

解説



ハードウェア記述言語

2. 主要なハードウェア記述言語の特徴と標準化状況

2.2 VHDL†

今井 正 治†

1. ま え が き

VHDL (VHSIC Hardware Description Language) は、IEEE の標準ハードウェア記述言語である*。VHDL の当初の開発目的は、米国防総省 (DoD) に納入される電子システムの仕様記述言語を確立することであった。VHDL の開発作業は 1981 年 6 月に米国防総省によって開始された。1983 年初頭には、言語に対する要求仕様提案書が作成され、言語設計チームに作業が引き継がれた。さまざまな組織の意見を求めるため、VHDL の言語仕様は設計段階の初期から公開された。言語設計チームは、1987 年 2 月に作業を完了し、VHDL Version 7.2 の仕様がドキュメントとしてまとめられた。

一方、VHDL は IEEE の標準ハードウェア記述言語として採用が検討されることになり、1986 年 5 月には DASS (Design Automation Standardization Subcommittee) のもとで VASG (VHDL Analysis and Standardization Group) が標準化作業を開始した。VASG は、現在用いられている VHDL の言語仕様を IEEE に 1987 年 12 月に提案した。IEEE はこれを IEEE Std-1076-1987 (以下では VHDL '87 と呼ぶ) として採用し、1988 年 3 月に言語リファレンスマニュアルを発表した¹⁾。

IEEE の標準化に関する規約では、すべての標準は少なくとも 5 年ごと以内に仕様の見直しを行い、その時々技術水準を仕様で反映させることになっている。現在、VHDL の言語仕様の再標準化 (re-standardization) の作業が行われており、近新しい言語仕様 IEEE Std-1076-1992 (以下で

は VHDL '92 と呼ぶ) が公表される予定である。

VHDL は急速に教育界および産業界に受け入れられつつあり、Dataquest 社によれば、1995 年には最大のユーザ数をもつハードウェア記述言語になると予想されている²⁾。また、教科書や参考書も多数執筆され、北米およびヨーロッパを中心に多くの教育機関での計算機アーキテクチャや VLSI 設計の教育に採用されている^{3)~10)}。また、VHDL に関連した学位論文も多数執筆されている¹¹⁾。

本稿では、VHDL の基本思想、言語の特徴、言語処理系の開発状況、標準化の状況、応用例などについて述べる。

2. 言語設計の基本思想

前章でも述べたように、VHDL 開発の当初の目的は、電子システムの仕様を記述するための言語の仕様を決めることであった。したがって、言語開発の初期の段階では主として仕様記述言語としての側面から言語仕様が検討された。その後の設計自動化技術の発達にともない、シミュレーションと論理合成についても研究が進められ、多数のシミュレータと論理合成系が開発された。また、形式的検証 (formal verification) や動作記述からの論理合成についても研究が行われつつある。

現在の VHDL (VHDL '87 および VHDL '92) の記述対象は、非同期回路を含む一般のデジタル・システムである。他のハードウェア記述言語の中には、同期式のデジタル回路のみを設計対象とするものもあるが、VHDL では非同期回路の記述が基礎になっている。

VHDL では、システム全体の動作と構造から論理ゲートレベルの動作と構造まで、広い範囲にわたって記述できる。しかも、集積回路の製造技術に関しても、CMOS のみならず、nMOS、バイポーラ、などの技術にも対応できるような枠組み

† VHDL by Masaharu IMAI (Toyohashi University of Technology).

† 豊橋技術科学大学情報工学系

* VHSIC (Very High Speed IC) は、米国防総省 (DoD: Department of Defence) が 1970 年代に開始したプロジェクトの名称である。

が用意されている。

また、複数の同期または非同期プロセスを含むシステムが記述できる。さらに、逐次的アルゴリズムを記述するための構文と並列アルゴリズムを記述するための構文が用意されている。

現在の VHDL では、アナログ回路の記述はできない。ただし、アナログ回路の記述を可能にするための言語構文の拡張の検討作業は行われており、標準化作業が完了すれば言語仕様の中に取り入れられることになっている。

3. 言語の主な特徴^{9)~10)}

言語の特徴を、技術的あるいは理論的な面と、言語の開発および保守の体制の二つの面から述べる。

まず、言語開発体制の特徴としては、VHDL は IEEE の標準ハードウェア記述言語であり、IEEE の責任のもとで言語の開発および保守が行われている点があげられる。すなわち、言語の標準化作業および保守作業は IEEE の標準化に関する規則に基づいて多数のボランティアによって公開で行われている。VHDL の標準化活動の詳細は、5. で述べる。

技術的な面では、VHDL は以下のような特徴をもっている。

(1) 複数の設計手法のサポート

大規模なデジタル・システムの仕様記述、設計、シミュレーションなどを可能にするために、さまざまな設計手法を用いることができる。すなわち、

- 階層設計 (トップダウン設計, ボトムアップ設計)

- 同期設計および非同期設計

などを支援する機能が備えられている。

(2) 広い記述範囲

システムレベルから論理回路レベル (PLA やランダム・ロジック) までが同一の言語で記述できる。また、動作 (behavior) の記述も構造 (structure) の記述も可能である。

(3) 記述がプロセス技術に依存しない

利用者は、特定の製造プロセスや IC ベンダに依存することなく、システムの記述が可能である。

(4) 記述が言語処理系に依存しない

VHDL の規格を厳密に守って作られた処理系

の間では、設計の記述のポータビリティが保証される。

(5) 複数の設計グループによる共同開発のサポート

複数の設計グループによるシステムの共同開発を可能とするために、次の機能をサポートしている。

- ライブラリの構築と利用
- 階層設計

(6) 豊富なデータ型と演算子

VHDL は、プログラミング言語 Ada で用いられているデータ型および演算子に基礎を置いている。また、演算子のオーバーロードが可能である*。

表-1 VHDL で使用できるデータ型

分類	対応するデータ型
スカラ型	列挙型, 整数型, 浮動小数点型, 物理型
複合型	配列型, レコード型
その他	アクセス型 (ポインタ型), ファイル型

表-2 VHDL で使用できる演算子

分類	記号	機能
算術演算子 (2項演算子)	+	加算
	-	減算
	*	乗算
	/	除算
	mod	モジュロ*
	rem	剰余**
	**	べき乗
算術演算子 (単項演算子)	+	正
	-	負
	abs	絶対値
関係演算子	=	等しい
	/=	等しくない
	<	より小さい
	>	より大きい
	<=	より小さいか等しい
	>=	より大きい等しい
論理演算子 (2項演算子)	and	論理積
	or	論理和
	nand	論理積の否定
	nor	論理和の否定
	xor	排他的論理和
	xnor	同値***
論理演算子 (単項演算子)	not	否定
接続演算子	&	接続

* mod の結果は、被除数と同じ符号をもち、その絶対値は除数の絶対値より小さい。

** rem の結果は、除数と同じ符号をもち、その絶対値は除数の絶対値より小さい。

*** xnor 演算子は VHDL'92 で導入された。

VHDL で使用可能なデータ型および演算子を表-1 および表-2 にそれぞれ示す。

(7) 信号と変数の分離

信号と変数を区別して記述できる。これによって、複雑な動作モデルの記述が容易になる。

(8) 逐次プロセスと並行プロセスのサポート
逐次プロセスと並行プロセスの両方を同一のモデル内で記述することができる。これにより、複雑なシステムの記述が可能になる。

これらの特徴の幾つかを利用した VHDL の記述例を図-1 に示す。図-1 は、4 状態をもつ同期式有限状態機械 (順序回路) の動作記述である。VHDL による記述は、一般にモデル (model) と呼ばれる。VHDL のモデルは、entity 部と architecture 部に分かれる。一般に、同一の entity に対して、複数の architecture を対応させることができる。システムのシミュレーションや論理合成の際にどの architecture を使用するかは、architecture 部の名前 (この場合「behave」) を参照することによって決まる。したがってまず、ある設計対象の動作レベルのアルゴリズムを記述し検証しておき、次にこの記述を別の architecture として詳細化して論理合成可能な記述とすることができる。この方法を用いると、トップダウン設計が容易に行える。(上記の特徴(1)および(2)の一部に対応する。)

entity 部では、記述されるモデルの入出力ポート (インタフェース) が記述される。この例では、信号 clock および input はそれぞれ 1 および 2 ビットの入力であり、信号 output は 2 ビットの出力であることを表している。bit_vector は VHDL のデータ型の一つであり、この場合 2 ビットの信号を表す。この例には現れていないが、バスのような双方向モードの信号も使用できる。(上記の特徴(6)に対応する。)

次に、architecture 部について解説する。architecture 部では、モデルの動作や構造が記述される。この例では、有限状態機械の動作が記述されている。このモデルは、4 つの内部状態 (ST 0, ST 1, ST 2, ST 3) をもち、初期状態は ST 0 である。一般に有限状態機械の内部状態の表現方法はモデルの入出力と直接関係付ける

必要はないので、内部状態は信号 (signal) ではなく変数 (variable) にしてある。状態割当は、論理合成の際に行えばよいからである。(上記の特徴(7)に対応する。)

さて、この有限状態機械の動作は、process 文を用いて記述されている。process 文は、直後の () 内に書かれた信号 (この場合 clock) の値が変化すると起動される。process 文の内部は逐次的に実行される。(上記の特徴(8)に対応する。)

case 文では、式 (この場合 state) を評価し、その値によって多方向の分岐を行う。この場合、state の値は、ST 0, ST 1, ST 2, ST 3 のどれか一つになる。したがって、state の値と一致する定数の後に書かれた「=>」の後の文に実行が移る。=> の後の if 文では、入力された信号の値によ

```
-- authors : M. Imai, S. Matsushita, R. Sakurai
-- (Toyohashi University of Technology)
-- history : Sep. 27, 1992 - initial version
entity FSM is
  port (clock : in bit;
        input : in bit_vector (1 downto 0);
        output : out bit_vector (1 downto 0));
end FSM;
architecture behave of FSM is
  -- There are four internal states ST0, ST1, ST2, and ST3.
  type states is (ST0, ST1, ST2, ST3);
begin
  process (clock)
    variable state : states := ST0; -- initial state is ST0.
  begin
    if (clock = '1') then -- rising edge.
      case state is
        when ST0 => if input = "01" then
                      output (<= "00");
                      state := ST1;
                    elsif input = "10" then
                      output (<= "01");
                      state := ST2;
                    end if;
        when ST1 => if input = "01" then
                      output (<= "01");
                      state := ST2;
                    end if;
        when ST2 => if input = "11" then
                      output (<= "10");
                      state := ST3;
                    end if;
        when ST3 => if input = "10" then
                      output (<= "11");
                      state := ST0;
                    end if;
      end case;
    end if;
  end process;
end behave;
```

図-1 VHDL による有限状態機械 (順序回路) の動作記述例

★演算子のオーバロードとは、その演算子の本来の定義域および値域を拡張して定義しなおすことである。この定義域および値域はユーザが決めてよい。この機能は VHDL の特徴の一つである。

て出力と次の状態の決定が行われる。信号と変数に対する代入は記法が区別されている。「<=」は信号に対する代入を表し、「:=」は変数に対する代入を表す。

4. 処理系の開発状況

VHDL の言語仕様書が公表された 1987 年以降、多数の VHDL の処理系が開発されてきた。また、商用の処理系も多数販売されている¹¹⁾。VHDL の言語仕様はかなり大きいので、VHDL のフルセットをサポートする処理系は比較的少なく、多くの処理系はサブセットをサポートしている。

VHDL の処理系は、次の三つの副処理系に分けられている場合が多い。

(1) アナライザ

アナライザは、VHDL のソースコードの構文チェックを行い、シミュレータおよびシンセサイザの入力となる中間コードを生成する。

(2) シミュレータ

シミュレータは、与えられたテストベクタを用いて VHDL で記述されたモデルの論理シミュレーションを行う。テストベクタは、モデル記述の内部で信号に一連の値を割り当てることによって与えてもよいし、モデルの外部からシミュレータに与えてもよい。現在実用化されている処理系の多くはイベント駆動型であり、タイムホイールを用いてシミュレーションを実行している。

(3) シンセサイザ

シンセサイザは、記述されたモデルから論理回路を合成する。他のハードウェア記述言語の処理系の場合と同様、モデルの記述には論理合成可能な記述方法と論理合成が不可能な記述方法の 2 通りがある。後者は主として、シミュレーションのためだけの動作記述を行う方法である。ただし、動作記述からの論理合成の方法に関する研究も進みつつあるので、近い将来論理合成可能な範囲が広がると期待される。

5. 標準化の状況

5.1 標準化作業の概要

VHDL '87 の標準化作業は主として北米で進められたが、今回の再標準化作業では、ヨーロッパおよびア

ジア太平洋地域にも VASG の支部が設置され、先進工業国をすべて網羅して議論が行われている。VASG の北米およびヨーロッパ支部は 1990 年 4 月に再標準化の作業を開始した¹²⁾。

日本では、1991 年 4 月に電子機械工業会 (EIAJ) の中に VHDL 小委員会 (主査: 平川和之) が設置され、VASG のアジア太平洋支部として標準化作業を行っている。VASG の組織図を図-2 に示す。

現在行われている VHDL の再標準化では、次の 5 つの作業が一部重なりながら並行して進められている。

(1) 要求分析 (Requirement Analysis)

ISAC (Issue Study and Analysis Committee) に提出された改善提案書 (Issue Report) の分析を行い、言語改善要求書 (Requirement Document) を作成する。

(2) 言語設計 (Language Design)

言語改善要求書にもとづいて、言語変更仕様書 (Language Change Specification) を作成し、新しい言語の文法および解釈 (意味の定義) を与える。

(3) 言語仕様の文書化 (Documentation)

言語変更仕様書にもとづいて、新しい言語のリファレンス・マニュアル (Language Reference Manual) を作成する。

(4) 言語仕様の検証 (Validation)

新しい言語仕様の矛盾点や書きやすさなどを検証するために、理論的な検討を行うとともに、新しい言語仕様で多数のベンチマーク・モデルを記述し検証を行う。

(5) 投票 (Balloting)

以前の規格 (Std-1076-1987) からの変更点を明記した、注釈付きのリファレンス・マニュアル

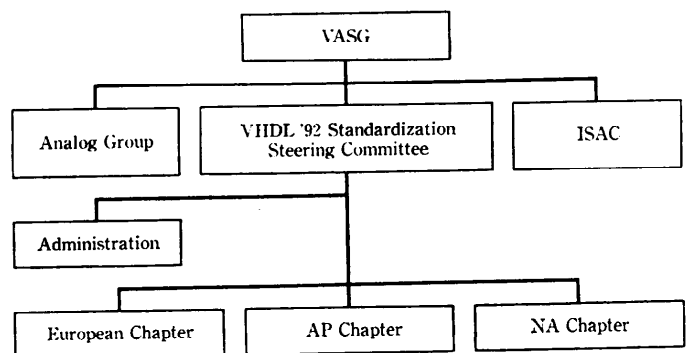


図-2 VHDL 標準化作業委員会の組織構成

(Annotated Language Reference Manual) を有権者に配布し、新しい言語仕様に対する承認の投票を依頼する。

1992年9月現在、上記の(1)~(4)の作業はほとんど完了し、投票の準備作業が進行している。ちなみに有権者としては、IEEEのComputer Societyの会員を中心に、ハードウェア記述言語の専門家、設計技術者など全世界で約400名が登録されている。このうち、アジア太平洋地域の有権者は約70名である。

言語改善要求書には、281項目の改善要求がある。これらの改善要求の中には、比較的容易に実現できる項目と長期にわたる検討が必要な項目がある。前者の例としては、演算子の追加や文法の一部の修正などがある。今回の改訂作業では、前者のみが対象にされ、これまでの言語仕様との互換性を重視して新しい規格が決定された。また後者の項目は、今後の課題として残されている。後者の例としては、アナログ回路の記述、モデルの検証方法、論理合成可能性、などに関する要求がある。

5.2 今後の予定

今後予定されている標準化作業は次のとおりである。

(1) アナログ回路の記述

VHDLはもともとデジタルシステムの記述を主な目的として開発されてきたので、アナログ回路の記述を行うためには、周波数領域での特性の解析やノイズの評価など、アナログ回路固有の問題を記述する必要がある。そのためには、言語の文法の変更、構文の追加、意味の定義などが必要である。

アナログ回路の記述に関する標準化作業は、現在言語の再標準化作業と並行して進行中であり、1993年中をめどに新しい言語仕様の提案が行われる予定である。

(2) モデルの検証、論理合成可能性

モデルの検証(formal verification)、論理合成可能性(synthesizability)などの要求に関しては、理論面を含め、長期にわたる検討作業が必要である。これらの課題に関しては、現在専門のワーキング・グループが検討中である。

(3) その他

現在、論理シミュレーションで用いられる論理

値の規格(Std-1164)がIEEEで標準化されつつある。この規格は9種類の論理値を用いるため、通称MVL-9と呼ばれているが、VASGではこの規格をVHDLでの標準として取り入れるべく検討している。そのほかにも、数値関数パッケージの標準化など関連する作業も積極的に行われている。VASGでの、VHDLの標準化に関連した他の標準化活動および研究活動を表-3に示す。

6. 応用例

米国では、国防総省の意向もあって、すでに多数のシステムがVHDLで記述され実現されている。日本国内では、VHDLによるシステムの開発例はまだ数件しか報告されていないが、今後の処理系の充実にもなって事例が増えると期待される^{13)~15)}。

図-3に、VHDLの記述例を示す。これは、VHDL '92(Std-1076-1992)にしたがって n ビットのリップルキャリアダーを記述した例である。この例では、generate文を用いて、可変個(N)のコンポーネント(フルアダー)をもつモジュールをインスタンスエートする。VHDL '92では、generate文に宣言部が追加され、コンポーネントとして使用されるarchitectureの指定が容易になった。

7. あとがき

本稿では、VHDLの基本思想、主な特徴、処理系の開発状況、標準化の動向、システムの記述例などについて述べた。

表-3 DASSで行われている標準化および研究活動
(上の半分はVHDLと直接関係のある活動)

グループ名	種類	標準規格
VASG	ASG	PAR 1076
VHDL Analog Extentions	WG	PAR 1076.1
VHDL Intermediate Form	ASG	PAR 1163
VHDL Modeling Practice	WG	PAR 1164 (MVL-9)
VHDL-EDIF Interoperability	WG	PAR 1165
VHDL Synthesis Package	SG	
VHDL Math Package	SG	
Information Modeling	WG	PAR 1078
System Design Languages	WG	
Timing and Backannotation	SG	
Test Standards		SCC 20

種類 ASG: Analysis and Standardization Group
SG: Study Group WG: Working Group

```
-- N-bit ripple carry adder model description using
-- a new generate statement feature of VHDL '92.
-- The architecture of FA (full adder) will be given
-- elsewhere, e.g. in a library.
-- authors: M. Imai, T. Nakata, S. Matsushita
-- (Toyohashi University of Technology)
-- history: May 4, 1992 - initial version
```

```
entity N_bit_adder is
  generic (N : natural);
  port (A, B : in bit_vector (N-1 downto 0);
        Cin : in bit;
        Sum : out bit_vector (N-1 downto 0);
        Cout : out bit);
end N_bit_adder;

architecture logic of N_bit_adder is
  -- I/O port description of one bit full adder
  component FA
    port (X, Y, Cin : in bit;
          Sum, Cout : out bit);
  end component;
  signal C : bit_vector (N-1 downto 0);
begin
  Lowest bit: -- Cin should be given from outside
  for i in 0 to 0 generate
    for unit_1 : FA use entity FA (logic);
  begin
    unit_1 : FA
      port map (A(i), B(i), Cin, Sum(i), C(i));
  end generate;
  Otherbit: -- the carry should be propagated
  for i in 1 to N-1 generate
    for unit_n : FA use entity FA (logic);
  begin
    unit_n : FA
      port map (A(i), B(i), C(i-1), Sum(i), C(i));
  end generate;
  Cout <= C(N-1);
end logic;
```

図-3 リップルキャリアダーの VHDL による記述例

VHDL は、汎用性を目指したハードウェア記述言語であり、言語仕様も複雑である。そのため、初心者の学習には長い時間が必要である。処理系が複雑になり処理効率が悪い、などの批判も受けている。

しかし、全世界的な規模で言語の標準化活動が行われているので、有効な教育方法と設計手法が近い将来確立されると期待できる。また、ワークステーションの処理性能は急激に向上しつつあるので、近い将来には、多少複雑な処理系であっても十分に実用になり得ると思われる。もちろん、処理系の改良による処理効率の向上も期待できる。したがって、VHDL は今後急速に普及してゆくとと思われる。

謝辞 本稿で示した VHDL の記述例を作っていたいただいた豊橋技術科学大学情報工学専攻の松下昭司君、中田武治君、同情報工学課程の桜井涼二君に深謝する。

参考文献

- 1) IEEE: *IEEE Standard VHDL Language Reference Manual*, IEEE, 1988. (邦訳: VHDL ハードウェア記述言語, 日本規格協会, 1991).
- 2) Dataquest: *Dataquest Perspective*, CCAM-EDA-DP-9201, April 27, 1992.
- 3) Armstrong, J. R.: *Chip-Level Modeling with VHDL*, Prentice-Hall, 1989. (邦訳: 白石, 山本訳, VHDL デザイン・テクニック, 電波新聞社, 1990).
- 4) Bhasker, J.: *A VHDL Primer*, AT&T (1992).
- 5) Coelho, D. R.: *The VHDL Handbook*, Kluwer Academic Publishers, 1989. (邦訳: 片桐訳, VHDL ハンドブック, 大倉商事, 1990).
- 6) Dewey, A.: *Analysis and Design of Digital Systems with VHDL*, Addison Wesley (1992).
- 7) Leung, S. S. and Shanblatt, M.: *ASIC System Design with VHDL: A Paradigm*, Kluwer Academic Publishers (1989).
- 8) Lipsett, R., Schaefer, C. and Ussery, C.: *VHDL: Hardware Description and Design*, Kluwer Academic Publishers, 1989. (邦訳: 杉山監約, VHDL: 言語記述によるハードウェア設計へのアプローチ, マグロウヒル, 1990).
- 9) Perry, D.: *VHDL*, McGraw-Hill (1991).
- 10) Schoen, J. M.: *Performance and Fault Modeling with VHDL*, the MITRE Corp. (1992).
- 11) *VHDL Resource Directory*, VHDL International (1992).
- 12) Shahdad, M.: 1992 VHDL Standardization Overview, Proc. of EuroDAC '92, pp. 666-667 (1992).
- 13) Borgstrom, T. H., Masuda, T. et al.: VHDL Design in Japan: A Real-Time Digital Video Processor, presented at EDA Standards Forum Japan '92 (Apr. 1992).
- 14) 町田, 吉田他: ハードウェア記述言語 (VHDL) を用いた ASIC 開発, 1992年電子情報通信学会春期全国大会講演論文集, A-103 (1992).
- 15) Shinde, H.: VLSI Developments using VHDL and Top-Down Design Methodology, Proc. of Fall 1992 VHDL Int'l User's Forum, to appear (1992).

(平成4年9月21日受付)



今井 正治 (正会員)

1974年名古屋大学工学部電気工学科卒業。1979年同大学院博士課程(情報工学専攻)修了。工学博士。

1979年より豊橋技術科学大学情報工学系勤務。現在、同助教授。1984年より85年にかけて米国サウスカロライナ大学に文部省在外研究員(客員助教授)として滞在し教鞭をとる。1991年より、日本電子機械工業会(EIAJ) VHDL 小委員会仕様ワーキンググループ主査としてVHDL '92の標準化活動に従事。これまで、組合せ最適化問題、計算機アーキテクチャ、VLSIの設計自動化などの研究に従事。IEEE, ACM, 電子情報通信学会, 人工知能学会各会員。