

プログラマブル・ロジック・デバイスの導入による 電子回路の LSI 化

2006 年 3 月

大井 正行 (研究支援推進員) , 福士 博樹 (技術部先端技術支援室)

我々ユーザーが考えたデジタル回路を自由に LSI として実現できるプログラマブル・ロジックデバイス PLD を使用すれば、さまざまな回路が 1 チップで実現できます。

本報告では、書き込み可能な CPLD に 1 秒間隔で 3 桁の 10 進表示を行うデジタル回路の LSI 化を試みました。LSI の設計については、パソコン上でハードウェア記述言語 HDL を用いて行いました。これを HDL トレーナにより CPLD に転送して書き込むことによって、本研究所技術部で最初の LSI が誕生しました。

1. はじめに

我々技術部がこれまで行ってきた計測・制御などを行う場合に必要なハードウェア (デジタルのプリント基板) は、AND-OR やフリップフロップ IC 等のいわゆる汎用ロジック IC(TTL/C-MOS) を使っていました。しかし、近年 PLD(CPLD/FPGA) の登場により LSI がより身近なものになるにともなって、これ等のインフラが安価で整えられるようになりました。

CPLD(Complex Programmable Logic Device) とは、我々ユーザーが設計したロジック回路をパソコン上で HDL を用いて何度も書き込める LSI のことです。我々はこの CPLD に比較的小規模のロジック回路が書き込まれた LSI の開発を目的としています。

このような理由から、我々はこれまでの汎用ロジック IC の利用から脱皮して CPLD へ移行しようとするものです。

2. 開発環境

今回おこなった CPLD の開発には、PC (Windows XP) の他に開発ツールとして以下に示すようにソフトウェアとハードウェアを必要とします。

- ソフトウェア (論理開発ツール)
 - ・ Xilinx ISE WebPACK Ver.7.1.02i (無償版) : PC 上で Project Navigator 画面を起動し、このツールを操作して開発を行います。
- ハードウェア (評価ボード)
 - ・ 評価ボード (HDL トレーナー) : Sophia Systems 社の XC9500 Evaluation Board を使用しました。
 - ・ ダウンロード・ケーブル : JTAG 方式で PC のパラレルポートと HDL トレーナー間を接続します。

3. 実験方法

3.1 3 桁 10 進表示回路の概要

実験回路の試みとして、1 秒のクロックパルスで歩進する “3 桁の 10 進回路” を HDL で記述することにより 1 チップで LSI を設計しました。この回路は図 1 に示すように、点線で囲った Watch1000 の部分を CPLD に置換え LSI 化しました。使用した評価ボードには、CPLD、クロック発信器、スイッチ類および 7 セグメント LED 表示器が 3 個あるので、これ等をそれぞれクロック入力 CLK、押しボタン用リセット・スイッチ入力 RESET および 3 桁の表示出力 LED として利用しました。

最初の divider17 は、評価ボードにあるクロック発信器から 500KHz の周波数を受け、これを 17 分周して約 1 秒のクロックパルス出力とします。続いて counter1000 は、10 進カウンターを 3 個接続した 1000 進カウンターと考えることができます。この出力をそれぞれ decoder8 へ接続します。

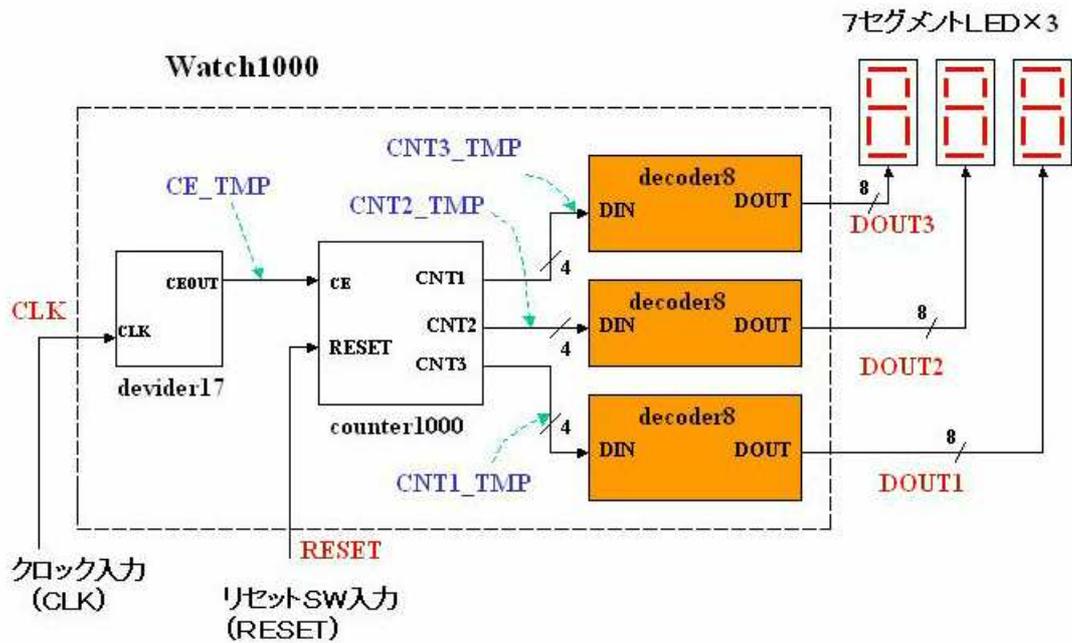


図1 3桁10進表示回路の構成

3.2 CPLD 開発の流れ

図2 に開発の流れを示します。最初の「仕様設計」は、従来の汎用ロジック IC を使った設計の場合の考え方と全く同じです。次の「コード記述」から「ダウンロード」までは、PC 上で“HDL 開発ツール”による Project Navigator 画面の操作です。

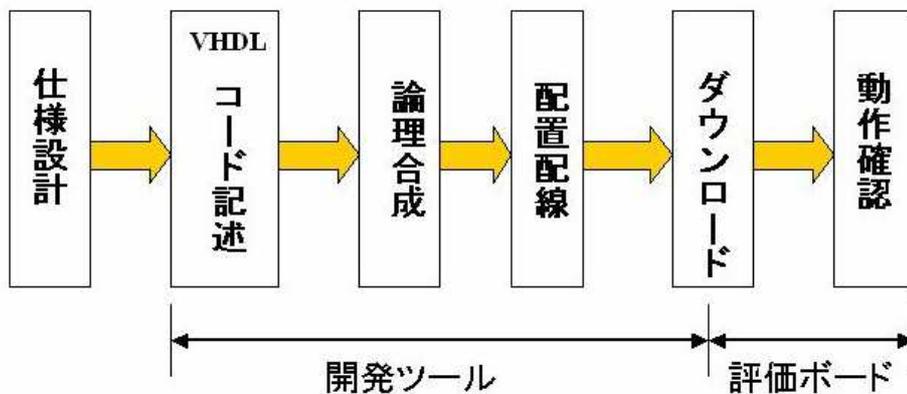


図2 開発の流れ

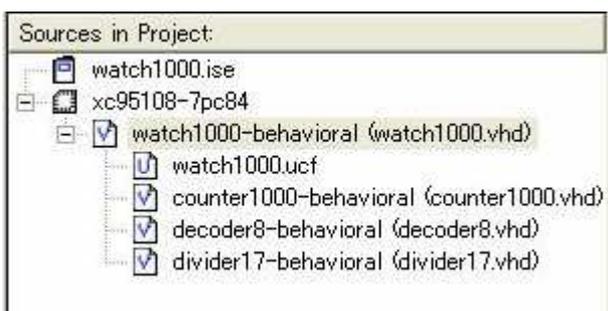


図3 コードファイルの階層構造

「コード記述」は、図1 の点線で囲まれた watch1000 の中を記述します。ここでは、デコーダ decoder8 は同じものを3個使用しています。そのため 図3 に示すように、watch1000.vhd をルートにしてこの下にそれぞれ 17 分周回路 divider17、1000 進カウンタ counter1000 およびデコーダ decoder8 から構成された階層設計にしました。

これによってデコーダ回路を3回記述することによるコード記述の煩雑さを避けることができます。したがって、ソースコードを合計4ファイル記述することになります。「論理合成」はコード記述をコンパイルするところです。コンパイルでエラーがあればコードを修正し再度コンパイルします。

続いて「配置配線」ではCPLD (XC95108) へのピン割り当てを行い、watch1000と該当する入出力を設定します。この結果watch1000.ucfファイルができます。

「ダウンロード」ではHDLトレーナへ書き込みを行うためのwatch1000.jedファイルを作成します。これができれば、後はCPLDへ書き込みをするだけです。最後に、評価ボードにある書き込まれたCPLDの「動作の確認」を行います。これが正常であれば完了です。

4. 結果および考察

試作した3桁10進表示用LSIのコード記述で、図1に示すcounter1000中の10進カウンタ2ヶ所の桁上げ(Ripple Carry)に最も時間を要しました。使用した開発ツールは、論理合成がパスした段階で必要があればその回路図を表示することができます。図4はコード記述から得たcounter1000の回路図の概略です。

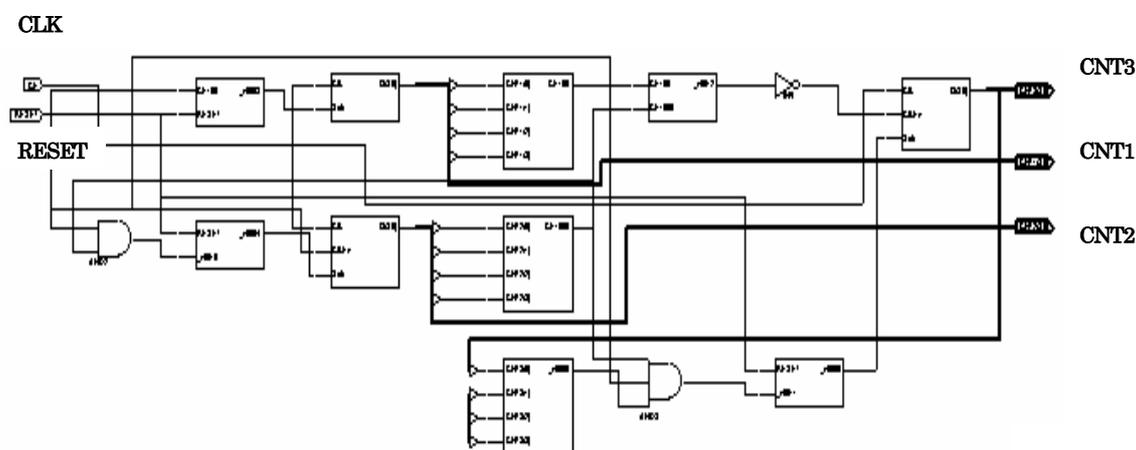


図4 開発ツールによるCounter1000の回路図

使用したデバイスは、Xilinx社のCPLD、XC95108です。この回路図で使用したセル数は、論理合成の結果からAND,OR,XOR等が合わせて370個、フリップフロップ数は分周回路の17ビットカウンタで17個、10進カウンタ回路に4ビットカウンタが3個で12個、計29個使用したことになります。これは使用したCPLD全容量の約1/5で納めることができました。

5. おおひらき

PLDを使ったロジック回路の試作にとりあえず第1歩を踏み出すことができました。記述した4つのコードファイルは紙面の都合で省略します。実際に試作して感じ得たことは、従来の汎用ロジックICの設計の経験とプログラム上でロジック回路とコード記述を如何に結びつけることができるかです。

今回の試作では行いませんでしたが、HDLで設計した回路をダウンロードしてCPLDを動作させる前に、ソフトウェア上で回路の動作を論理シミュレーションする機能があります。これによって得られたタイムチャートを検討して、今後のより複雑な回路設計に対応したいと考えます。