

VHDLでのタイプあれこれ

VHDLではsignal, variableなどのオブジェクトにさまざまなタイプを定義できます。標準的に使用されるstd_logicタイプ, integerタイプにはじまり, テストベンチや動作モデルの作成などに使用するrealタイプなどです。さらに, 設計者が独自にタイプを定義して使用することもできます。また, VHDLはタイプの区別が非常に厳密です。異なるタイプをもったオブジェクト同士の接続は禁止されますし, 算術演算などの関数も引き数のタイプを厳密に区別して定義されています。

竹本 悟

今回は, このタイプを取りあげ, タイプにまつわるトピックを中心に解説していきます。

1. デジタル信号の値とEDA ツール

◆デジタル信号の値は'0'と'1'だけではない

コンピュータ・シミュレーションにおいて, デジタル信号はどのように表現されるかご存じですか? デジタル信号は, 論理的に'0'と'1'の2値しかとりません。したがって, シミュレーションを行うときも信号のとり得る値として'0'と'1'の2値のみが用意されていれば大丈夫でしょうか?

実際のデジタル回路では, ハイ・インピーダンス状態をはじめ, さまざまな状態が現れます。たとえば, 3ステート・バッファを考えたとき, '1'と'0'の2値のみでは, ハイ・インピーダンス状態を表現できません。また, 規定どおりのセットアップ時間/ホールド時間を満たさない信号がフリップフロップに入力された場合, 出力信号はどんな値をとるかわかりません。すなわち出力値は不定になるはずですが, '0'と'1'の2値のみでは不定状態を表せません。つまり, デジタル信号のとりうる値として'0'と'1'の二つだけでは, デジタル回路で実際に起こる現象をうまく表せない, ということです。

VHDLに用意されているデータ・タイプとしてbitとbit_vectorがありますが, これらのタイプがとり得る値は'0'と'1'のみです。つまり, 3ステート・バッファのハイ・インピーダンス状態, あるいは出力が不定になっているフリップフロップを表すことはできません(図1)。

◆ツールのデータ・セットは, メーカーごとにバラバラ

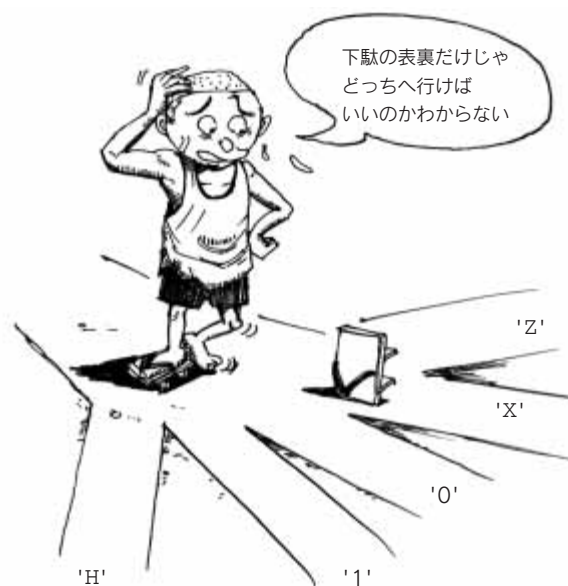
それでは, '0'と'1'以外の値も用意するのでしょうか。しかし, ハイ・インピーダンスと不定だけでよいのでしょうか? 実は, ツールがとる値のセットは標準化されていたわけではなく, ツール・メーカーごとにまちまちでした。たとえば, あるメーカーのシミュレータでは7通りの値をもつが, 別のシミュレータでは16通りの値をもっている, といった具合でした。これは回路図エディタの回路ライブラリも同様です。回路図

エディタのメーカーごとに回路ライブラリの中身(回路図シンボル名やパラメータなど)が異なるのです。たとえば, 2入力のANDシンボルに対して, A社では“A-AND2”というシンボル名を付けているのに, B社では“B-AND2”というシンボル名を付けていたりします。同様に遅延情報などのパラメータの名前や種類もメーカーごとにバラバラ, という状況です。

結局, ASICベンダは各メーカーの回路図エディタに合わせたライブラリを提供するはめになりました。これはASICベンダにとっては頭痛の種でした。各メーカーのツールごとにライブラリを作成するため, 非常に多くの労力を要したのです(図2)。

◆VHDLの信号の取り扱いは世界共通

VHDLシミュレータが登場したころは, 多くのEDAベンダは, それまでの自社の設計ツールと互換性をもたせるため



〔図1〕デジタル回路がとる値

デジタル回路といえども'0'と'1'だけでは回路の動作を完全には表現できない。



〔図2〕EDA ツールのライブラリ

メーカーごとにライブラリのフォーマットが異なったため、ASIC ベンダは対応に追われた。

に、それぞれ独自の方法でデジタル信号を表現していました。このため、VHDL で記述された設計データの間の互換性が損なわれる結果となったのでした(図3)。しかし、これらの不都合を解決するため、VHDL のデジタル信号の表現方法も IEEE という米国の規格化団体によって標準化されました。これは、IEEE Std 1164 と呼ばれています。標準化されたライブラリは、ieee.std_logic_1164 というライブラリ名で利用できます。EDA ベンダも、現在はIEEE が定めた文法にしたがって自社のEDA ツールを開発しています。したがって、デジタル信号の記述方法は一元化され、異なるメーカーのツール間で高い互換性をもつにいたっています(図4)。

2. VHDL の基本タイプと標準タイプ —std_ulogic と std_logic

◆std_ulogic は9種類の値をもつ

std_logic_1164 で定義されるデジタル信号の基本タイプは、std_ulogic です。std_ulogic はリスト1に示すように9種類の値(リテラル)をもちます。

〔リスト1〕std_ulogic タイプの定義

9 個の値(リテラル)をもつ。

```
type std_ulogic is (
  'U', -- 未初期化状態
  'X', -- 強い不定
  '0', -- 強い0
  '1', -- 強い1
  'Z', -- ハイ・インピーダンス
  'W', -- 弱い不定
  'L', -- 弱い0
  'H', -- 弱い1
  '-' -- ドント・ケア
);
```

'U' は、初期化されていないことを表します。これは入力信号やフリップフロップの値がシミュレーション開始時に初期化されていない状態であることを表しています。もちろん実際の回路において'U'という状態があるわけではなく、なにがしかの値を取るわけです。どちらになるか予想できない状態を示しています。

'X' は、その名のとおり不定です。不定といってもそれに対応する回路の状態はいろいろあります。一般には正常でない回路動作を示したときに出力されるように使われます。たとえばフリップフロップのゲートレベル・モデルにおいて、セットアップ時間が足りない場合に出力を不定にするのに使います。実際の回路では'U'と同じようになんらかの値をとるのでしょうか、'X'は、異常信号によって誤った信号を出力する可能性を示すのに使われます。

'0'と'1'は、その値が示すとおり、それぞれ信号が'0'、'1'であることを表します。'Z'はハイ・インピーダンスを示します。

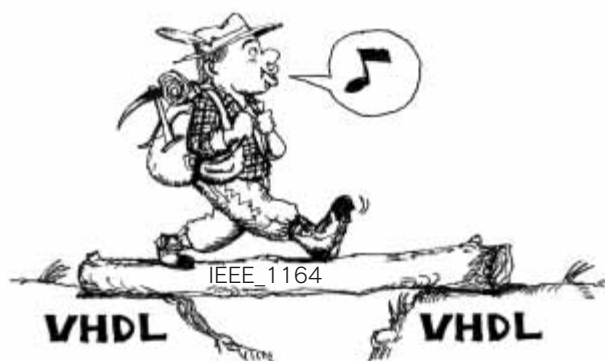
'W'、'L'、'H'は弱い(weak)信号を表します。“弱い”とは、パッシブ・プルアップ/プルダウンされたネットのように、ほかのドライバによってその値が強制変更されるような状況を示します(図5)。「-」はドント・ケアを表します。

3 ステート・バッファを扱わなければ、論理合成を視野に



〔図3〕VHDL シミュレータが登場しはじめたころ

おなじVHDL でも記述作法が違ってかみ合わなかったことも。



〔図4〕現在のVHDL シミュレータの状況

1164 パッケージがあって高い互換性が確保されている。