



Spansion シミュレーションモデル

ユーザーマニュアル

Version 1.0

目次

1	変更履歴.....	3
2	はじめに.....	4
3	VHDL/Verilog モデル	5
3.1	提供ファイル.....	5
3.1.1	VHDL シミュレーションモデル	5
3.1.2	Verilog シミュレーションモデル	5
3.1.3	プリロードファイル	5
3.1.4	プリロードファイルの作成.....	6
3.2	モデルファイルのコンパイル	7
3.2.1	VHDL モデルのコンパイル.....	7
3.2.2	Verilog モデルファイルのコンパイル.....	7
3.3	モデルシミュレーション.....	8
3.3.1	基本的なシミュレーション.....	8
3.3.2	SDF ファイルの作成	8
4	サポート情報	10

1 変更履歴

版数/日付	変更履歴
1.0/2004 年 1 月 23 日	初版

2 はじめに

SPANSION フラッシュメモリシミュレーションモデルには、以下の特徴があります。

- High/Low & Top/Bottom の各ブートタイプをサポート
- Verilog / Vhdl 言語をサポート
- 配線遅延を含んだ正確なタイミングでのシミュレーションが可能
- タイミングチェック機能内蔵
- 各設定をプリロードすることが可能
 - 保護セクタの設定
 - メモリデータの初期設定
 - Secure Silicon のデータの設定
 - トップ/ボトム ブートタイプの設定

サイトからダウンロードした EXE ファイルをインストール後に、Figure 1 のようなファイル構成になっていることをご確認ください。

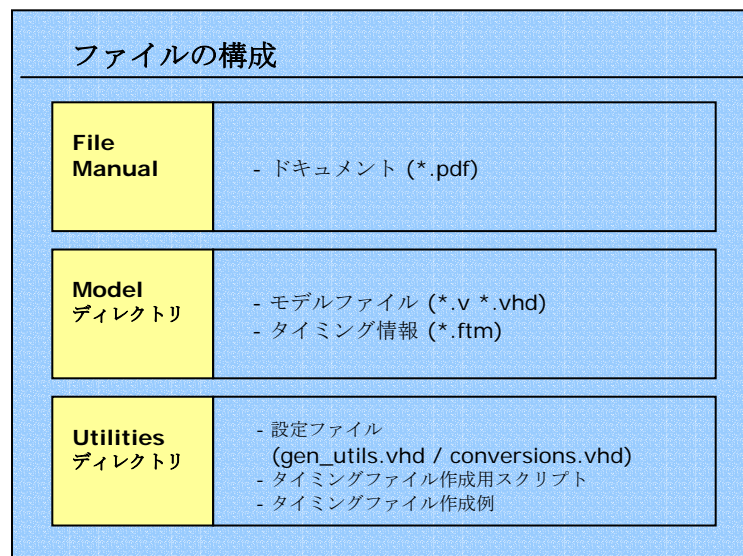


Figure 1 ファイル構成

3 VHDL/Verilog モデル

3.1 提供ファイル

3.1.1 VHDL シミュレーションモデル

インストールしたディレクトリ内の“model”ディレクトリにある model.vhd がモデルファイルとなります。モデルファイルは、gen_utils.vhd と conversions.vhd によって定義されます。これら3つのファイルがシミュレーションにおける最小ファイル構成となります。

3.1.2 Verilog シミュレーションモデル

インストールしたディレクトリ内の“model”ディレクトリにある model.v がモデルファイルとなります。Verilog シミュレーションでは他のファイルは必要なく、このファイルがシミュレーションの為の最小構成となります。

3.1.3 プリロードファイル

短時間でシミュレーションを行う為に、SPANSION シミュレーションモデルにはプリロード機能が用意されています。あらかじめメモリの初期データが決まっている場合には、プリロード機能を使うことによりメモリにデータをプログラムする時間を省略できます。プリロード機能は、以下の2項目を設定することで使用可能となります。

- UserPreload
- preload_file_name(s)
 - mem_file_name --- メモリデータの初期設定
 - prot_file_name --- 保護セクタの設定
 - secsi_file_name --- Secure Silicon のデータの設定

VHDL の場合 top_level.vhd で、Verilog の場合は model.v でこれらの設定を行います。デフォルトではプリロード機能は使用しない設定となっている為、これらの設定を行わずにシミュレーションをすることも可能です。

異なる初期条件でシミュレーションを行う場合には、Figure 2のようなファイル構成にすることも可能です。

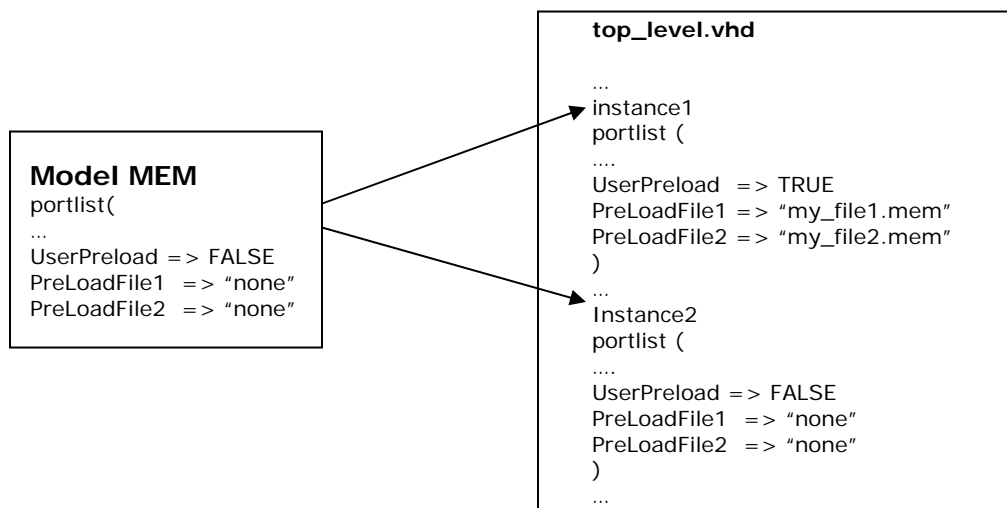


Figure 2 異なる初期条件におけるプリロードファイルの読み込み

3.1.4 プリロードファイルの作成

メモリデータ、CFI、保護セクタ情報などは、プリロードファイルからメモリにロードされます。初期設定ではメモリデータ 0xFFFF、データシートと同じ CFI、保護セクタなしとなっていますが、プリロードファイルを作成することにより、これらの初期設定は変更可能です:

プリロードファイルによるメモリ/Secure Silicon のデータの設定例

```
// select Addr0: ADDR_0= AA, ADDR_1= 55, ADDR_2= 11
// select AddrA8: ADDR_8= 01, ADDR_9= 02, ADDR_10= 03
@00000
AA
55
11
@000A8
01
02
03
```

プリロードファイルによる保護セクタの設定例

```
// select sector 01, 19, EE: set 1 to protect
@01
1
@19
1
@EE
1
```

**** プリロードファイルには空行を入れないで下さい
**** アドレスの"0"は省略不可能です

3.2 モデルファイルのコンパイル

3.2.1 VHDL モデルのコンパイル

ファイルのコンパイル先は以下のように設定します:

- conversions.vhd : ライブラリ"FMF"へコンパイル
- gen_utils.vhd: ライブラリ"FMF"へコンパイル
- model.vhd: ライブラリ"work"へコンパイル

model.vhd をコンパイルする前に、conversion.vhd と gen_utils.vhd がコンパイルされている必要があります。ライブラリ"FMF"を作成し、これらの2 ファイルをライブラリ"FMF"へコンパイルした後で model.vhd のコンパイルを実行してください。

3.2.2 Verilog モデルファイルのコンパイル

Verilog の場合、ライブラリを設定する必要はありません。モデルファイル (model.v) はデフォルトのままコンパイル可能となっています。

3.3 モデルシミュレーション

3.3.1 基本的なシミュレーション

クリティカルなタイミングを使用しないシミュレーションは、デフォルトのファイル構成で行えます。単純にシミュレーションを流して波形を見るだけであれば、新たにファイルを用意する必要はありません。

タイミング情報を記述した SDF ファイルを作成することで、より詳細なシミュレーションが可能となります。配線遅延を含めたシステムレベルのシミュレーションを、ASIC における一般的なシミュレーションフローと同一の手法で行えます。

3.3.2 SDF ファイルの作成

全てのスピードバージョンにおいて、“model”ディレクトリにある FTM ファイルから SDF ファイルを作成することができます。FTM ファイルから SDF ファイルを作成するには、以下の 2 通りの方法があります。:

1 FTM ファイルからコピー&ペースト (VHDL/Verilog)

FTM ファイル内から使用するスピードバージョンを探します。任意のスピードバージョンについて記述されている部分 (<TIMING>~</TIMING>の中身) をコピーし、utilities/CreateSDF ディレクトリにある SDF ファイルに上書きした後、保存します。

2 SDF ファイルの自動作成 (VHDL のみ、UNIX 環境)

各スピードバージョンに対応した FTM ファイルがある場合、Perl スクリプトを実行できる UNIX 環境が必要となりますが、FTM ファイルから SDF ファイルを自動的に作成することができます。utilities/CreateSDF ディレクトリにある Perl スクリプトは、最上位のネットリスト (top_level.vhd) を分析してタイミング情報を抽出し、SDF ファイルを作成します。この場合、各インスタンスはデータシートに準ずるタイミング情報を持つことが前提となります。(Figure 3 参照).

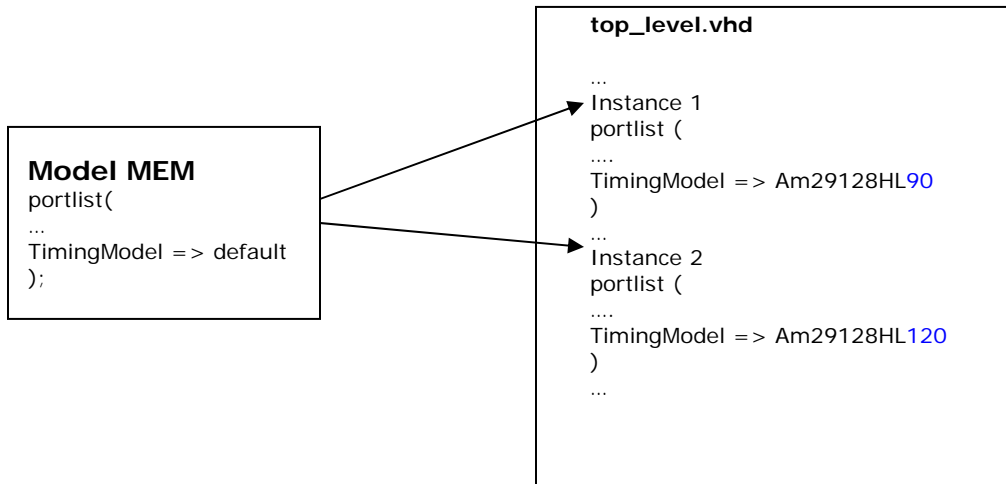


Figure 3 異なるスピードバージョンに対応したネットリストの例

Perl スクリプトは、タイミング属性 (TimingModel) に基づいて適切なタイミング情報を抽出し、SDF ファイルを作成します。コマンドラインで以下のコマンドを入力することにより、Perl スクリプトは実行されます。

```
perl mk_sdf_204.pl top_level.vhd
```

SDF ファイルを作成する為には、mk_sdf.cmd ファイル内に記述されている timing_dir のパスを正確に記述する必要があります。

****VHDL のみのモジュール構成である場合に限り使用可能です。

****UNIX 環境でのみ使用可能です。Windows 環境で実行したい場合は、サポートまでご連絡願います。

参考ファイル

utilities/CreateSDF に参考ファイルがあります。この場合は、Am29LV640M_vhd.ftm ファイルから toplevel.sdf が抽出されています。

4 サポート情報

ご質問やご不明な点がございましたら、下記メールアドレスまでお問い合わせ願います。

memorytools.help@spansion.com