

Verilog-HDL による 8 ch PWM 回路の論理合成

天 谷 純 治・多 田 昭 晴*

岡山理科大学大学院工学研究科修士課程情報工学専攻

*岡山理科大学工学部情報工学科

(1996年10月7日 受理)

1. ま え が き

近年の LSI 技術の進展により、LSI は広範囲に使用され大規模化・複雑化してきている。また、製品サイクルの短縮化傾向にあり、設計期間の短縮化が要求されている。従来、回路図入力をベースに行われてきた特定用途向 IC である ASIC (Application Specific IC) は多品種少量生産が行われるため低コスト設計が必須となり、回路の大規模化に伴う複雑度の増大によって機能検証が限界に達しようとしている。細かで複雑な表現で行われているゲートレベルの回路設計を人間が理解できるためには、抽象度の高い設計手法が必要となっており、回路図入力ベースの設計手法に代わるより抽象度の高いレベルの設計手法として、HDL (Hardware Description Language) を用いた手法が注目されてきている。HDL は LSI 設計を仕様レベルで行うために提唱されている機能記述言語で、従来の回路図入力ベースの設計手法と比較して回路図入力等に必要な時間を短縮でき、設計期間を大幅に短縮することが可能であるとともに、CAD ツールの高性能化によって人手設計を超えた設計スピード、設計品質が得られるようになってきている。^{1,2)}

本研究では、Verilog-HDL を用いてディスプレイ制御用 8 ch PWM (8 channel Pulse Width Modulation) 回路の機能記述を行い、ASIC の一種であるゲートアレイを対象に論理合成し HDL による回路設計について検討する。

2. Verilog-HDL

Verilog-HDL は CADENCE 社が論理シミュレータ Verilog-XL 用の機能記述言語として開発したもので、1995年に IEEE で標準化されている。

Verilog-HDL による回路設計では、下位 (論理ゲートレベル) から上位 (システムレベル) まで広範な機能記述が可能であり、階層化設計を行うことができる。このため回路設計は、半導体プロセス技術やデバイス技術に直接依存せずシステムの設計が可能である。記述は C 言語に類似しており、逐次的または並列的動作を実現することができる。また、シミュレーションの入力パターンやシミュレーションの結果の表示、期待値比較も一つの言語体系の中で記述できるという特徴を持つ。

論理合成ツールでは、回路面積最小や出力最速等の設計目標にあわせて任意の回路を得ることができ、設計効率の向上、回路の大規模化を計ることが可能である。従来の回路図入力を用いた設計では、ゲートレベルの論理設計からテストデータの作成、機能タイミングシミュレーションの順に作業を行うボトムアップ設計であったが、HDLを用いた手法では回路をモジュールという単位に分割し、階層化構造を明確にした上で設計するトップダウン設計を行えるため、システムの最適化、ミスの早期発見による効率化が可能である。³⁾

3. 設計開発フロー

Verilog-HDLによる設計フローを図1に示す。本研究で行った工程は(A)から(E)の部分である。

4. 8ch PWM 回路

4.1 仕様

8ch PWM 回路はパソコン等に使用される17インチディスプレイの制御回路であり、PWMデータおよびアドレスデータを入力し、PWMデータをパルス幅に変調し、輝度、照度等の出力チャンネルの選択を行う。PWMデータが0の場合基準クロック1周期、1の場合2周期出力するように設計する。最終出力はCLKと同期させて出力する。要求出力を図2、入力波形タイミングを図3に、入出力端子機能を表1に示す。

4.2 RTL図作成

仕様に基づいてRTL図を作成する。RTL (Register Transfer Level) とは回路機能をレジスタ間の信号の流れで記述するレベルであり、これによって回路の階層化、信号の明確化を行い、回路機能の明確化を計る。図4に作成した8ch PWM 回路のRTL図を示す。

4.3 各モジュール動作

(1) シリアルーパラレルインターフェース

シフトレジスタ(shf)にデータ部8ビットとチャンネル選択用のアドレス部3ビットからなるシリアルデータ(SD)をシリアルクロック(SC)の立ち上がりで入力する。デコーダ(dec)にはアドレスデータがロード信号(LD)の立ち下がりで読み込まれ、8本の信号線(Gn)のうち一本を選択する。データ部8ビットはDnに出力する。

(2) カウンタ(count)

要求波形より、PWM波形の周期はクロック(CLK)信号256周期なので、8ビットカウンタを用いる。リセット(RST)はローアクティブの非同調リセットである。クロックの立ち上がりでカウント動作を行いQnに出力する。

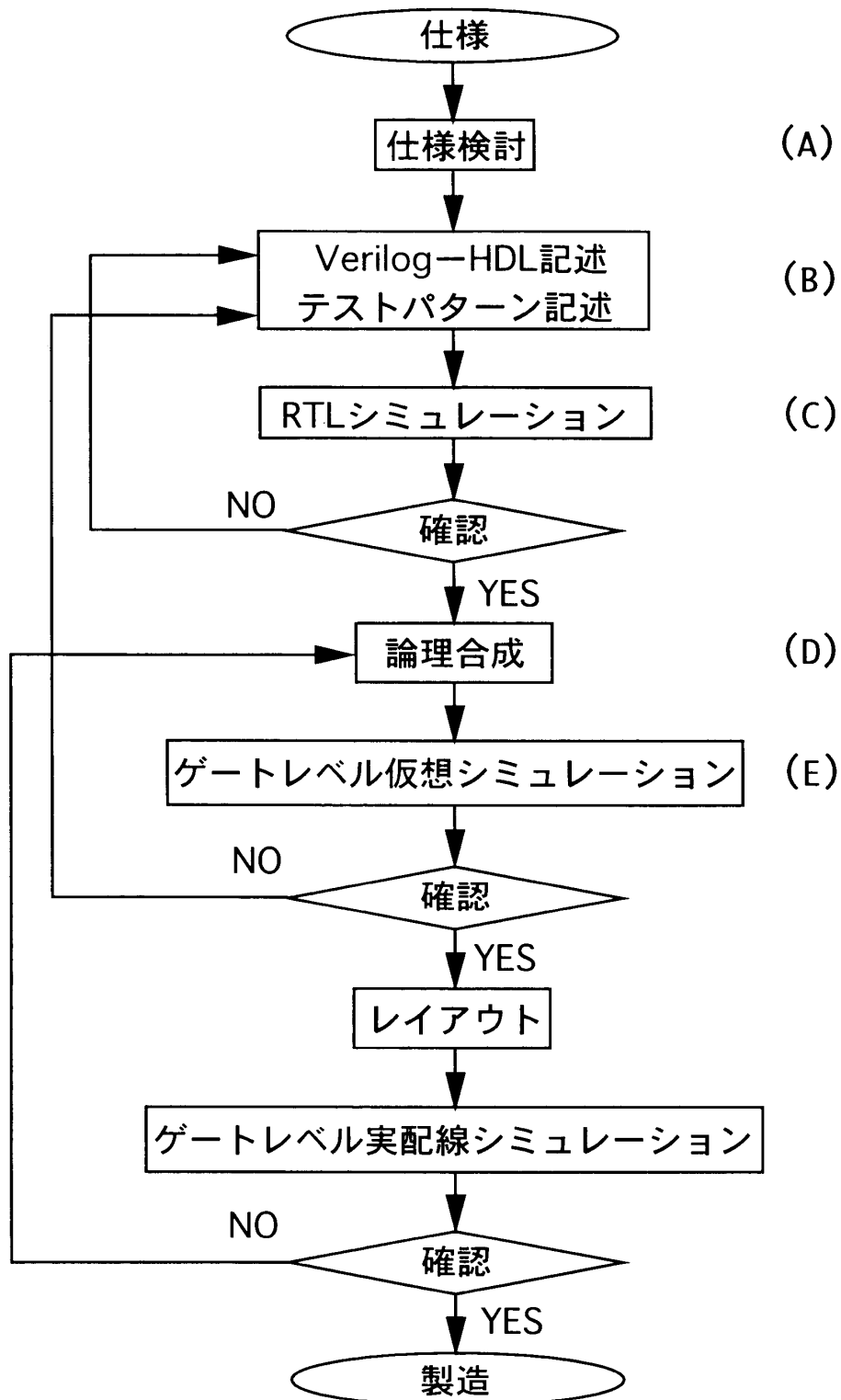


図1 Verilog-HDLによる設計フロー

(3) PWM

spifより入力された8ビットのデータをラッチ (latch) の入力 G_n でレベルラッチし B_n に出力する。コンパレータ (cmp) で、 B_n とカウンタ出力 Q_n を比較し、 $B_n \geq Q_n$ が真である場合出力 S を H にする。コンパレータの出力 S をフリップフロ

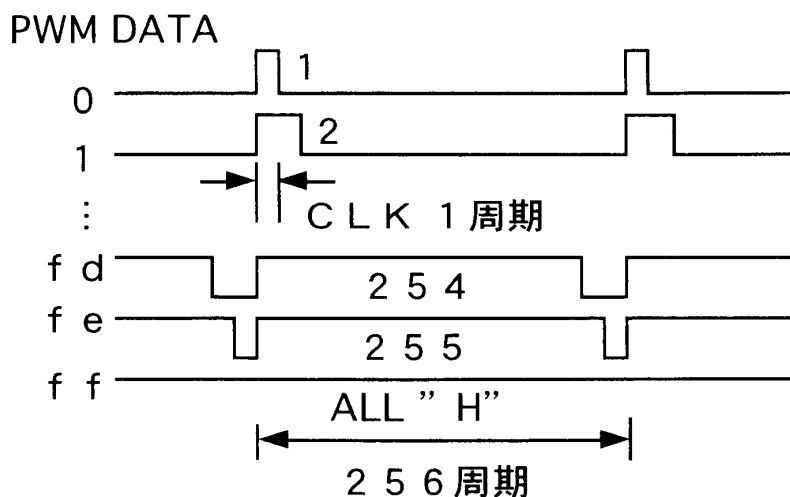


図2 要求出力

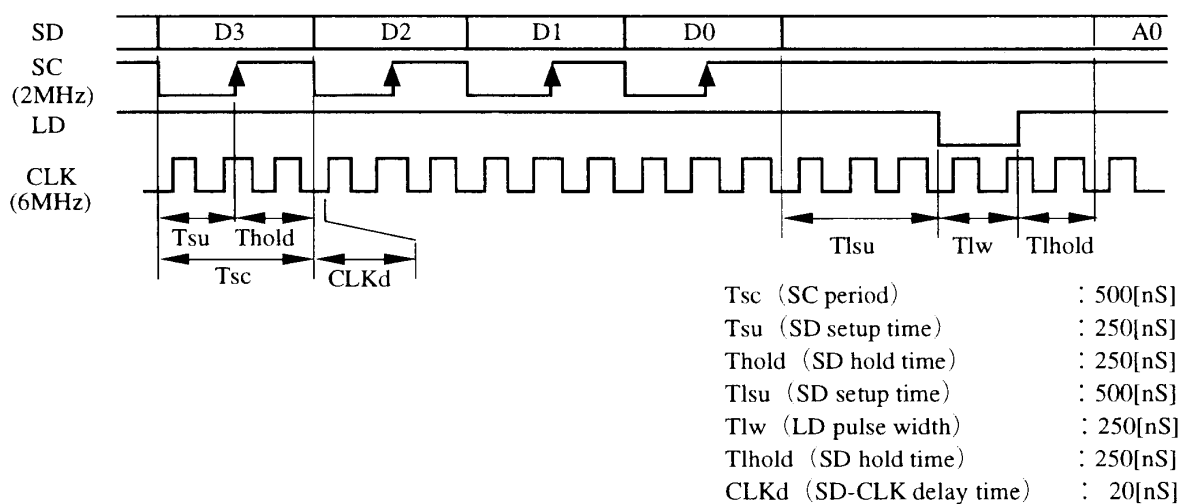


図3 入力波形タイミング

表1 入出力端子機能

端子名	名 称	I/O	備 考
SD	シリアルデータ	I	アドレス, PWMデータ
SC	シリアルクロック	I	シフトレジスタのクロック (2MHz)
LD	ロード	I	
CLK	システムクロック	I	基準クロック (6MHz)
RST	リセット	I	ローアクティブ
DA0	PWM出力	O	アドレス0のPWM出力
DA1	PWM出力	O	アドレス1のPWM出力
DA2	PWM出力	O	アドレス2のPWM出力
⋮	⋮	⋮	⋮
DA6	PWM出力	O	アドレス6のPWM出力
DA7	PWM出力	O	アドレス7のPWM出力

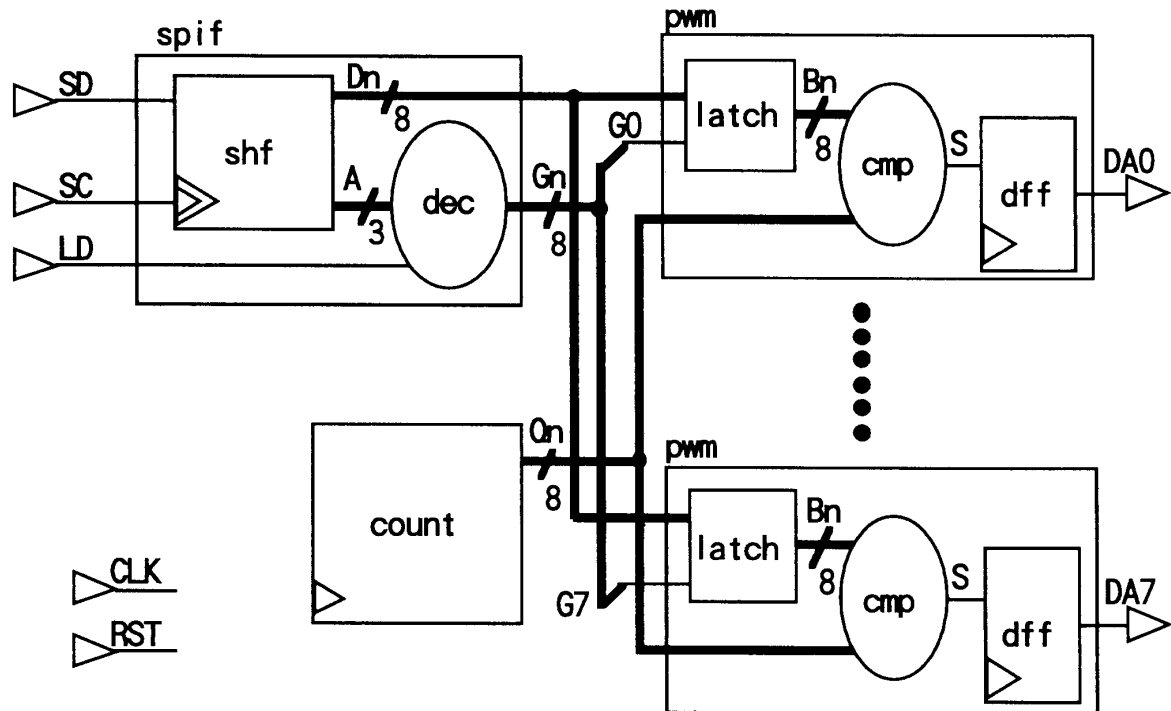


図4 8ch PWM 回路 RTL 図

ップ (dff) に入力し CLK と同期させ、DA0～DA7 に出力する。

4.4 機能記述

作成した RTL 図を参照し機能記述を行う。Verilog-HDL では一つの機能を果たすまとまりをモジュールと呼ばれる単位で定義し、各モジュール毎に機能の記述を行い、最終的に全体をまとめる記述を行う。モジュールにおける各入出力ポートは、シフトレジスタ等の記憶素子や組み合わせ回路等、その属性によりあらかじめ宣言される必要がある。モジュール内に複数または単数のモジュールを持つ場合、C 言語における関数と同様な記述をすることにより下位モジュールを呼び出すことができ、階層化設計が可能である。⁴⁾

また、Verilog-HDL ではテストパターンも機能記述と同様に一つのモジュールとして捉えられており、記述文法等は同じであるが時間の概念があるためテストパターンにのみ許される構文がある。信号の変化を逐次的または並列的に記述することによって、テストパターンを発生させることが可能である。さらに、Verilog-HDL ではシステムタスクが用意されており、期待値比較や波形の表示等、様々なシミュレーション情報を得ることが可能である。ここで作成したテストパターンは、次に述べる機能検証を行う RTL シミュレーションだけでなくゲートレベル仮想シミュレーションでも同じものを使用する。

Verilog-HDL では一つの機能を広範な記述レベルで表現することができるため、抽象的な記述では論理合成ができない場合がある。現在の論理合成技術ではモジュール間の接続情報を示す構造情報の全く無い記述から最適な回路を得ることは困難であり、設計者は半

導体プロセス技術やデバイス技術に依存しない範囲で構造情報を記述し、論理合成を意識した記述を行う必要がある。例として8ビットカウンタの機能記述を図5に、8ビットカウンタのテストパターンの記述を図6に示す。

4.5 RTLシミュレーション

RTLシミュレーションを行い、機能記述が仕様を満たしているか確認する。この段階で仕様を満たしていなければ、再度仕様を検討し前行程にもどる必要がある。早い段階で仕様を満たしているか確認できるため、開発期間を短縮することが可能である。シミュレーションにはCADENCE社のVerilog-XLを用いた。8ビットカウンタのシミュレーション結果を図7に示す。仕様を満たされているのが確認できる。

5. 論理合成

5.1 制約条件

入力波形の条件等を仕様に基づいて設定し論理合成を行なった。論理合成にはCADENCE社のSynergyを用い、制約条件には回路面積最小を設定、目的とするゲートアレイのライブラリは0.8 μ mのものを使用した。図8に8ビットカウンタの合成結果を示す。回路の出力部に8個のレジスタ(FDDS)が合成されているのが確認できる。

5.2 ゲートレベル仮想シミュレーション

論理合成を行った結果に対して、再度テストパターンを入力し仕様を満たしているか確認する。シミュレーションにはRTLシミュレーション様Verilog-XLを用いた。ゲートレベル仮想シミュレーションでは、仮想的に各ゲートの遅延時間および配線領域部分の遅

```

module count (CLK, RST, Q);           //モジュール宣言
input CLK;                             //入力ポート宣言
input RST;                             //入力ポート宣言
output [7:0] Q;                       //出力ポート宣言
reg [7:0] Q;                          //Qを8ビットのレジスタ型で定義

always@ (posedge CLK or negedge RST) //CLKの立ち上がりまたはRSTの立下がり
begin                                 //以下の処理を行う
    if (!RST)                        //ローアクティブのリセット動作記述
        Q<=8'h00;
    else if (Q==8'hff)               //カウンタの動作記述
        Q<=8'h00;
    else
        Q=Q+8'h01;
end
endmodule                             //モジュールの記述終了

```

図5 8ビットカウンタの機能記述

```

module test;                                //テストモジュールの宣言
reg CLK, RST;                               //CLK, RSTはレジスタ型で定義
wire [7:0] Q;                               //Qをwire型で定義
count top (Q, CLK, RST);                   //テストするモジュールの呼び出し

initial begin                               //全シミュレーション時間の定義
#2000 $stop;
end

initial                                     //CLK信号の記述
begin
    CLK=1'b0;                               //CLK delayの記述
    #20 CLK=1'b1;
    forever begin                           //CLKは6 MHz より周期は約166ns
        #83 CLK=1'b0;                       //よって83ns ごとに 0 と 1 を繰り返す
        #83 CLK=1'b1;
    end
end

initial begin                               //RST信号の記述
    RST=1'b0;
    #400 RST=1'b1;
end

endmodule                                  //テストモジュールの記述終了

```

図6 8ビットカウンタのテストパターン記述

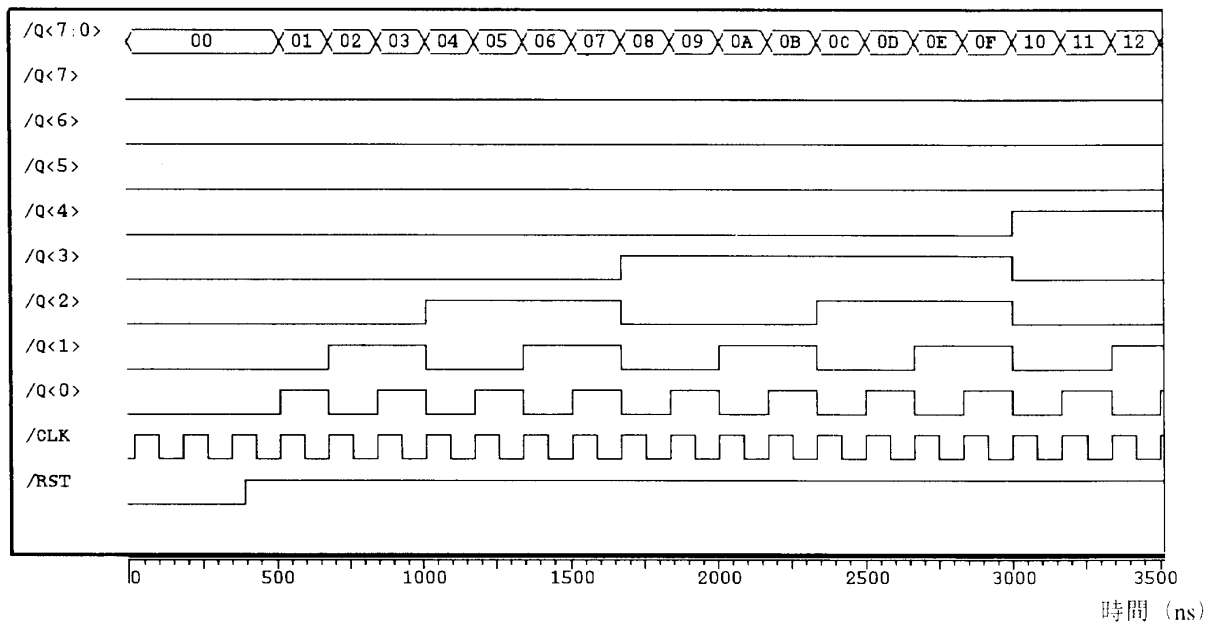


図7 8ビットカウンタのRTLシミュレーション結果

延が計算される。遅延を考慮した上でも仕様を満たしているか検討されなければならない。図9に8ch PWM回路のゲートレベル仮想シミュレーション結果を示す。入力データは出力DA0~DA7が順に0, 1, 2, 3, fc, fd, fe, ffを出力するよう設定した。要求出

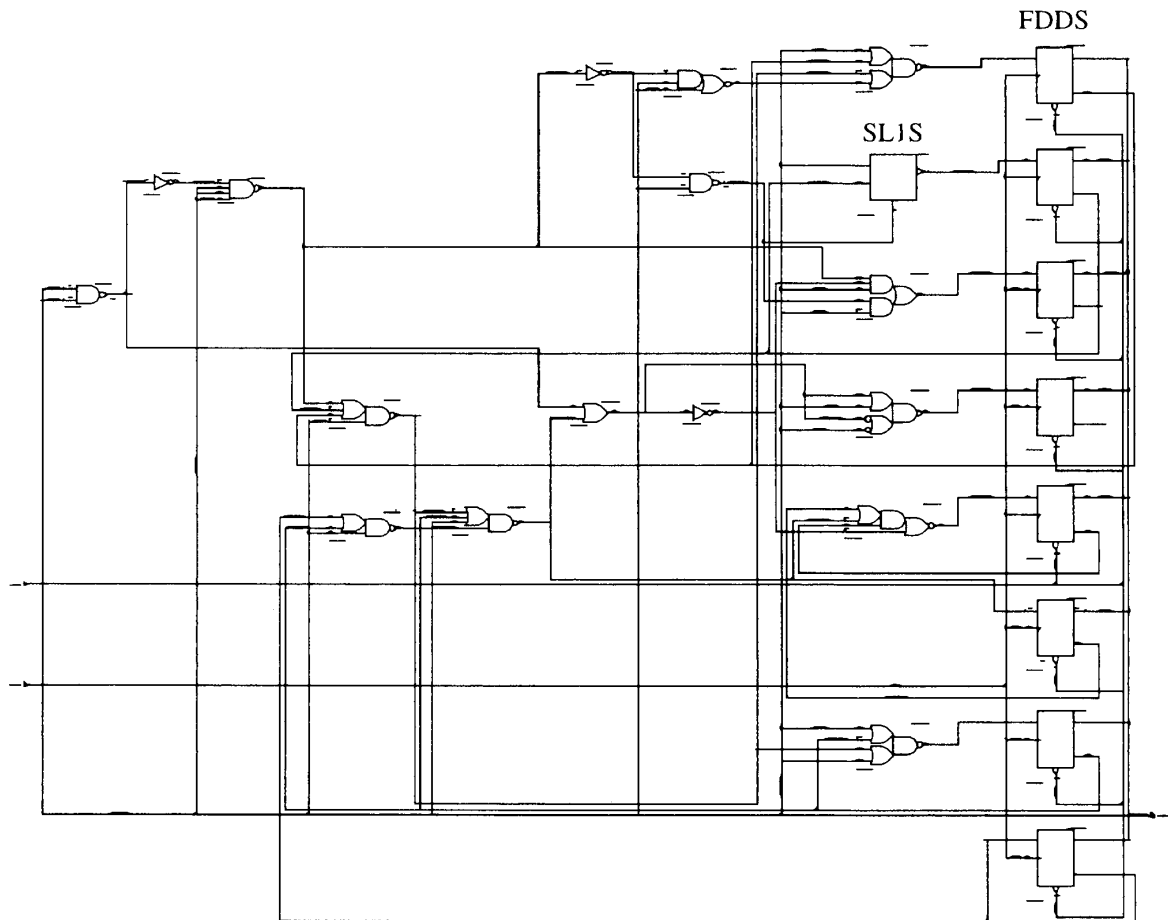


図8 8ビットカウンタの合成結果

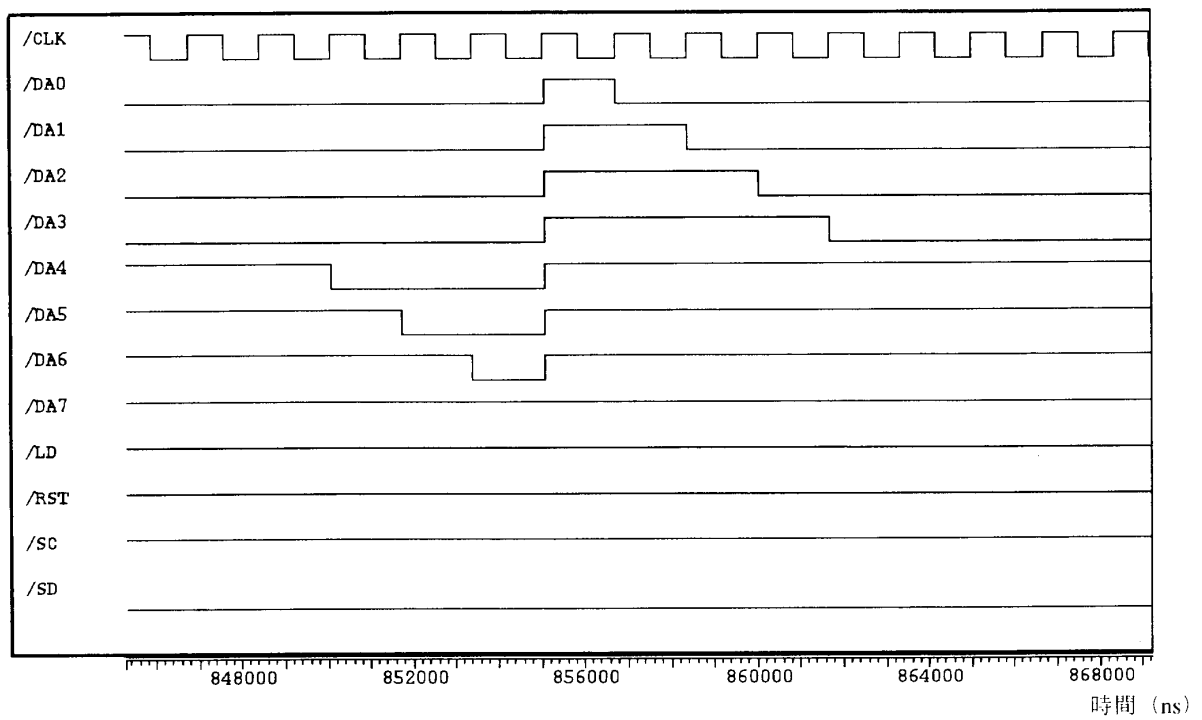


図9 PWM回路のゲートレベル仮想シミュレーション結果

力が得られており、仕様が満たされているのが確認できる。

6. ま と め

ハードウェア記述言語 Verilog-HDL を用いた 8 ch PWM 回路の論理合成を行い、合成された回路は仕様を満たしていることが確認できた。論理合成をするためには、合成回路を意識した機能記述をする必要があるため記述に制限があるが、回路機能検証の問題やミスの早期発見という特徴を考慮すれば開発にかかる時間は短いと考えられる。また、本研究で対象とした回路は規模が小さく(2000ゲート程度)、論理合成時の面積最小と速度優先の合成結果には差異がほとんどみられなかった。今後はさらに大規模な回路について検討するとともに、記述による合成結果の差異についても検討する。

参 考 文 献

- 1) 唐津, 星野, 石浦, 安浦: 論理合成時代のハードウェア記述言語: UDL/I, 電子情報通信学会論文誌, Vol. J74A, No. 2, pp. 170-178 (1991).
- 2) Thomas, D. E. and Moorby, P.: The Verilog Hardware Language, Kluwer Academic Publishers (1989).
- 3) Sternheim, E. et al. (井上・鈴木訳): Verilog HDL によるトップダウン設計, CQ出版, pp. 135-153 (1994).
- 4) Thomas, D. E. and Moorby, P. (飯塚・浅田訳): 設計言語 Verilog-HDL 入門, 培風館 (1995).

Logic Synthesis for 8 channel Pulse Width Modulation with Verilog-HDL

Junji AMAYA and Akiharu TADA*

Graduate School of Engineering,

**Department of Information and Computer Engineering,*

Okayama University of Science,

Ridai-cho 1-1, Okayama 700, Japan

(Received October 7, 1996)

Recently, LSI technology have been progressed to be widely use. Systems are highly complex and LSI have over a million elements. Inspection of its function becomes very difficult. And the costs of LSI design such as Application Specific Integrated Circuit (ASIC) became higher cause of various kinds and small quantity production. Rapidly LSI design and low costs method of LSI design have been required. Hardware Description Language (HDL) solves these problems. In LSI design, HDL method provides high quality with computer-aided design tools. Verilog-HDL is one of a HDL provides the system designer with a wide range of levels of abstraction, and, at the same time, provides access to computer-aided design tools to aid in the design process at these levels.

This paper describes process of synthesis for 8 channel Pulse Width Modulation circuit with verilog-HDL.