

ワンチップ・マイコンを応用した制御回路設計の技術修得

メタデータ	言語: Japanese 出版者: 公開日: 2013-06-11 キーワード (Ja): キーワード (En): 作成者: 本堂, 義記, 酒井, 孝則, 岡井, 善四郎, 田畑, 功, 小川, 勇治 メールアドレス: 所属:
URL	http://hdl.handle.net/10098/7462

ワンチップ・マイコンを応用した制御回路設計の技術修得

専門研修受講者 本堂 義記、酒井 孝則、岡井 善四郎（第三技術室）、
田畑 功（第二技術室）、小川 勇治（第一技術室）

1. はじめに

本専門研修受講者 5 名は平成 10 年度より本年度までの 5 年間、デジタル回路の基礎からワンチップ・マイコンの PIC (Peripheral Interface Controller) の基礎、さらに PIC の機能の不足部分を補うプログラマブル・ロジックの CPLD (Complex Programmable Logic Device) についての基礎技術修得を行うとともに、それらに関する多種のモデル回路の設計・製作実習を行ってきた。

本年度は CPLD を動作させる VHDL (Very high speed integrated circuit Hardware Description Language) 言語の RTL (Register Transfer Level) 記述によるロジック回路の生成法について技術修得を行い、さらにこの 5 年間の専門研修で蓄積したデジタル回路技術の応用として、PIC と CPLD を搭載した周波数カウンタ¹⁾ を製作実習課題として選択した。

2. VHDL 言語による RTL 記述^{3) ~ 5)}

VHDL は HDL (ハードウェア記述言語) のひとつであり、電子回路システムのモデル化と文書化を意図して作られた言語で、一般のソフトウェア言語とは異なりハードウェアを意識した言語構造を有している。すなわち、VHDL は CPLD や FPGA (Field Programmable Gate Array) などのプログラマブル集積回路の設計において、ロジックシンボルをテンプレートなどで回路を描く代わりにプログラムの形で記述する言語であり、本専門研修ではこの言語の中で論理合成可能なゲートレベルやクロックを意識した RTL 記述についての研修を行った。

2. 1 VHDL によるロジック回路設計

図 1 に簡単な VHDL 記述の具体例として半加算器の論理合成回路を示す。図のデジタル回路を設計するには、

- (1) どのような信号が A、B に入力されるのか？
- (2) 入力信号に、どのような加工を施すのか？
- (3) どのような信号を C、S に出力させるのか？

を明確にする必要がある。VHDL では (1)、(3) に関する情報 (インターフェースに関する情報) はエンティティ (entity) 節で、(2) に関する情報 (内部の動作・構造に関する情報) はアーキテクチャ

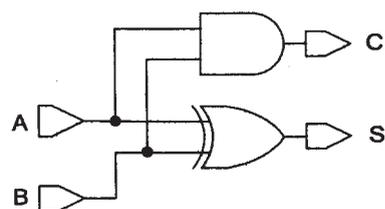


図 1 論理合成回路

(architecture) 節で別々に記述する。

2.2 VHDLによるプログラム例

図2にロジック回路において一般的によく用いられるDフリップ・フロップ(以後D-FFと略)素子の動作を表すVHDLプログラム例を示す。

プログラムの①はライブラリ宣言であり、②のポート名で用いるデータタイプに必要なパッケージの呼び出しを行う。

つぎに②の entity 宣言ではD-FFの入出力信号をポート名で記述し、その後続く()の中に信号名、信号方向、信号のデータタイプを指定する。

③の architecture 宣言ではD-FFの内部動作を記述する。この宣言では“architecture” “begin” と “end” の間に直接信号代入文やプロセス文などを記述し、その動作は同時並列もしくは順次動作として処理される。

④の process 文はつぎに続く()の中の信号の値が変化した時に活性化され、記述の上から順次処理される。この()内の信号をセンシティブティ・リストと呼び、最終行 “end process” まで実行すると最初に戻り、()内の信号が変化するまで動作を停止する。

⑤の if 文は process 文の中で記述することができ、他に case 文や for-loop 文などがあり、elsif の後の

()内記述はクロック CLK の立ち上がり動作を指定している。立ち下がり動作の場合は ‘1’ が ‘0’ に代わる。また、elsif 文の行は “else (wait until CLK’ event and CLK= ‘1’)” と記述することもできるが、その場合はセンシティブティ・リスト内の CLK を記述しない。この wait until 文はクロックの立ち上がり信号が入力されるまで、順次処理文を停止させる。

以上のような信号処理を行うことでD-FFの特徴であるラッチ動作が if 文あるいは wait until 文によ

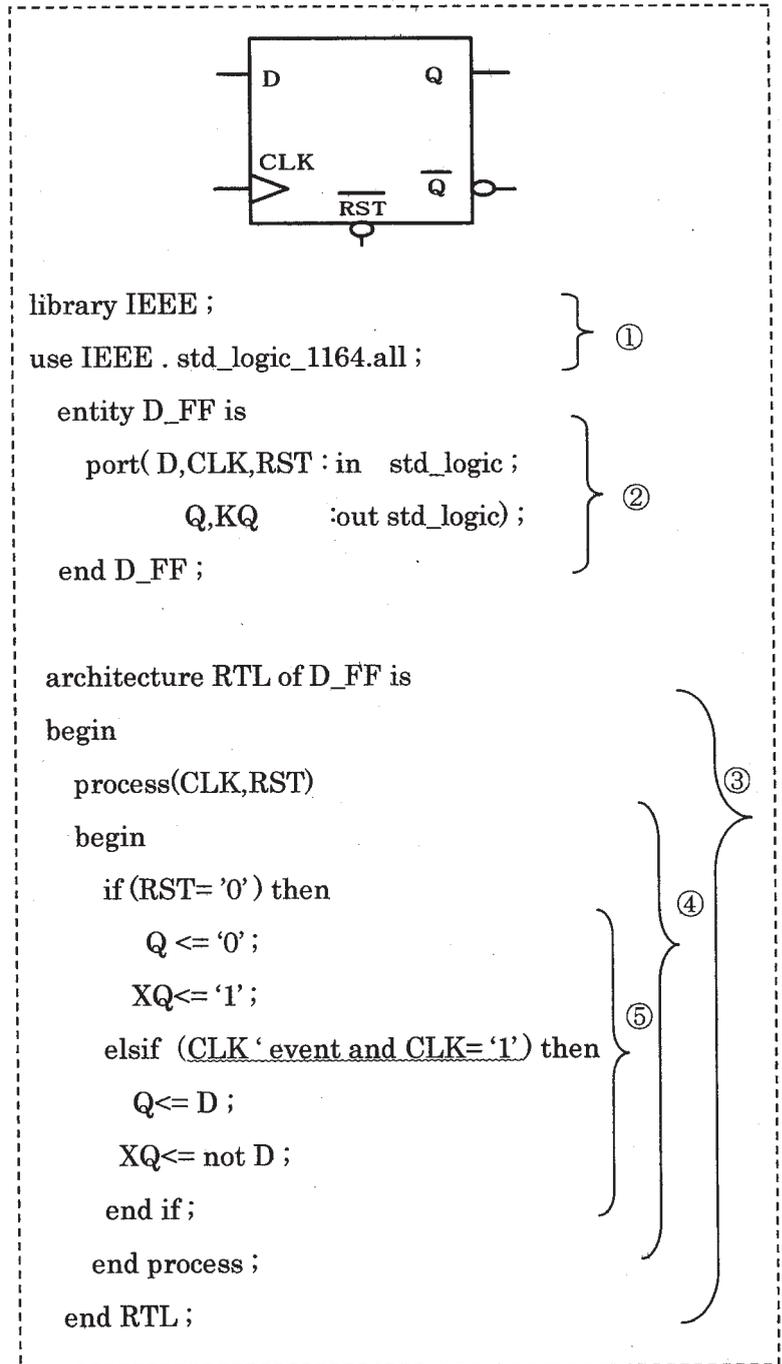


図2 Dフリップ・フロップのVHDLプログラム

り実現でき、入力データ信号 D の状態をクロック信号の立ち上がり信号によりラッチし、出力することが可能となる。

3. CPLD を使用した製作実習

周波数カウンタの設計は従来の専門研修で技術修得した PIC のみでも可能であるが、周波数の測定上限が PIC の性能により 20 MHz 以下となり、それより高い周波数を測定する場合は応答周波の早い CPLD などを使用して高速化を図る必要がある。そこで、本実習では参考文献 1 に記載されている周波数カウンタの製作を研修課題とし、カウンタ本体は CPLD で、表示部は PIC で制御する 50MHz 以上測定可能な周波数カウンタの設計方法および製作実習に関する技術修得を行った。

3. 1 周波数カウンタの構成

カウンタに使用した CPLD は Xilinx 社の XC9572-15 (最大動作速度 83.3 MHz、44 ピン)、PIC は Microchip Technology 社のミッドレンジファミリー PIC16F873 (28 ピン) を使用した。周波数カウンタの特性を表 1 に示す。ただし、表の周波数測定範囲上限 70 MHz は推測値である。これは利用可能な周波数発振器の上限周波数が 20 MHz で、これ以上の周波数での動作確認ができないためである。このため、後で述べる CPLD の「WebPACK ISE」による合成結果の動作タイミングより推測を行った。

周波数測定範囲	1 Hz～70 MHz
計数時間	1 sec
表示単位	Hz (表示部 8 桁)
入力電圧範囲	0.1～10 v

表 1 周波数カウンタの特性

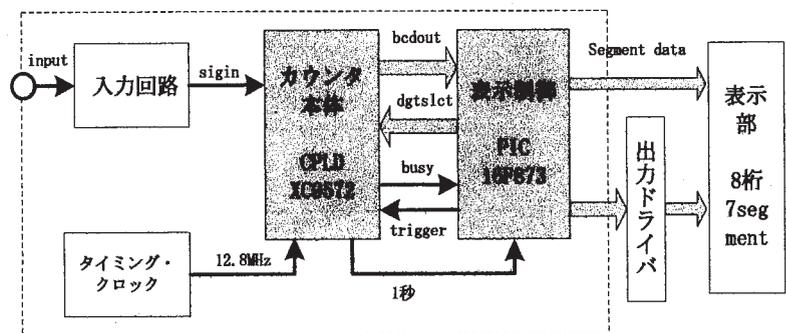


図 3 周波数カウンタ全体構成

周波数カウンタ全体構成を図 3 に示す。カウンタ本体 (CPLD) に

入力する測定信号(input)は入力回路中のスイッチング時間の早い MOSFET により増幅した後、応答時間の早いインバータ 74AC14 で波形整形を行い、カウント信号 (signal) として入力する。

計数時間は 1 秒間とする。ここでは、高精度の水晶発振器(12.8MHz) を使用して正確な 1 秒間の計数時間をタイミング・クロックにより作る。入力周波数測定は、表示制御(PIC16F873)からの trigger 信号で入力ゲートを開き開始される。また、ゲートが開いている時 (カウント中) は busy 信号をオンにして、その間 PIC は以前のデータをラッチして表示している。

計数が終了すると busy 信号をオフにしてカウンタ本体に取り込んだ計数データに更新して表示する。表示データの inputs は PIC からの桁選択信号 (dgtslct) に応じて、順次カウンタ本体から BCD (Binary Coded Decimal code) データ (bcdout) 8 桁分を読み込み、セグメントデータに変換して、8 桁 7segment LED に表示する。また、表示は LED のちらつきをなくすために、カウント中 (1 秒間) も含め約 2 ms

ごとに1桁ずつ制御信号を出力して8桁分を繰り返し表示している。

3. 2 VHDL による CPLD 設計法

図4に示すカウンタ本体であるCPLD回路構成の動作タイミングは次の手順となる。

(1) 開始指令信号 (trigger= '0') を外部より入力する。

(2) trigger 信号により内部カウンタ (24ビットハインカウンタ、BCD8桁カウンタ)、内部ビジー (intbusy= '0') 信号をクリアし、同時にゲート (GATE) を閉じる。

また、内部ビジー信号を出力ビジー (busy) 信号に代入し、その信号により表示制御部 (PIC) のデータをラッチする。

(3) タイミング・クロック信号の最初のクロック波形の立ち上がりによりゲートを開きカウント信号を入力し、BCD8桁カウンタでカウントする。

また、trigger 信号をカウンタリセット用に取り込む。

(4) タイミング・クロック信号が1秒間、カウントするとゲートを閉じ、さらに内部ビジー信号、出力ビジー信号を '1' にして表示制御部へのBCD8桁カウンタデータ転送を可能とする。

(5) 表示制御部より出力された桁信号 (dgtslct) の入力により、1桁ずつラッチされた4ビットカウンタデータを出力し、8桁分が終了すると(1)に戻る。

以上のVHDL記述によるプログラムの流れを図5に示す。

プログラム中においてCPLDの回路構成に関する信号は、入出力信号の記述中のportの()中で信号名、方向、データ

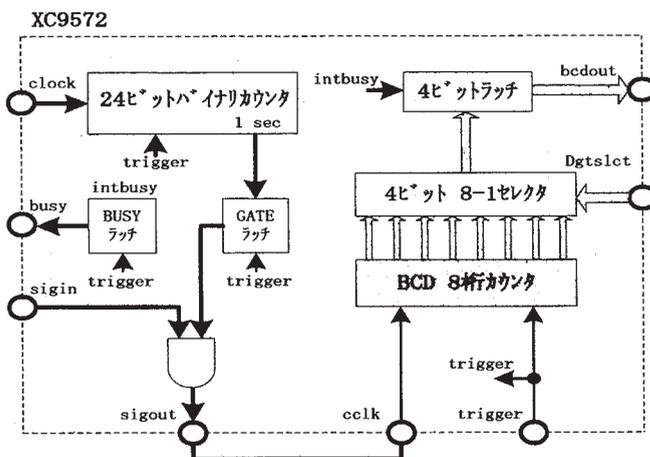


図4 CPLDの回路構成

```

library IEEE ;
use IEEE . std_logic_1164.all ;
use IEEE . std_logic_unsigned.all ;

entity fcounter is
    port (信号名、方向、データ型) ;
end fcounter ;

architecture RTL of fcounter is

    component count10
        port (信号名、方向、データ型) ;
    end component ;

    process (変化する信号のみ記述)
        処理文 (スタート開始) ;
    end process ;

    ↓
    この間に幾つかの
    ↓
    プロセス文を記述

    digit0:count10 port map (コンポーネント文で
        記述した信号名と関連) ;

end RTL;
    
```

図5 VHDL記述によるプログラムの流れ

タイプのすべてが記述される。また、動作タイミングに関するロジックの記述については幾つかプロセス文により記述される。

BCD カウンタ部分は 8 桁分のカウント繰り返しとなるためコンポーネント文による構造化記述を行い最終行近くの count10 の port map () で digit0 から digit7 まで 8 桁分を処理する。また、コンポーネント文として指定した count10 のリセット付き BCD カウンタのプログラムは本プログラムとは別に準備しリンクする必要があるが、ここでは紙面の関係で説明を省略する。

なお、プログラム中のライブラリ宣言 “use_logic_unsigned.all;” は、符号ビットなし算術演算を行うために必要な関数を呼び出すための構文である。

3. 3 PIC と表示部

表示制御部に用いた PIC のプログラムは Microchip Technology 社のアセンブラ「MPASM」のフォーマットより作成され、そのフローチャートを図 6 に示す。

図において、まず PIC の I/O ポートの初期化を行う。この PIC には A/D 変換器が内臓されている関係で I/O ポートの一部がデフォルトでアナログ入力になっている。そこでこれをデジタル I/O モードに変更し、CPLD に対して瞬時パルスの trigger を出力する。

その後は busy 信号が閉じられるまでメモリ内より 1 桁ずつの BCD データを取り出し、セグメントデータに変換して表示用の LED に出力し、2 ms 時間を置いて次の桁に移る流れを繰り返している。ただし、最初のデータは 0 以外のデータを入力するまではブランキング制御を行っている。

busy 信号が閉じると、瞬時に CPLD より 8 桁分の BCD データを PIC 内のメモリに入力し、終了すると開始信号 trigger を CPLD へ出力する。

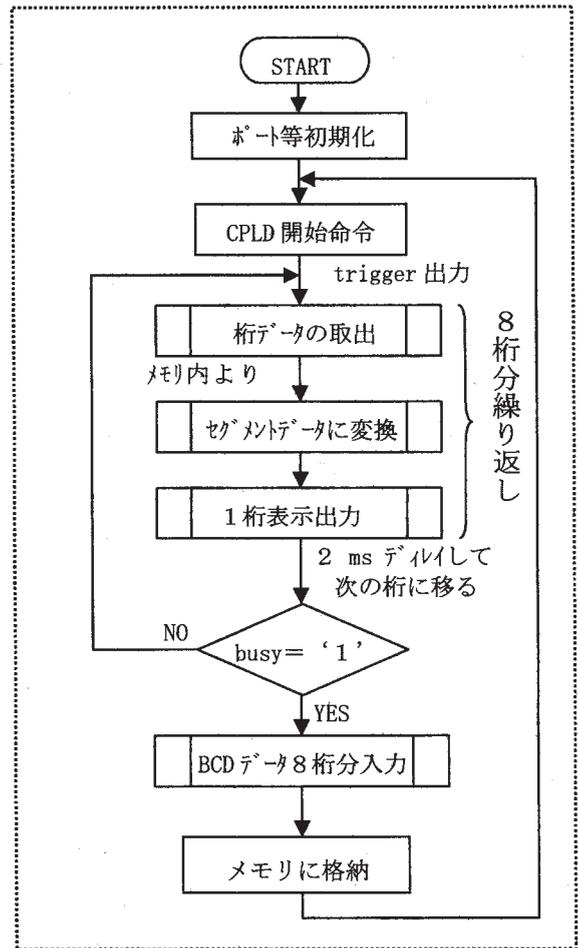


図 6 表示部制御フローチャート

3. 4 周波数カウンタの製作

最初に周波数カウンタ構成の中心である CPLD と PIC で使用するプログラムの動作および設計法について技術修得を行った^{1), 2)}。

CPLD へのプログラム書き込みは、Xilinx 社が無料で提供している開発ツール「WebPACK ISE」と昨年度の専門研修で製作したダウンロードケーブル (XCKIT HUMANDATA 社) を使用した。また、PIC へのプログラム書き込みは、2 年前の専門研修で製作した PIC ライターを使用した。

つぎに、入力回路、カウンタ本体、表示制御部、表示部への出力ドライバを 1 枚の基板にレイアウト

ウトし実装を行った。また、表示部の8個の7セグメントLEDおよび制御部、表示部への電源(DC5V)はそれぞれ別の基板に実装を行い、最後に3枚の基板をケースの中に納めてそれぞれを結線して完成した。図7に製作した周波数カウンタを示す。

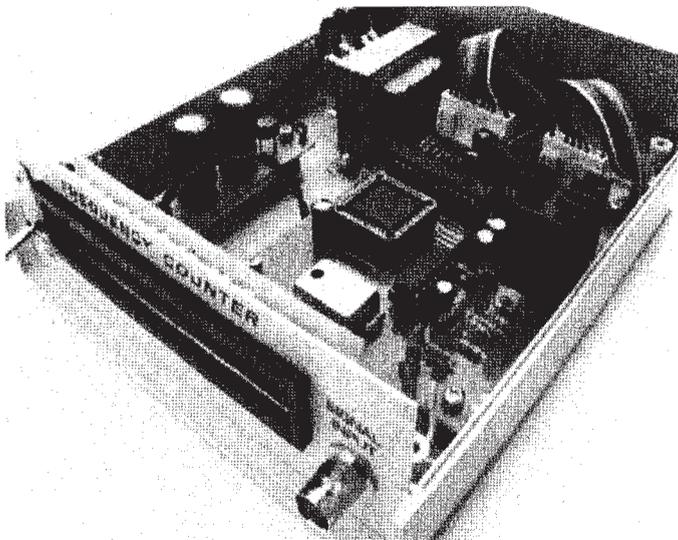


図7 製作した周波数カウンタ

実装に関しては、専門研修歴5年間の経験を生かし短時間で完成することができたが、回路構成全体および単独素子(CPLD、PICなど)の動作原理の理解には意外と苦労した。

これは、同メンバーで5年間にデジタル回路の基礎からハードウェア記述による大規模プログラマブル回路設計の基礎まで研修内容を広げすぎたことが起因していると考えられる。

なお、各プログラムの理解および設計法については専門研修日に共同して技術修得を行い、実習に関しては各自が空き時間を利用して製作を行った。

4. あとがき

今回の専門研修は参考文献のモデル回路を利用し、その回路動作原理、設計・製作方法の理解に努めた。すなわち、物づくりに最小限必要な最初の回路構成から設計、つぎに回路部品とレイアウトを行い、最後に製作して実験する技術者として一連の流れを各自が経験する研修を行った。

今後は得られたデジタル回路関連の基礎技術修得を派遣先の職場で応用技術として発揮していくことができれば、5年間同メンバーで続けた専門研修は大いに有意義であったと考えられる。

研修日程

9月1日	研修計画と役割分担
9月8日	購入物品・図書の検討、「PIC 応用ガイドブック」輪読
9月29日～11月10日(計5回)	「VHDLによるハードウェア設計入門」輪読と課題検討
11月17日～1月19日(計5回)	周波数カウンタ回路の製作と動作原理の解析
1月26日, 2月16日	研修報告原稿の検討と会計確認

参考文献

- 1) 後閑 哲也 : PIC 応用ガイドブック、技術評論社(2002)
- 2) 後閑 哲也 : PIC 活用ガイドブック、技術評論社(2000)
- 3) 長谷川 裕恭 : VHDLによるハードウェア設計入門、CQ出版社(2002)
- 4) 吉田 たけお、尾知 博 : デジタル回路設計、CQ出版社(2002)
- 5) 湯山 俊夫 : デジタルIC回路の設計、CQ出版社(1998)