

# SoC デバイス HPS 側 SDRAM の ModelSim シミュレーション方法

ver.14

# SoC デバイス HPS 側 SDRAM の ModelSim シミュレーション方法

## 目次

1. はじめに .....	3
2. シミュレーション手順 .....	3
2-1. Qsys でのハードウェア作成とテストベンチ生成 .....	3
2-2. シミュレーションに必要なファイルの作成 .....	5
2-3. ModelSim-Altera の起動からシミュレーションの実行まで .....	6
3. 詳細 .....	8
3-1. test_program.sv .....	8
3-2. top.sv .....	8
3-3. load_sim.tcl .....	9
改版履歴 .....	10

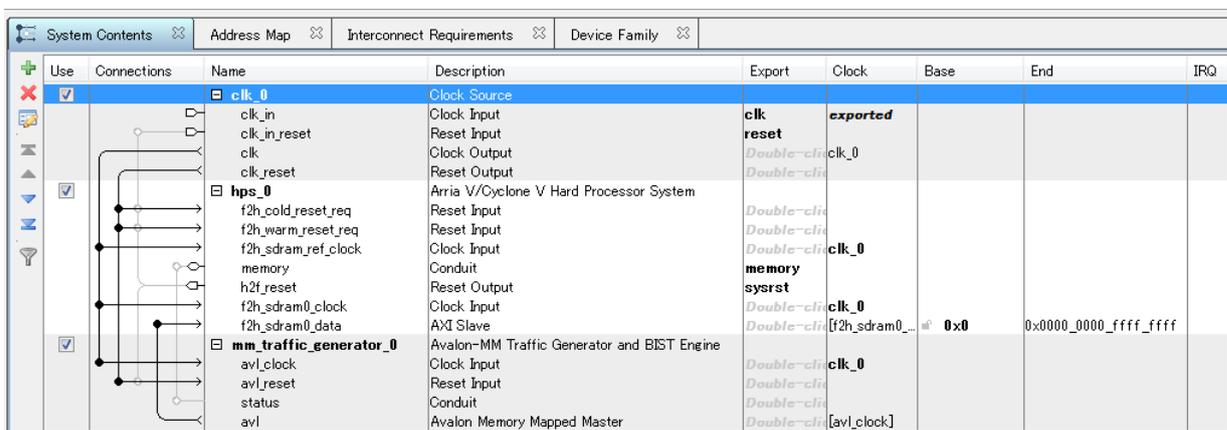
## 1. はじめに

この資料は、アルテラ社 SoC デバイスの Hard Processor System（以降、HPS と略）側に実装されている HPS Hard Memory Controller を ModelSim を使用してシミュレーションする方法について記載します。この資料では、Quartus® II 14.0、ModelSim-Altera 10.1e を使用して説明しています。ここでは、ModelSim-Altera を使用していますが、ModelSim-PE、ModelSim-SE でも同様にシミュレーションが可能です。

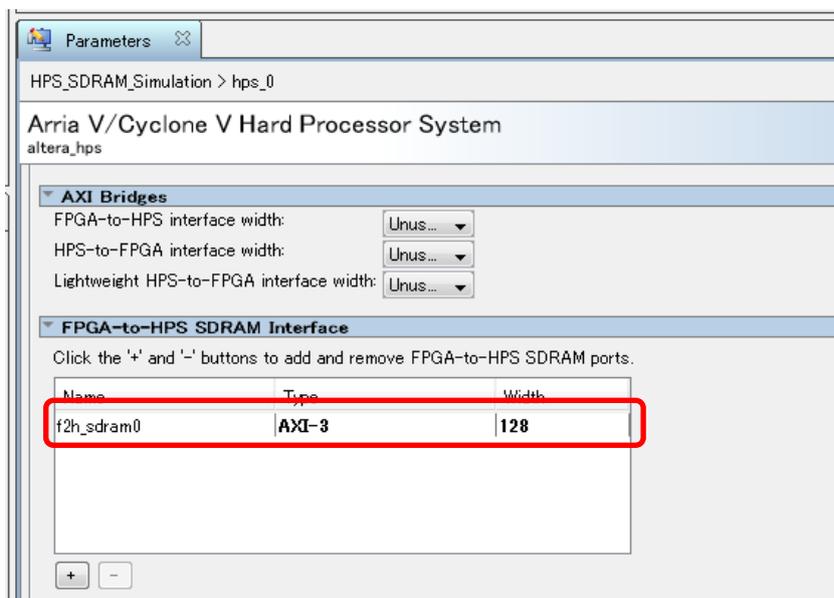
## 2. シミュレーション手順

### 2-1. Qsys でのハードウェア作成とテストベンチ生成

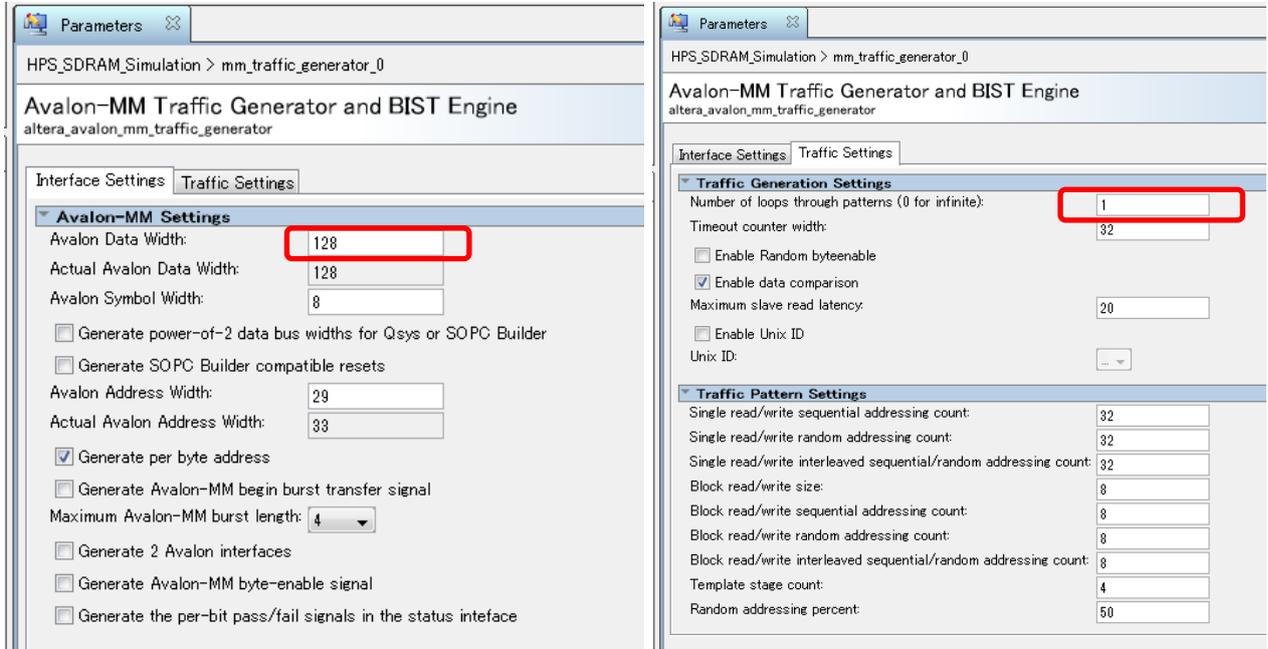
- ① ここでは、簡単な構成のシステムを想定します。HPS\_SDRAM\_Simulation という名前(任意)でシステムを Qsys にて生成します。下図のように Qsys 上でクロック/リセット、HPS、Avalon-MM Traffic Generator をインスタンスし、それぞれを適切に接続します。



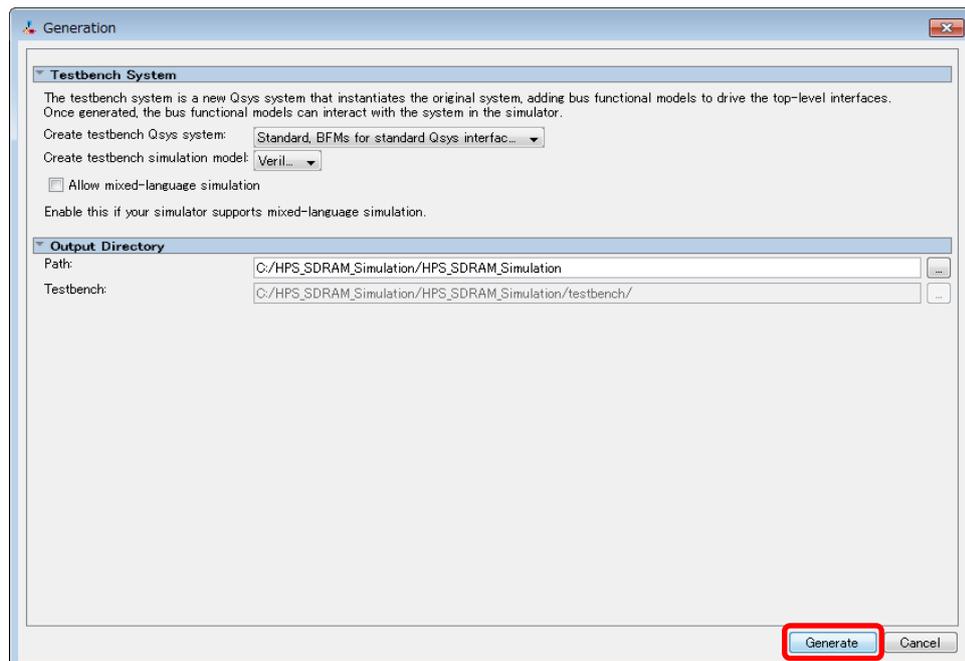
- ② HPS の設定については、下図のように、f2h\_sdrām0 を AXI-3、バス幅 128 ビットで設定します。Avalon-MM も選択可能ですが、ここでは、あえて AXI-3 を選択しています。



- ③ Avalon-MM Traffic Generator は、外部メモリなどのアクセス・テスト用に汎用的に使用できるテスト・パターン生成 & チェック機能を実現するモジュールとなります。今回は、下図のようにバス幅を 128 ビット、テスト・パターンの繰り返しを 1 回と設定しました。なお、このモジュールは、Avalon-MM となっているのに対して、HPS の f2h\_sdrām0\_data は、AXI Slave となっています。Qsys では、Avalon-MM ⇄ AXI の変換は、自動的に行ってくれるので、Avalon-MM Traffic Generator からの信号は、Qsys 内で AXI-3 に変換され f2h\_sdrām インタフェースに接続されます。



- ④ Generate メニュー ⇒ Generate Testbench System... を実行し、シミュレーションに必要なテストベンチ・ファイル一式を生成します。Generate 画面が起動するので、デフォルト設定のまま、Generate ボタンをクリックします。ここでは、言語を Verilog としています。Generate が完了したら Close ボタンをクリックします。



- ⑤ テストベンチの生成が完了すると、testbench フォルダの中には、下図のようなフォルダ/ファイル一式が生成されます。aldec、cadence、mentor、synopsys とあるフォルダは、各種シミュレーション用ツールを提供しているベンダごとのシミュレーション実行用スクリプト・ファイルが格納されています。本例では、Mentor Graphics 社の ModelSim を使用するので、mentor フォルダ内の msim\_setup.tcl スクリプトを使用します。HPS\_SDRAM\_Simulation\_tb フォルダ内には、テストベンチ・ファイルとシミュレーションに必要なライブラリ・ファイル一式が格納されています。

HPS\_SDRAM\_Simulation\_tb¥simulation¥HPS\_SDRAM\_Simulation\_tb.v  
がテストベンチ・ファイルとなります。



## 2-2. シミュレーションに必要なファイルの作成

これまでで、Qsys で自動生成できるファイルの準備ができましたが、ユーザが独自に手動で作成するファイルが必要となります。ここでは、各種ファイルの説明を簡単に行います。詳細については、後の章で後述します。なお、ここで出てくるフォルダ名、ファイル名はすべてユーザが任意に指定可能で、以下は、一例となります。

- ① test\_program.sv

SystemVerilog で記載したテスト・プログラムです。HPS に対するリセットを制御して、SDRAM インタフェースを有効にしています。

- ② top.sv

test\_program.sv と Qsys が自動生成したテストベンチ・ファイル HPS\_SDRAM\_Simulation\_tb.v をインスタンスしたテストベンチのトップファイルとなります。

- ③ load\_sim.tcl

ModelSim の実行用スクリプト・ファイルです。各種環境変数の設定や Qsys が自動生成した Mentor Graphics 社用スクリプト・ファイル msim\_setup.tcl の呼び出し、シミュレーションの実行を行います。

本例では、これらのファイルを user\_test\_program というフォルダに格納し、ここを ModelSim 実行用のカレント・フォルダに設定します。



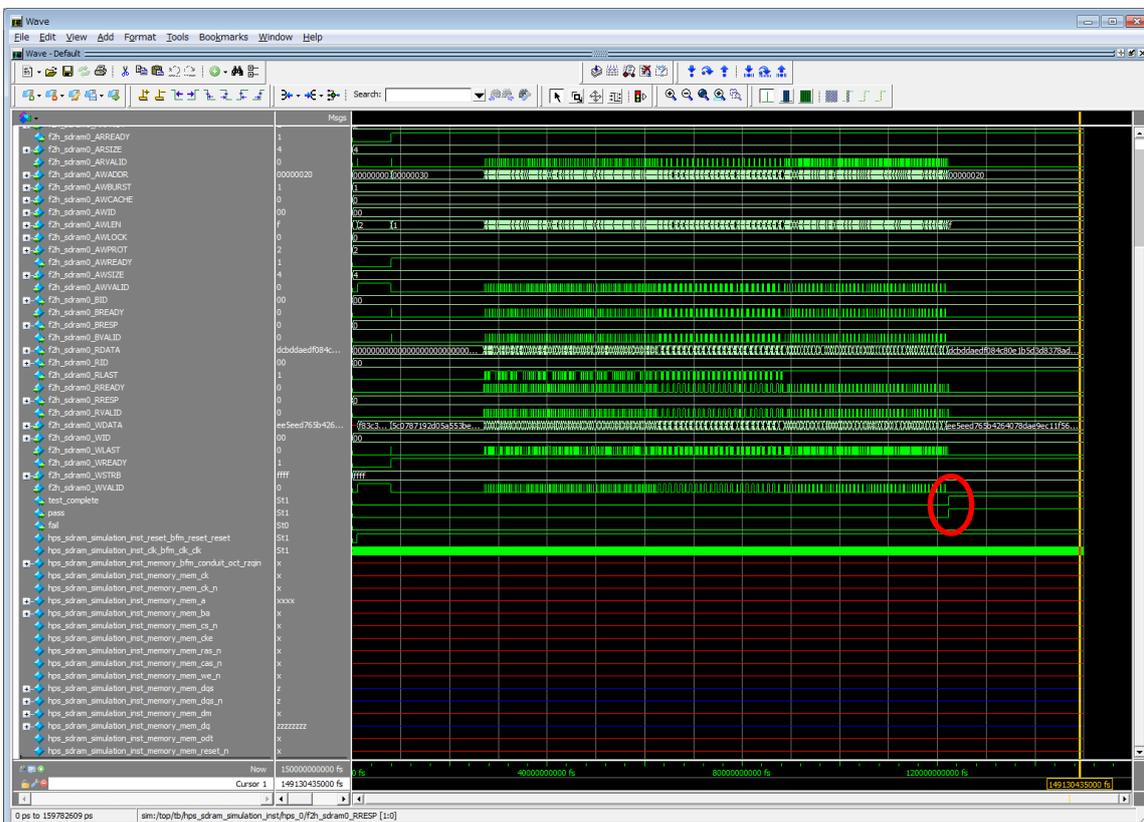
- ③ シミュレーションを実行します。ModelSim のコンソールから ”run 150us” を実行します。このコマンドで 150us のシミュレーションを実行します。完了までは、数分の時間を要します。完了すると以下のような画面となります。

```

Transcript
# [122132508] [DWR=000]: Refreshing bank 0
# [122150008] [DWR=000]: Reading data 78ad4e34 @ 4 (BRC=0/0/4 ) burst 0
# [122151258] [DWR=000]: Reading data 1b5d3d83 @ 5 (BRC=0/0/4 ) burst 1
# [122152508] [DWR=000]: Reading data f084c80e @ 6 (BRC=0/0/4 ) burst 2
# [122153758] [DWR=000]: Reading data dcbddaed @ 7 (BRC=0/0/4 ) burst 3
# [127935008] [DWR=000]: PRECHARGE - ALL BANKS
# [127952508] [DWR=000]: REFRESH Command
# [127952508] [DWR=000]: Refreshing bank 0
# [127952508] [DWR=000]: Refreshing bank 1
# [127952508] [DWR=000]: Refreshing bank 2
# [127952508] [DWR=000]: Refreshing bank 3
# [127952508] [DWR=000]: Refreshing bank 4
# [127952508] [DWR=000]: Refreshing bank 5
# [127952508] [DWR=000]: Refreshing bank 6
# [127952508] [DWR=000]: Refreshing bank 7
# [135755008] [DWR=000]: REFRESH Command
# [135755008] [DWR=000]: Refreshing bank 0
# [135755008] [DWR=000]: Refreshing bank 1
# [135755008] [DWR=000]: Refreshing bank 2
# [135755008] [DWR=000]: Refreshing bank 3
# [135755008] [DWR=000]: Refreshing bank 4
# [135755008] [DWR=000]: Refreshing bank 5
# [135755008] [DWR=000]: Refreshing bank 6
# [135755008] [DWR=000]: Refreshing bank 7
# [143557508] [DWR=000]: REFRESH Command
# [143557508] [DWR=000]: Refreshing bank 0
# [143557508] [DWR=000]: Refreshing bank 1
# [143557508] [DWR=000]: Refreshing bank 2
# [143557508] [DWR=000]: Refreshing bank 3
# [143557508] [DWR=000]: Refreshing bank 4
# [143557508] [DWR=000]: Refreshing bank 5
# [143557508] [DWR=000]: Refreshing bank 6
# [143557508] [DWR=000]: Refreshing bank 7
VSM: >
Now: 150 us Delta: 17 top
    
```

- ④ 波形の表示を行うと以下(一例)のようになります。Avalon-MM Traffic Generator の設定で、テスト・パターンの繰り返しを 1 回と指定したので、一連のシーケンスが完了した後に、test\_complete 信号が Hi となり処理が完了します。下図では、pass 信号が Hi となっているので、一連のテストが全て正常であることが確認できます。

※SDRAM 側のアクセス信号は、全て不定となっています。HPS 側の SDRAM のモデルは提供されないため、シミュレーション上では、すべて下図のように不定となります。これは仕様となりますので予めご了承ください。



### 3. 詳細

#### 3-1. test\_program.sv

以下が、test\_program.sv の本文となります。本例では、クロックの発行、リセットの制御、テスト用シーケンスの実行などは、HPS\_SDRAM\_Simulation\_tb.v 内の各インスタンスが行っているため、このプログラムで行うのは、HPS に対するリセット制御のみとなります。23 行から 25 行が該当の行となります。

```

1  `timescale 1ns/1ns
2
3  //Console messaging level
4  define VERBOSITY VERBOSITY_INFO
5
6  //BFM hierachy
7  define HPS tb.hps_sdram_simulation_inst
8
9
10 module test_program();
11
12     import verbosity_pkg::*;
13     import avalon_mm_pkg::*;
14
15     //-----
16     // Set verbosity before the test starts
17     // Qsys-generated testbench activates clock and reset BFM's
18     //-----
19     initial begin
20         set_verbosity(`VERBOSITY);
21
22         `HPS.hps_0.fpga_interfaces.h2f_reset_inst.reset_assert();
23         #100
24         `HPS.hps_0.fpga_interfaces.h2f_reset_inst.reset_deassert();
25     end
26
27
28 endmodule
29
    
```

#### 3-2. top.sv

以下が、top.sv の本文となります。test\_program.sv と Qsys が自動生成したテストベンチ・ファイル HPS\_SDRAM\_Simulation\_tb.v をインスタンスしたテストベンチのトップファイルとなります。

```

1 module top ();
2     HPS_SDRAM_Simulation_tb tb();
3     test_program pgm();
4 endmodule
    
```

### 3-3. load\_sim.tcl

以下が、load\_sim.tcl の本文となります。2 行目から 3 行目が環境変数の設定です。作成した環境に合わせて正しい値を設定します。7 行目で Qsys が自動生成した msim\_setup.tcl を呼び出しています。10 行目以降は、ライブラリのコンパイル、20 行目で vsim コマンドのエイリアス (msim\_setup.tcl 内で定義) を実行しています。

```
1 # Set the hierarchy variables used in Qsys-generated msim_setup.tcl
2 set TOP_LEVEL_NAME "top"
3 set SYSTEM_INSTANCE_NAME "tb"
4 set QSYS_SIMDIR "../HPS_SDRAM_Simulation/testbench"
5
6 # Source Qsys-generated script and set up alias commands used below
7 source $QSYS_SIMDIR/mentor/msim_setup.tcl
8
9 # Compile device library files
10 dev_com
11
12 # Compile the design files in correct order
13 com
14
15 # Compile the additional test files
16 vlog -sv ./test_program.sv -L altera_common_sv_packages
17 vlog -sv ./top.sv
18
19 # Elaborate top level design
20 elab_debug
21
22 # Load the waveform "do file" Tcl script
23 do ./wave.do
24
25 # Log
26 add log -r sim:/top/tb/*
```

## 改版履歴

Revision	年月	概要
1	2014 年 8 月	新規作成

### 免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
2. 本資料は予告なく変更することがあります。
3. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。

株式会社アルティマ ホームページ: <http://www.altima.co.jp> 技術情報サイト EDISON: <https://www.altima.jp/members/index.cfm>

株式会社エルセナ ホームページ: <http://www.elsena.co.jp> 技術情報サイト ETS : <https://www.elsena.co.jp/elspear/members/index.cfm>

4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカー発行の英語版の資料もあわせてご利用ください。