

3

HDL

半導体製造技術の進歩により LSI の集積度が、飛躍的に向上しているため、システム LSI の設計には、大規模な回路を短期間で設計し、さらに設計資産の再利用による設計時間の短縮が要求されています。

そこで、従来の CAD(Computer Aided Design) によるゲート・レベル設計よりも、上位の言語レベルで記述するハードウェア記述言語 (HDL: Hardware Description Language) による設計が必要になってきました。

HDL は、プログラミング言語の構文に似ていて、入力信号の変化によって平行に動作する並行処理やタイミング時間の指定が可能です。また、HDL を用いることにより、仕様の動作記述やシステムの抽象的なレベルの検証から、論理回路、ネットリストまでの幅広い範囲の設計が可能となります。

代表的な HDL には、VHDL と Verilog-HDL があります。

VHDL

VHDL(VHSIC Hardware Description Language) は、米国国防総省の VHSIC(Very High Speed Integrated Circuit) 委員会で 1981 年に提案されました。当時、国防総省向けの ASIC 開発は長いもので 3 年から 4 年もかかっていたため、開発当初の時点では一番スピードが速い ASIC を使用していても、半導体プロセスの進歩によって、開発が完了する時点では時代遅れになってしまうという問題が発生していました。そこで、直接ロジック・ゲートを回路図(スキーマティック)で入力するのではなく、HDL で記述することによって、開発終了時に一番スピードの速い ASIC を選択して論理合成できるようになりました。

現在では、国防総省が調達するすべての ASIC は、電子システムの仕様記述言語として VHDL の記述付きで納入されるように義務づけられています。

VHDL は、プログラミング言語 ada を基に開発された言語であり、デジタル・システムのドキュメント記述から、設計、シミュレーションまでの幅広い範囲を記述することが可能です。VHDL は、その後 IEEE の標準ハードウェア記述言語として採用され、IEEE-1076 仕様が 1987 年に VHDL'87、さらに 1993 年に VHDL'93 として標準化されました。

Verilog-HDL

Verilog-HDL は、シミュレータ Verilog-XL のためのシミュレーション専用言語として、ゲートウェイ・デザイン・オートメーション社 (現在はケイデンス社に合併) によって開発されました。

Verilog-HDL は、C 言語を基にシミュレーション専用の言語として開発されたため、LSI 設計を重視した言語であり、記述性やシミュレーションの機能が充実しています。Verilog-HDL は、IEEE-1364 仕様として標準化されました。

HDL を用いたハイレベル設計では、HDL による機能検証を行った後に、論理合成ツールを用いてゲート回路を自動生成することにより、同じ HDL の記述から設計制約条件によってコスト・パフォーマンスなどに合った LSI の製造が可能です。

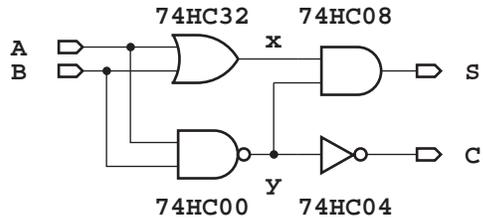
真理値表

半加算器

入力		出力	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

⇒ CADでは、
シンボル記号と
配線で設計する

CAD Half Adder



↓
HDLでは、機能を記述している
遅延時間は、論理合成の設計制約条件とライブラリで決定される
CADによる回路図は、機能と遅延時間を表現している

VHDL

```
library IEEE;
use IEEE.std_logic_1164.all;

entity HA is
  port (
    A, B : in std_logic;
    S, C : out std_logic
  );
end HA;

architecture RTL of HA is
  signal x, y : std_logic;
begin
  x <= A or B;
  y <= A nand B;
  S <= x and y;
  C <= not y;
end RTL;
```

Verilog-HDL

```
module HA (S, C, A, B);
  input A, B;
  output S, C;

  wire x, y;

  assign x = A | B,
         y = ~(A & B),
         S = x & y,
         C = ~y;
endmodule
```

下流工程の記述に優れている
論理合成や論理検証など

上流工程の記述に優れている アルゴリズム設計やアーキテクチャ設計など

論理合成

