

[Verilog HDL 1]

<ハードウェア記述言語>

『ハードウェアの設計』という言葉を知れば、論理素子(AND, OR, NOT など)を用いて、回路図エディタに設計することを想像するかもしれない。ところが近年では、ハードウェアを言語で表現できるハードウェア記述言語(Hardware Description Language, 以下ではHDLと略する)を用いて、ハードウェア及びLSIの設計が行われている。

これはHDLを用いることで、ハードウェア・アーキテクチャの設計からレイアウト設計、テストに至る一連の工程の中で、動作レベルやゲートレベルなどの記述を経たトップダウン設計を行うことが可能となり、設計生産性と設計信頼性が飛躍的に向上するためである。

これまでは図1に示すように回路図エディタに設計者が論理素子を配置していく「手組み設計」と呼ばれる手法が主流であった。しかし、微細化技術の発達により図2のようなHDLを使用した設計の方がより効率的であり、かつ信頼性の高い設計が可能となった。

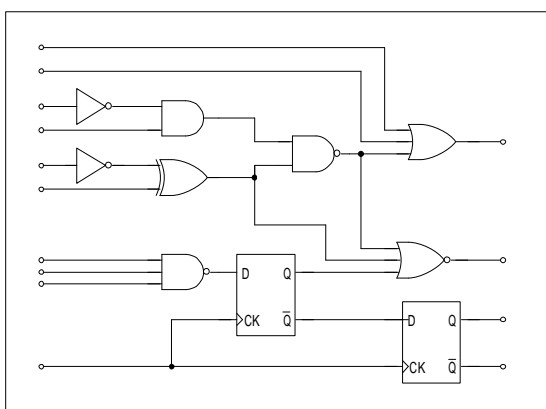


図1: 論理回路図

```

1 module chat_filt(clk, rst, sw, sw_filt);
2   input  clk, rst;
3   input  [3:0] sw;
4   output [3:0] sw_filt;
5   reg    [3:0] sw_filt;
6   reg    [11:0] count0, count1, count2, count3;
7
8   always@(posedge clk or posedge rst)
9     if(rst)
10      begin
11        sw_filt[0] <= 1'd0;
12        count0 <= 12'd0;
13      end
14   else if(!sw[0])
15     begin
16       sw_filt[0] <= 1'b0;
17       count0 <= 12'd0;
18     end
19   else if(count0 == 12'd2400)
20     begin
21       sw_filt[0] <= 1'b1;
22       count0 <= count0;
23     end
24   else
25     count0 <= count0 + 12'd1;
26
27   always@(posedge clk or posedge rst)
28     if(rst)
29       begin
30         sw_filt[1] <= 1'd0;
31         count1 <= 12'd0;
32       end
33   end

```

図2: Verilog-HDL 記述

表1に手組み設計手法とHDLを用いた設計手法での比較をまとめる。

表1: 手組み設計手法とHDLを用いた設計手法の比較

	手組み設計手法	HDLを用いた設計手法
設計時間	小規模のものなら短時間であるが大規模になると入力に時間がかかる。	記述言語を理解するのに時間がかかるが理解すればテキストに入力するだけ。
設計手法	論理式を最初に考える必要がある。	論理式を考える必要はない。
回路変更	規模が大きくなるに連れて困難。	容易
可読性	設計者以外では理解が困難。	言語を理解している人であれば容易
エディタ	特定のメーカーのライブラリを使用する必要がある。	どのメーカーでも作成可能。
回路面積	設計者の技術力に依存。	合成ツールの能力に依存。能力次第では面積の大きい設計になる可能性もある。

<HDL の種類>

1990 年代当時, HDL には VHDL (VHSIC HDL), Verilog-HDL, UDL/I (Unified Design Language for Integrated Circuit), SFL (Structured Function description Language) などが存在した. それぞれの特徴を表 2 に示す. このうち, 現在では VHDL と Verilog-HDL が広く普及し使用されるようになった. 今日ではロジック回路設計の大半がこの二つの HDL によって設計されている.

表 2 : VHDL の種類と特徴

言語名	開発元	特徴
VHDL	アメリカ国防省が中心に開発	幅広い分野の記述が可能であるが言語レベルが高度.
Verilog-HDL	シミュレーターVerilog 言語として開発がスタート	幅広い分野の記述が可能であり言語レベルは容易. VHDL より言語能力は劣る.
SFL	PARTHENEN システムの言語として開発がスタート	同期回路のみ記述できる. 単純で分かりやすい言語.
UDL/I	日本電子工業振興協会が開発	同期回路のみ記述できる. 単純で分かりやすい言語.

<Verilog-HDL について>

Verilog-HDL は C 言語をベースに構成されたハードウェア記述言語である. よって, C 言語に似た記述形態をとっており C 言語経験者にとっては理解しやすい言語である. C 言語と異なる点としては入出力に関する記述機能が追加されていること, 時間の概念が追加されていること, ポインタや構造体等の機能がないことである. 制御文としては if 文や case 文などがサポートされている.

記述に必要な情報は以下に示す 3 つのみである.

- (1) 機能ブロックの名前(モジュール名)
- (2) 入出力のピン名称とその属性
- (3) ブロックに求める機能

これらを分かりやすく図示したものを図 3 に示す.

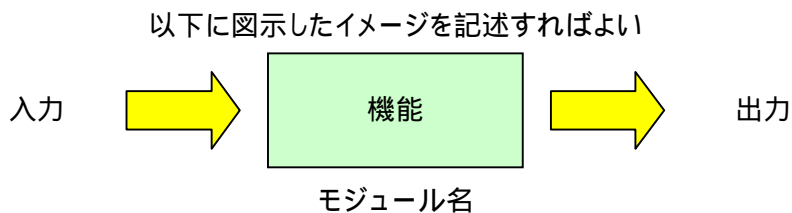


図 3: Verilog-HDL での記述に必要な情報