VMM Verification Methodology Manual 活用テクニック

デバイスの記事

赤星博輝

第3回 ランダム生成の機能を使いこなそう

検証ライブラリとその利用ガイドラインである“Verification Methodology Manual for SystemVerilog（VMM）”の活用法を解説する連載の第3回である．今回は，vmm_atomic_genを中心としたランダム生成について解説する．VMMのランダム生成とSystemVerilogの機能を使い，さまざまなテスト・パターンを効率的に作成する．　（編集部）

新しい技術を導入しようとすると，いろいろな問題（壁）が発生し，その壁を乗り越える必要があります．現在の状態と理想とする状態の差が大きければ大きいほど，その壁は高くなります．

● VMM導入の三つの壁

これまでの経験から，VMMの壁は**図1**に示すように三つの要因が組み合わさってできていると感じています．

1）新しい言語であるSystemVerilogに対するギャップ
2）新しい検証ライブラリであるVMM標準ライブラリに対するギャップ
3）新しい考え方であるVerification Methodologyに対するギャップ

SystemVerilogでは，検証に関する機能が大幅に向上しています．また，オブジェクト指向プログラミングが可能になりました．これまでのVerilog HDLやVHDLといったハードウェア記述言語は，オブジェクト指向言語ではありません．このため，ハードウェア設計者はこれまで使ったことがない継承やオーバロード，オーバライドといった機能を使う必要が出てきます．C++やJavaなどに慣れた技術者であれば既に使い慣れた機能です．しかし，ハードウェア系の技術者の場合にはこれまで，C++やJavaを用いてオブジェクト指向でプログラミングする機会が少なく，導入に対する大きなギャップになっています．

次に越えるべきギャップは新しいライブラリです．初めてのものは誰でも戸惑うものです．この点については場数を踏むことで慣れる必要があります．

そして，一番問題になるものは，新しい考え方です．VMMはオブジェクト指向の技術を使い，検証に必要なライブラリを構築し，最小のコーディング量で最大の検証パターンを生成することを目標にしています．この目標を満たすため，VMMにはいろいろなルールが記述されています．ただし，ルールの数が多いため，すべて覚えておくことは難しいと思います．ルールを覚えるよりは，その背景を理解することが重要になります．

この三つの壁はそれなりに高いものだと思いますが，逆に乗り越えてしまえば，検証効率を大幅に改善することが

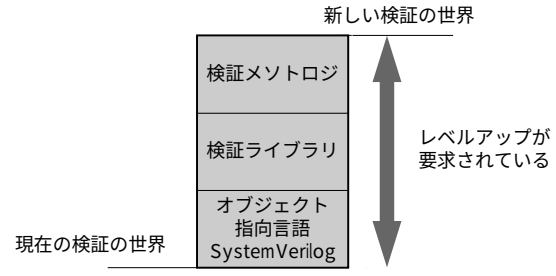

図1　VMMを導入する際の三つの壁
既存のHDLベースのテストベンチ環境からVMMに移行するには三つの壁があるので，一つずつ壁を乗り越えていく必要がある．

KeyWord　SystemVerirog，VMM，ランダム生成，vmm_atomic_gen，vmm_channnel，vmm_xactor，vmm_env，vmm_data，rand，rand_mode，テスト・パターン，constraint_mode

できます．ぜひ，この機会にVMMをマスタしてしまいましょう．

● 少ない記述量でランダム値を生成できる

前回（本誌2006年10月号，pp.139-147を参照）は，**図2**のような2次元データの領域判定を行う回路の検証を行いました．VMMのライブラリを使用し，**図3**のようにランダム生成（vmm_atomic_gen），チャネル（vmm_channel），トランザクタ（vmm_xactor），制御（vmm_env）を行うテストベンチ環境を作成しました．今回は，vmm_atomic_genを中心としたランダム生成について解説します．

ここで前回の復習をしましょう．まず，データ用のクラスを定義するときには，vmm_dataを継承して作成します．あらかじめマクロが用意されており，このデータ型を用いてマクロを呼び出すことでランダム生成を行うクラスを作成できます．少ない記述量でランダム値を生成できることがVMMのランダム生成の特徴といえます．

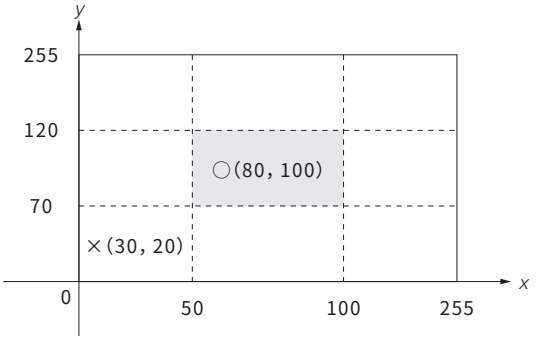

図2　ターゲットの判定回路の動き

256×256の中に（50，70）～（100，120）の長方形があり，その領域内であれば‘1’，領域外であれば‘0’と判定する．

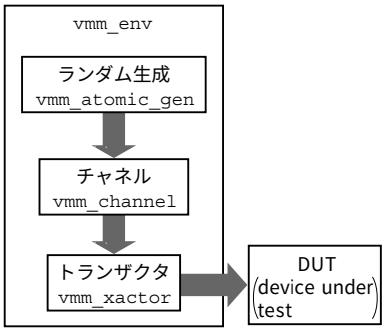

図3　前回の構成と今回の注目点

前回はランダム生成，チャネル，トランザクタ，制御について概要を説明したが，今回はランダム生成について詳細を説明する．

```
class xy_dat extends vmm_data;
//省略
endclass
// データ型定義
'vmm_atomic_gen(xy_data)
// xy_dataのランダム生成クラス定義
```

今回は，VMMのランダム生成とSystemVerilogの機能を使って，さまざまなテスト・パターンを効率的に作成する方法について紹介します．

一部の変数のランダム生成を停止する

図2のような2次元データ（mX，mY）で考えてみます．二つの変数を同時にランダム生成せずに，mXの値は固定にして，mYのみをランダム生成することで，特定の状態についてチェックしたい場合があります（ある特定の縦方向のチェックを行うケース）．このとき，新しいクラスを作って，mYのみランダム変数とすることができます．しかし，個別にクラスを作っていくと，**表1**に示すように組み合わせの数だけクラスができることになります．今回は変数が2個なので四つの組み合わせで済みますが，8変数なら256，16変数なら65536もの組み合わせを使う可能性があります．これでは，最小の記述量で最大のテスト・パターンを発生させるというVMMの目標からは，遠ざかっているといえます．

● randによるランダム生成のON/OFF

SystemVerilogでは，randというアトリビュートを付けた変数に対して，rand_modeというメソッドを使用して，ランダム生成をONしたり，OFFしたりできます．この機能を利用し，状況に応じてランダム生成を制御できます．

2次元のデータ（mX，mY）のmXに対してランダム生成を

表1　アトリビュートの組み合わせ

変数が増加すると，randあり・なしの組み合わせは，指数的に増加してしまう．そのため，VMMではrandを付けることを推奨している．

	mX	mY
Case 1	randなし	randなし
Case 2	randなし	randあり
Case 3	randあり	randなし
Case 4	randあり	randあり