Quartus II プロジェクトに IPFS モデルを手動で追加する必要はありません。Quartus II NativeLink 機能によってサードパーティ・シミュレーション・ツールが起動しシミュレーションが開始されると、シミュレーションに必要な IPFS モデル・ファイルが Quartus II プロジェクト・ディレクトリに存在する限り、それらのファイルが自動的に追加されます。

サードパーティ・シミュレーション・ツールの選択

図 5-3 に示すように、Project Settings メニューからサードパーティ・シミュレーション・ツールを選択することができます。

図 5-3. サードパーティ・シミュレーション・ツールの選択

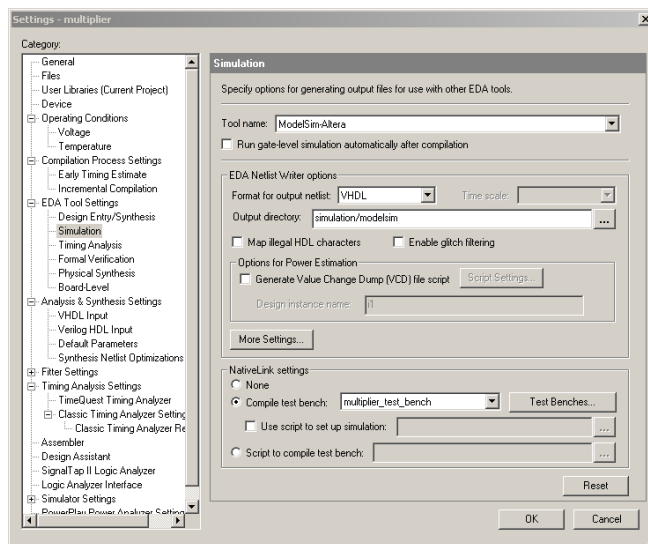


表 5-1 に、Quartus II NativeLink 機能でサポートされるサードパーティ・シミュレータを示します。

サードパーティ・シミュレータ	Quartus II から 起動可能	テストベンチ・ サポート	混在デザイン (Verilog および VHDL)
ModelSim PE/SE	○	○	○
ModelSim Altera Edition	○	○	×
Synopsys VCS	○ (1)	○	×
Synopsys VCS-MX	○ (1)	○	○
Cadence NC-Sim	○ (1)	○	○
Aldec	○	○	○

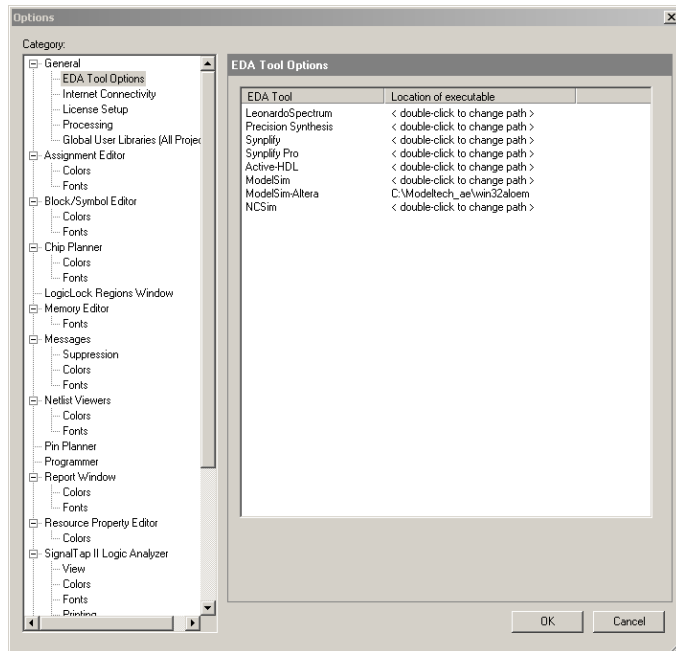
表 5-1 の注：

- (1) シミュレータを UNIX または Linux プラットフォーム上で実行する場合、シミュレーション・ツールを起動するには、Quartus II ソフトウェアが同じプラットフォーム上で実行されている必要があります。

サードパーティ・シミュレータのパスの指定

サードパーティ・シミュレーション・ツールを起動するには、Tools メニューの **Options** ページに、選択するシミュレータの絶対パスを指定する必要があります。図 5-4 を参照してください。Location of executable フィールドをダブルクリックして、絶対パスを変更または指定します。

図 5-4. シミュレータのパスの指定

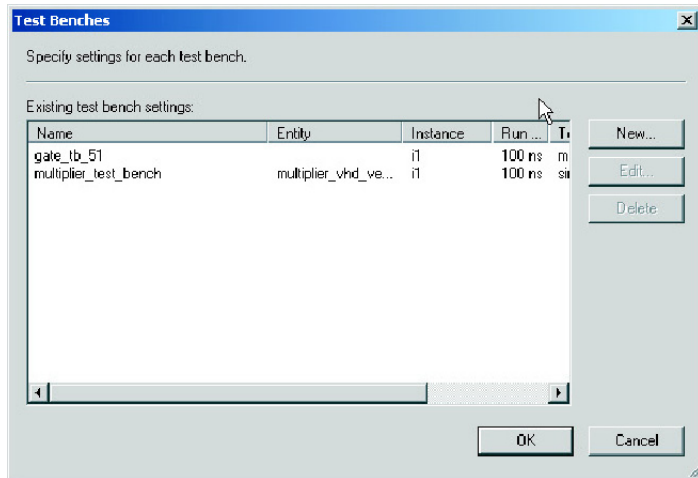


テストベンチ設定の指定

該当するテストベンチ設定を次のとおり指定します。

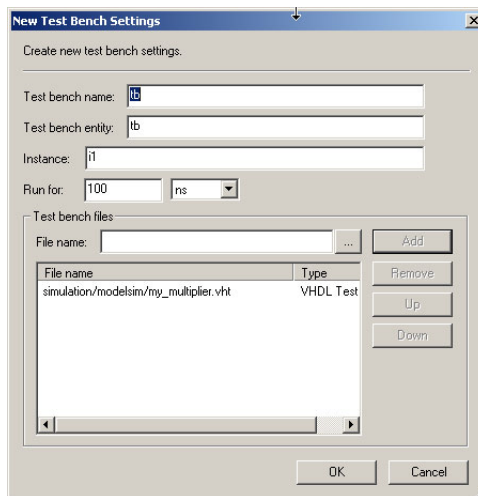
1. **Settings** ダイアログ・ボックスの **NativeLink Settings** で (図 5-3)、**Compile Test Bench** ラジオ・ボタンを選択し、**Test Benches** をクリックして **Test Benches** ダイアログ・ボックスを表示します。図 5-5 を参照してください。

図 5-5. Test Bench ダイアログ・ボックス



2. **New** をクリックして、**New Test Bench Settings** ダイアログ・ボックスを表示します (図 5-6 に示す)。

図 5-6. New Test Bench Settings ダイアログ・ボックス



3. **New Test Bench Settings** ダイアログ・ボックスで、該当するフィールドにテストベンチの名前を設定します。



MegaCore メガファンクションのテストベンチ設定を指定するための具体的な指示については、MegaCore ファンクション・ユーザ・ガイドを参照してください。

4. テストベンチ・ファイルを指定したら、**New Test Bench Settings**、**Test Benches**、および **Settings** ダイアログ・ボックスを閉じます。

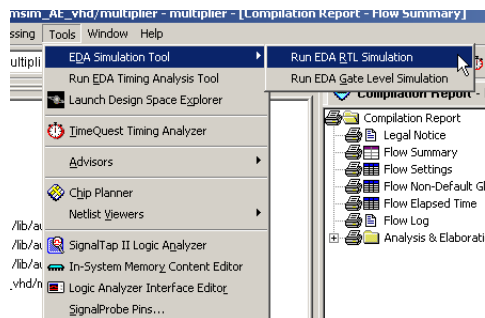
Quartus II プロジェクトの分析および詳細設定

NativeLink 機能を使用してシミュレーションを開始する前に、各 IP メガファンクションのバリエーション・ファイルがデザイン・プロジェクトに含まれていることを確認します。Quartus II Processing メニューで、**Start** をポイントし、**Start Analysis & Elaboration** をクリックします。

RTL 機能シミュレーションの実行

デザインの分析と詳細設定を実行した後、Tools メニューの **Run EDA-RTL Simulation** をクリックしてシミュレーションを開始できます。図 5-7 を参照してください。RTL 機能シミュレーションでは、IPFS モデルはシミュレータによってコンパイルされて使用されます。

図 5-7. NativeLink を使用した、IP の機能シミュレーションの実行



Quartus II NativeLink 機能を使用しないアルテラ IP のシミュレーション

アルテラ IP は、サードパーティ・シミュレータを使用して直接シミュレートすることもできます。デザインで IP メガファンクションをインスタンス化する場合は、そのバリエーション・ファイルをシミュレーション・プロジェクトに追加します。IP メガファンクションに IPFS モデル・ファイルが必要な場合は、メガファンクションのバリエーション・ファイルをシミュレーション・プロジェクトに追加しないでください。バリエーション・ファイルではなく、IPFS モデル・ファイル (Verilog または VHDL のいずれか) をシミュレーション・プロジェクトに追加します。Quartus II ソフトウェアによって生成される IPFS モデルは、加算器、乗算器、およびマルチプレクサなどの高レベル・プリミティブはもとより、LPM (Library of Parameterized Modules) ファンクションやアルテラ・メガファンクションもインスタンス化します。

IP メガファンクションを適切にコンパイル、ロード、およびシミュレートするには、まずシミュレーション・ツールの以下のライブラリをコンパイルする必要があります。

- **sgate**—高レベルのプリミティブの定義が含まれます (IPFS モデルに必要)。
- **altera_mf**—アルテラ・メガファンクションの定義が含まれます。
- **220model**—LPM ファンクションの定義が含まれます。

これらのライブラリ・ファイルは、アルテラがサポートする任意のシミュレーション・ツールで使用することができます。ModelSim® アルテラ・ソフトウェアを使用する場合、ライブラリは事前にコンパイルおよびマップされています。



Nios® プロセッサまたは Avalon® ペリフェラルを含むデザインをシミュレートするには、「AN 189 Simulating Nios Embedded Processor Designs」を参照してください。

表 5-2 は、シミュレーション・ライブラリ・ファイルの一覧であり、`<path>` は Quartus II ソフトウェアがインストールされているディレクトリです。

表 5-2. シミュレーション・ライブラリ・ファイル		
場所	HDL 言語	説明
<path>leda/sim_lib/sgate.v	Verilog HDL	IP 機能モデル用のシミュレーション・モデルを含むライブラリ
<path>leda/sim_lib/sgate.vhd	VHDL	
<path>leda/sim_lib/sgate_pack.vhd	VHDL	sgate.vhd ライブラリに対する VHDL コンポーネント宣言を含むライブラリ
<path>leda/sim_lib/220model.v	Verilog HDL	アルテラ LPM version 2.2.0 のシミュレーション・モデルを含むライブラリ
<path>leda/sim_lib/220model.vhd	VHDL	
<path>leda/sim_lib/220pack.vhd	VHDL	220model.vhd ライブラリに対する VHDL コンポーネント宣言を含むライブラリ
<path>leda/sim_lib/altera_mf.v	Verilog HDL	アルテラ固有のメガファンクションのシミュレーション・モデルを含むライブラリ
<path>leda/sim_lib/altera_mf.vhd	VHDL	
<path>leda/sim_lib/altera_mf_components.vhd	VHDL	altera_mf.vhd ライブラリに対する VHDL コンポーネント宣言を含むライブラリ

デザイン言語 の例

以下のデザイン言語の例では、サードパーティ・シミュレーション・ツールを使用してIPメガファンクションを直接シミュレートする方法を説明します。これらのデザイン例は、以下を使用したシミュレーションを示しています。

- ModelSim Verilog
- ModelSim VHDL
- NC-VHDL
- VCS

Verilog HDL の例 : ModelSim ソフトウェアの IPFS モデルのシミュレーション

以下の例では、Verilog HDL ベース・メガファンクションのシミュレーションのプロセスを示します。この例では、メガファンクション・バリエーションと IPFS モデルが生成されるものと仮定しています。

1. 以下のステップを実行し、ModelSim プロジェクトを作成します。
 - a. ModelSim ソフトウェアで、File メニューから New をポイントし、Project をクリックします。Create Project ダイアログ・ボックスが表示されます。
 - b. シミュレーション・プロジェクトの名前を指定します。
 - c. シミュレーション・プロジェクトの場所を指定します。
 - d. デフォルトのライブラリ名を指定し、OK をクリックします。
 - e. シミュレーション・プロジェクトに以下の関連ファイルを追加します。
 - ユーザのデザイン・ファイル
 - Quartus II ソフトウェアで生成された IPFS モデル (ModelSim-Altera ソフトウェアを使用している場合は、ステップ 5 に進みます)
 - sgate.v、220model.v、および altera_mf.v ライブラリ・ファイル
2. ModelSim プロンプトで以下のコマンドを入力し、必要なシミュレーション・ライブラリを作成します。

```
vlib sgate ↓
```

```
vlib lpm ↓
```

```
vlib altera_mf ↵
```

3. ModelSim プロンプトで以下のコマンドを入力し、必要なシミュレーション・ライブラリにマップします。

```
vmap sgate sgate ↵
```

```
vmap lpm lpm ↵
```

```
vmap altera_mf altera_mf ↵
```

4. ModelSim プロンプトで以下のコマンドを入力し、HDL をライブラリにコンパイルします。

```
vlog -work altera_mf altera_mf.v ↵
```

```
vlog -work sgate sgate.v ↵
```

```
vlog -work lpm 220model.v ↵
```

5. ModelSim プロンプトで以下のコマンドを入力し、IPFS モデルをコンパイルします。

```
vlog -work work <my_IP>.vo ↵
```

6. ModelSim プロンプトで以下のコマンドを入力し、RTL をコンパイルします。

```
vlog -work work <my_design>.v ↵
```

7. ModelSim プロンプトで以下のコマンドを入力し、テストベンチをコンパイルします。

```
vlog -work work <my_testbench>.v ↵
```

8. ModelSim プロンプトで以下のコマンドを入力し、テストベンチをロードします。

```
vsim -L <altera_mf_library_path> -L <lpm_library_path>  
-L <sgate_library_path> work.<my_testbench> ↵
```

VHDL の例 : ModelSim ソフトウェアの IPFS モデルのシミュレーション

以下の例では、VHDL ベース・メガファンクション IPFS モデルの機能シミュレーションを実行するプロセスを示します。この例では、メガファンクション・バリエーションと IPFS モデルが生成されるものと仮定しています。

1. 以下のステップを実行し、ModelSim プロジェクトを作成します。
 - a. ModelSim ソフトウェアで、File メニューから New をポイントし、Project をクリックします。Create Project ダイアログ・ボックスが表示されます。
 - b. シミュレーション・プロジェクトの名前を指定します。
 - c. シミュレーション・プロジェクトの場所を指定します。
 - d. デフォルトのライブラリ名を指定し、OK をクリックします。
 - e. シミュレーション・プロジェクトに以下の関連ファイルを追加します。
 - ユーザのデザイン・ファイルを追加します。
 - Quartus II ソフトウェアで生成された IPFS モデルを追加します (ModelSim-Altera ソフトウェアを使用している場合は、ステップ 5 に進みます)。
 - **sgate.vhd**、**sgate_pack.vhd**、**220model.vhd**、**220pack.vhd**、**altera_mf.vhd**、および **altera_mf_components.vhd** ライブラリ・ファイルを追加します。
2. ModelSim プロンプトで以下のコマンドを入力し、必要なシミュレーション・ライブラリを作成します。

```
vlib sgate ↓
```

```
vlib lpm ↓
```

```
vlib altera_mf ↓
```

3. ModelSim プロンプトで以下のコマンドを入力し、必要なシミュレーション・ライブラリにマップします。

```
vmap sgate sgate ↓
```

```
vmap lpm lpm ↓
```

```
vmap altera_mf altera_mf ↵
```

4. ModelSim プロンプトで以下のコマンドを入力し、HDL をライブラリにコンパイルします。

```
vcom -work altera_mf -93 -explicit  
altera_mf_components.vhd ↵
```

```
vcom -work altera_mf -93 -explicit altera_mf.vhd ↵
```

```
vcom -work lpm -93 -explicit 220pack.vhd ↵
```

```
vcom -work lpm -93 -explicit 220model.vhd ↵
```

```
vcom -work sgate -93 -explicit sgate_pack.vhd ↵
```

```
vcom -work sgate -93 -explicit sgate.vhd ↵
```

5. ModelSim プロンプトで以下のコマンドを入力し、IPFS モデルをコンパイルします。

```
vcom -work work -93 -explicit <output_netlist>.vho ↵
```

6. ModelSim プロンプトで以下のコマンドを入力し、RTL をコンパイルします。

```
vcom -work work -93 -explicit <RTL>.vhd ↵
```

7. ModelSim プロンプトで以下のコマンドを入力し、テストベンチをコンパイルします。

```
vcom -work work -93 -explicit <my_testbench>.vhd ↵
```

8. ModelSim プロンプトで以下のコマンドを入力し、テストベンチをロードします。

```
vsim work.my_testbench ↵
```

NC-VHDL の例 : NC-VHDL ソフトウェアの IPFS モデルのシミュレーション

以下の例では、NC-VHDL ベース・メガファンクション IP 機能シミュレーション・モデルの機能シミュレーションを実行するプロセスを示します。この例では、メガファンクション・バリエーションと IPFS モデルが生成されるものと仮定しています。

1. 以下のエントリを入力して、**cds.lib** ファイルを作成します。

```
DEFINE worklib ./worklib

DEFINE sgate ./sgate

DEFINE altera_mf ./altera_mf

DEFINE lpm ./lpm
```

2. コマンド・プロンプトで以下のコマンドを入力し、ライブラリ・ファイルを適切なライブラリにコンパイルします。

```
ncvhdl -V93 -WORK lpm 220pack.vhd ↵

ncvhdl -V93 -WORK lpm 220model.vhd ↵

ncvhdl -V93 -WORK altera_mf
altera_mf_components.vhd ↵

rncvhdl -V93 -WORK altera_mf altera_mf.vhd ↵

ncvhdl -V93 -WORK sgate sgate_pack.vhd ↵

ncvhdl -V93 -WORK sgate sgate.vhd ↵
```

3. コマンド・プロンプトで以下のコマンドを入力し、ソース・コードとテストベンチ・ファイルをコンパイルします。

```
ncvhdl -V93 -WORK worklib <my_design>.vhd ↵

ncvhdl -V93 -WORK worklib <my_testbench>.vhd ↵

ncvhdl -V93 -WORK worklib
<my_IPtoolbench_output_netlist>.vho ↵
```

4. コマンド・プロンプトで以下のコマンドを入力し、デザインを綿密に調整します。

```
ncelab worklib.<my_testbench>:entity ↓
```

Verilog HDL の例 : VCS の IPFS モデルのシミュレーション

以下の例では、Verilog HDL ベース・メガファンクション IPFS モデルを含むデザインの機能シミュレーションを実行するプロセスを示します。この例では、メガファンクション・バリエーションと IPFS モデルが生成されるものと仮定しています。

シングル・ステップ・プロセス

シングル・ステップ・プロセスでは、コマンド・プロンプトで次のように入力します。

```
vcs <testbench>.v <RTL>.v <output_netlist>.v -v 220model.v  
altera_mf.v sgate.v -R ↓
```

2 ステップ・プロセス (コンパイルおよびシミュレーション)

コンパイルとシミュレーションでは、以下のステップを実行します。

1. コマンド・プロンプトで以下のように入力し、デザイン・ファイルをコンパイルします。

```
vcs <testbench>.v <RTL>.v <output_netlist>.v -v 220model.v  
altera_mf.v sgate.v -o simulation_out ↓
```

2. コマンド・プロンプトで以下のように入力して、シミュレーションをロードします。

```
source simulation_out ↓
```

VCS でのデザインのシミュレーションについて詳しくは、「Quartus II ハンドブック Volume 3」の「Synopsys VCS Support」の章を参照してください。

まとめ

アルテラの Quartus II ソフトウェアは、サードパーティ・ツールを使用した直接または NativeLink 機能による IP メガファンクションのシミュレーションを完全にサポートしています。Quartus II ソフトウェアを使用すると、サポートされているメガファンクション用の IPFS モデルを生成して、デザイン検証を強化および簡素化することも可能です。IPFS モデルの使用時にユーザは何も意識する必要はなく、プロジェクトを合成およびシミュレートするための別のファイルを追加するだけです。

参考資料

この章では以下のドキュメントを参照しています。

- 「Quartus II ハンドブック Volume 3」の「Synopsys VCS Support」
- 「Quartus II ハンドブック Volume 1」

改訂履歴

表 5-3 に、本資料の改訂履歴を示します。

表 5-3. 改訂履歴		
日付 & ドキュメント・バージョン	変更内容	概要
2007 年 10 月 v7.2.0	5-19 ページの「参考資料」を再編成。	—
2007 年 5 月 v7.1.0	<ul style="list-style-type: none"> ● 図 5-6 を更新。 ● 「参考資料」の項を追加。 	Quartus II ソフトウェア・バージョン 7.1 のためのマイナー・アップデート。
2007 年 3 月 v7.0.0	Quartus II ソフトウェア 7.0 のリビジョンおよび日付のみ更新。	—
2006 年 11 月 v6.1.0	<ul style="list-style-type: none"> ● Quartus II NativeLink 機能の情報を追加 ● 図 5-1 および 5-2 を更新 ● 図 5-3、5-4、5-7 を追加 ● 表 5-1 を追加 	Quartus II NativeLink 機能のサポートによる章の更新。
2006 年 5 月 v6.0.0	Quartus II ソフトウェア・バージョン 6.0 のためのマイナー・アップデート。	—
2005 年 10 月 v5.1.0	Quartus II ソフトウェア・バージョン 5.1 のための更新。	—
2005 年 5 月 v5.0.0	バージョン 4.2 の Vol 3 のセクション I を 4 章に変更。	—
2004 年 12 月 v1.0.0	初版	—