第4回 Verilog 2001ってどんなもの？

大串哲弘

Verilog HDLの標準規格であるIEEE 1364-1995には，文法的なあいまいさが存在した．今年（2001年）3月にIEEEが承認したIEEE 1364-2001（Verilog 2001）では，このあいまいとされてきた部分が改善された．今回はVerilog 2001の概要を，記述例と併せて紹介する．（編集部）

IEEE 1364-1995は，策定されてからすでに5年が経過しました．現在，Verilog HDLは，大規模な半導体設計で必要とされる仕様に沿って，改善が進められています．1999年から始まったVerilog HDLの仕様の改善は，2001年になってようやくIEEEで承認されました．今回は，Verilog HDLの新仕様への各EDAベンダの対応と，Verilog 2001について説明します．

なお，新仕様のトピックスなどは，ドラフト版であるLRM（Language Reference Manual）において確認できたもののみを記載しています．すべてのVerilogソース・コードは，ツールで検査されたものではありません．あらかじめご了承ください．

●Verilog HDLの歴史

簡単にVerilog HDLの歴史を振り返ってみましょう．

1983年，米国Gateway Design Automation社が「Verilog-XL」というディジタル・シミュレータの販売を開始しました．このVerilog-XLはこの後，Verilog HDLのシミュレータの標準として扱われることになります．当時は標準のハードウェア記述言語というものは存在していませんでした．そこでGateway社は，このシミュレータのために「Verilog HDL」というハードウェア記述言語を開発しました．

1989年，Gateway社が米国Cadence Design Systems社に買収されました．このときすでにVerilog-XLシミュレータは，Verilog HDLによって記述するASIC設計のためのもっとも一般的なシミュレータとなっていました．

その後，米国Synopsys社，米国Logic Automation社，米国Logic Modeling Systems社は，Cadence社からVerilog HDLのライセンスを受け，ASIC向けの設計ツールを開発しました．

1990年，Cadence社はVerilog HDLをパブリック・ドメインとして公開しました．このことにより，特別なライセンス許諾を得ることなくVerilog HDLを利用することが可能となりました．

同年，米国Open Verilog Internatinal（OVI）が，Verilog HDLの普及促進のために組織されました．この後，Verilog HDLの管理はCadence社からOVIに移行します．OVIは，最初のパブリック・ドメインのVerilog HDLである「Verilog HDL 1.0」をリリースしました．さらにVerilog HDLをIEEE標準にするため，言語仕様案をIEEEに提出しました．

1995年，IEEEはIEEE 1364-1995を発表しました．このときIEEE 1364-1995では，それまで利用されていた言語仕様の標準規格化のみを目標としており，ハードウェア記述言語としての強化は考慮されませんでした（よって，今回のVerilog 2001に盛り込まれた内容は，5年以上前の言語に対する不足を補ったものになっている）．

2001年，IEEEは新仕様IEEE 1364-2001（Verilog 2001）を承認しました．IEEE 1364 Verilog Standard Group（VSG）が約3年の月日をかけてVerilog HDLの改善仕様を吟味し，策定に至りました．

●Verilog 2000？2001？

Verilog 2001はすでに2000年の段階で完成していました．そして，各EDAベンダもこの新仕様の完成を見て，

ツールへの実装を開始しました．主要なツールのほとんどはVerilog 2001への対応を開始しており，一部の新規仕様はすでに実装されています．「Verilog 2000」，「Verilog 2001」と記載が異なることがありますが，実質的に同じ内容です．IEEEの承認が2001年3月にずれ込んだため，IEEE 1364-2001という名称になったようです．現在は，LRMの最終バージョンが待たれるのみとなっています．

Verilog 2001の主な目標としては，

- 文法的に「あいまい」と指摘されてきた部分の修正
- システム・レベルの記述に対して対応可能な構文の追加

が挙げられています．長い間，従来のVerilog HDLに対する不満が蓄積されたためか，“修正”と呼ぶものが多く見受けられます．

しかし，基本的にIEEE 1364-1995の記述は，そのまま使用可能となっています．IEEE 1364-1995との互換性を損なうことがないようにIEEE 1364-2001の仕様は作成されています．

●新仕様のトピックス

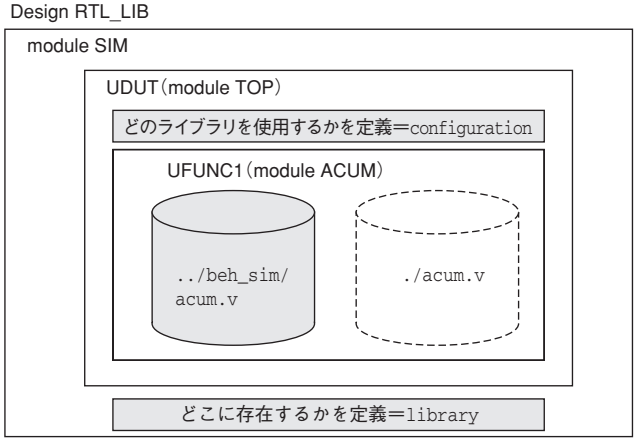
では，具体的に仕様はどのように改善されたのでしょうか．ここでは，新しい仕様について記述例を用いながら説明します．

1）configuration

今までEDAツールのオプション（-v，-yなど）によって行われてきたディレクトリ，ファイル，モジュール，ファンクションなどの物理マッピングを文法で定義したものです．これによって，Verilog HDLの物理的なコードを書き換えなくても，論理的に指定したファイルを加えれば，論理・物理マッピングを行えます．

- インスタンスに接続されるモデルのソース・コードを指定するための構文＝論理マッピングの機能（configuration）
- 任意の場所に存在するソース・コードをまとめて管理する構文＝物理マッピングの機能（library）

で実現します．configuration名の定義とlibraryの定義（マッピング・ファイル）は，それぞれ異なるファイルでも定義できます．例えば，**図1**のインスタンスUFUNC1は，ビヘイビア・レベルのシミュレーション時に，.../beh_sim/acum.vが接続されます．また，RTLのシミュレーション時には，.../acum.vが接続されます．このときの記述例を**リスト1**に示します．



〔図1〕configuration

インスタンスUFUNC1は，ビヘイビア・レベルのシミュレーション時に，.../beh_sim/acum.vが接続されます．また，RTLレベルのシミュレーション時には，.../acum.vが接続される．

2）generate

generateに続くライン・ブロックに条件を加えることにより，generate（生成）の制御を行えるようになりました．制御文を使用して（例えばパラメータ化されたビット幅の指定によって）生成する回路の構造を変えることができます．forループを使用して構文のgenerateが可能になります．また，if else caseを使用して構文のgenerate制御が可能になります．**リスト2**に記述例を示します．

3）定数関数

より複雑な式（function）を用いることによって，定数指定の定義を行えます．定数の指定時に，関数の呼び出しを直接記述できるので，複雑な定数演算も効率良く

〔リスト1〕configurationの記述例

```
//
// configuration名の定義.
config CFGN;
  // このconfigurationを適用するトップ・モジュールの定義
  design RTL_LIB.SIM;
  // デフォルト・ライブラリの検索順序を指定
  default liblist BEH_LIB RTL_LIB;
  // インスタンスに対して適用するライブラリを明示
  instance SIM.UDUT.UFUNC1 liblist BEH_LIB;
endconfig
//
// library の定義.(マッピング・ファイル)
library BEH_LIB "../beh_sim/*.v";
library RTL_LIB "./*.v";
```