

# ハードウェア記述言語(HDL)のシステム設計エンジニアリングへの適用

1Q-8

佐野直樹 滝鼻容子 久保典夫  
 横河電機(株) デバイス研究所

## 1. はじめに

近年、プロセス制御システムに代表されるように、制御システムは大規模化かつ複雑化し、そのシステム設計エンジニアリングの効率向上が益々その重要性を増している。

今回、システム設計エンジニアリングに関してハードウェア記述言語(HDL)設計手法<sup>1), 2)</sup>に基づく設計・テスト・デバッグ手法について提案する。

HDLは、元来VLSI設計のためのハードウェア記述言語であるが、電子システム設計のためのシミュレーション言語、モデリング言語、プログラミング言語、論理合成言語でもある。当社では、既にこのHDL設計手法を採用し、各種ASICチップ開発に適用することにより、従来の回路図面を利用したゲートレベルの設計手法と比較してその開発期間の大幅短縮を計ることが出来た。<sup>3), 4)</sup>

我々は、VLSI設計において自動論理合成と相まって開発効率の飛躍的な向上をもたらしたこのHDLを”システム記述言語”と位置づけ、システム設計エンジニアリングへの適用を現在検討している。このHDLによるエンジニアリング手法は、マルチシステムベンダ及びエンドユーザに対して共通なエンジニアリング・データベースとシミュレーション環境を提供し、コンカレントな設計・テスト・デバックを可能にするものである。本手法により、汎用ワークステーション上で制御内部計器とプロセスモデル等を結合した統合シミュレーションを効率よく行うことができる。本手法を簡易プロセスモデルを含む実験システムに適用し、その有用性を確認した。

本稿では、HDLによるエンジニアリング手法の開発の狙い、特徴、HDLによるシステム設計エンジニアリング環境、実験システム適用事例等について述べる。

## 2. HDLによるエンジニアリング手法の開発の狙い

本エンジニアリング手法の開発の狙いは次の通りである。

- ①共通なエンジニアリング・データベースの構築  
 → マルチシステムベンダ及びエンドユーザがエンジニアリング・データベースを共有可能
- ②共通なシミュレーション環境の提供  
 → マルチシステムベンダ及びエンドユーザがシミュレーション環境を共有可能
- ③エンジニアリング生産性の向上  
 → 開発、設計、テスト、デバックの各工程の生産性向上、シミュレーションでの仮想プロトタイプが可能  
 以上の狙いにより、”コンカレントに効率よく設計・テスト・デバックが可能な環境を構築する”ことが本エンジニアリング手法の目的である。

## 3. HDLによるエンジニアリング手法の特徴

本エンジニアリング手法の特徴を以下にまとめる。

- ①HDLでのエンジニアリング・データベース化  
 → HDLによるシステムの機能設計  
 (制御内部計器、シーケンス制御、マンマシンインタフェース、簡易プロセスモデル等)  
 → HDLによるテストプログラム設計
- ②汎用シミュレータによる機能レベルシミュレーション  
 → サブシステム及びシステム全体のシミュレーションによるテスト・デバック  
 → 入力条件や動作条件の変化に対するシステム応答の確認  
 → 物理的に再現できないような環境でのシステム動作の確認/解析
- ③エンジニアリング効率の向上  
 → 共通なエンジニアリング・データベースとシミュレーション環境によるコンカレントエンジニアリング

## 4. HDLによるシステム設計エンジニアリング環境

図1に本エンジニアリング手法を用いたシステム設計エンジニアリング環境を示す。本エンジニアリング環境は、汎用シミュレータ上に構築され、制御システムとプロセスモデルよりなる対象システムとテストプログラムとマンマシンインタフェースよりなるテストベンチとから構成される。

制御システムには、フィードバック制御内部計器モデル、シーケンス制御機能等が含まれ、これらは全てHDLにて記述されている。テストプログラムは、対象システムのテストを行うためのものである。テストすべき機能ブロック毎にサブテストプログラムとして

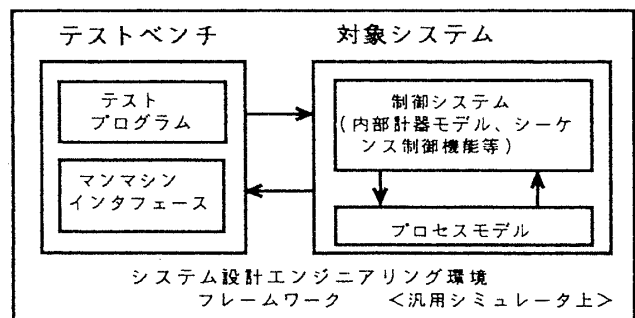


図1 システム設計エンジニアリング環境

Application of HDL to System Design Engineering  
 Naoki Sano, Yoko Takihana, Norio Kubo  
 Devices Laboratory, Yokogawa Electric Corporation

制御システムと同様にHDLにて記述されている。マンマシンインタフェースは、主としてデバック時にインタラクティブなコマンドの実行、システム変数の表示、システムパラメータの修正等に用いられる。

我々は、HDLとしてVerilog-HDL、汎用シミュレータとしてVerilog-XLを現在用いている。

5. 実験システム適用事例

本エンジニアリング手法を実験システムの1つとして、抄紙機制御におけるJ/W(ジェット/ワイヤ速度比)制御システムに適用し、その有用性を確認した。以下、この適用事例について紹介する。

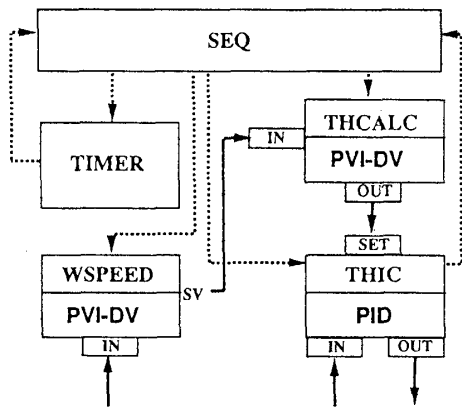


図2 J/W制御システム主要構成図

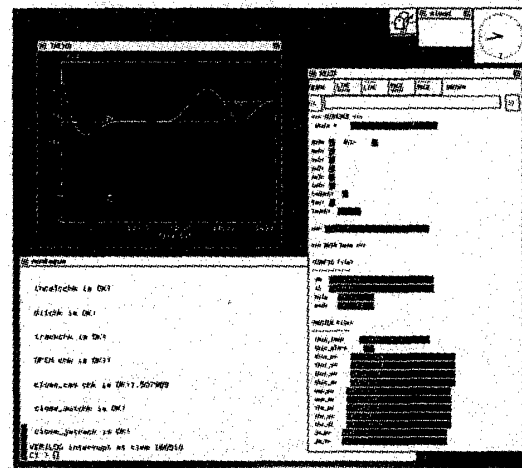


図4 HDLによるシミュレーション環境

```

module fcs(xrst, cp, state,
           di0, di1,
           P_wspeedin, P_thicin, P_thicout);
    timer timercell(xrst, cp,
                   seq_timer,
                   b_timer_STATUS);
    seq seqcell(xrst, state,
               di0, di1,
               b_thic_LOOP,
               b_timer_STATUS,
               seq_SW0,
               seq_SW2,
               seq_timer,
               seq_thic_LOOP);
    wspeed wspeedcell(xrst, state,
                     P_wspeedin,
                     b_wspeed_SV);
    thcalc thcalccell(xrst, state,
                     b_wspeed_SV,
                     b_thcalc_out,
                     seq_SW0,
                     seq_SW2,
                     b_thic_SV,
                     b_thic_PV,
                     thcalc_SW3);
    thic thiccell(xrst, state,
                 P_thicin,
                 b_thcalc_out,
                 seq_thic_LOOP,
                 b_thic_PV,
                 b_thic_SV,
                 b_thic_LOOP);
endmodule

module seq(xrst, state,
           di0, di1, b_thic_LOOP, b_timer_STATUS,
           seq_SW0, seq_SW2, seq_timer, seq_thic_LOOP);
    always @(negedge xrst or state) begin
        if(xrst == 1'b0) begin
            :
        end
        else if(state == `SEQ_work) begin
            if((di0==`Y)&&(timer_st==`TOP)) timer_new= `ON;
            if(di0==`N) timer_new= `OFF;
            if((di1==`Y)&&(b_thic_LOOP==`MAN)) thic_st_new= `AUT;
            if(di1==`N) thic_st_new= `MAN;
            if(timer_st==`HI) sw0_new= `ON;
            :
        end
    end
endmodule

module thic(xrst, state, b_in, b_set, b_out, seq_thic_LOOP,
            b_thic_PV, b_thic_SV, b_thic_LOOP);
    e = pv - sv;
    iel = `CDT / 1 + e;
    del = D / `CDT * (dpv - dpvl);
    dmvl = 100.0 / P * (pel + iel + del);
    mvl = mvl + dmvl; // manipulation value
    :
    dpvl = dpv;
    mvl = mvi;
    :
endmodule
    
```

図3 HDL機能モデル

6. まとめ

HDLによるエンジニアリング手法及びそのプロセス制御システムへ適用事例について紹介した。本エンジニアリング手法はプロセス制御システムに限定されることなく、一般のシステム設計エンジニアリングにおいても有効な手法であると思われる。今後の展開としては、本手法の実システムへの適用検討等が考えられる。

<参考文献>

- 1) E. Sternheim, et al., "Digital Design with Verilog HDL", Automata Publishing Company, 1990
- 2) J. Armstrong, "Chip Level Modeling with VHDL", Englewood Cliffs, NJ: Prentice Hall, 1989
- 3) 佐野、久保: "HDLによるASIC開発手法", 情報処理学会第43回全国大会, 4R-5, (1991-10)
- 4) 佐野、久保: "大規模ASIC設計・デバックサポートシステム: CEEDS-ASIC", 情報処理学会第45回全国大会, 4K-4, (1992-10)