

卒業研究報告

題目

初期設定可能デジタル時計の設計、製作、評価

指導教員

原 央 教授

報告者

能勢 義和

平成 14 年 2 月 7 日

高知工科大学 電子・光システム工学科

もくじ

1 . 概要	- 1 -
2 . 全体回路設計	- 2 -
2 . 1 . 何を作るのか	- 2 -
2 . 2 . どうやって実現させるのか	- 2 -
2 . 3 . 回路シミュレーション (VHDL)	- 4 -
2 . 3 . 1 . VHDL とは	- 4 -
2 . 3 . 2 . 60 進カウンタの機能実現	- 4 -
2 . 3 . 2 . 1 . 10 進カウンタ	- 5 -
2 . 3 . 2 . 2 . 6 進カウンタ	- 7 -
2 . 3 . 2 . 3 . 60 進カウンタ	- 8 -
2 . 3 . 3 . 24 進カウンタの機能実現	- 10 -
2 . 3 . 3 . 1 . 12 進カウンタ	- 10 -
2 . 3 . 3 . 2 . 2 進カウンタ	- 11 -
2 . 3 . 3 . 3 . 24 進カウンタ	- 12 -
2 . 3 . 4 . 時計の機能実現	- 14 -
3 . 部分回路の詳細設計	- 16 -
3 . 1 . 標準 IC を用いて設計するための基礎知識	...	- 16 -
3 . 1 . 1 . IC パッケージ	- 16 -
3 . 1 . 2 . IC の性能	- 16 -

3 . 1 . 3 . どのICを使用するか	18
3 . 1 . 3 . 1 . メインIC	18
3 . 1 . 3 . 2 . フリップフロップ	19
3 . 1 . 2 . 2 . 1 . ラッチの基本	19
3 . 1 . 2 . 2 . 1 . Dフリップフロップ	20
3 . 2 . Multisimによる詳細シミュレーション	22
3 . 2 . 1 . Multisimとは	22
3 . 2 . 2 . カウンタ	22
3 . 2 . 2 . 1 . 60進カウンタ	22
3 . 2 . 2 . 2 . 24進カウンタ	24
3 . 2 . 3 . LED数字表示素子	25
3 . 2 . 4 . デコーダ	25
3 . 2 . 4 . 1 . 表示素子用デコーダ	25
3 . 2 . 4 . 2 . 24進デコーダ	27
3 . 2 . 5 . 発振器	29
3 . 2 . 5 . 1 . 発振器とは	29
3 . 2 . 5 . 2 . 水晶発振器	29
3 . 2 . 5 . 3 . 分周器	29
3 . 2 . 6 . スイッチ	31
4 . 回路製作	32
4 . 1 . 回路製作時の注意点	32
4 . 2 . 回路設計と回路製作の違い	34
4 . 2 . 1 . ICソケット	34
4 . 2 . 2 . 表示素子までの配線	34
4 . 2 . 2 . 1 . 実際に接続した配線	34
4 . 2 . 2 . 2 . つなぐべき配線	35

4 . 2 . 3 . 入力信号の注意	- 35 -
4 . 2 . 3 . 1 . ノイズ対策	- 35 -
4 . 2 . 3 . 2 . HとLを明確にするには	- 36 -
4 . 3 . スイッチ	- 38 -
4 . 3 . 1 . スイッチ配置	- 38 -
4 . 3 . 2 . スイッチ利用の注意点	- 40 -
5 . 評価	- 43 -
5 . 1 . 機能確認	- 43 -
5 . 2 . 時計としての精度	- 43 -
5 . 3 . 電源電圧	- 43 -
5 . 4 . 電源電流	- 45 -
5 . 4 . 1 . 電源の電流値	- 45 -
5 . 4 . 2 . 表示素子の電流値	- 46 -
5 . 5 . 製作初期の問題点	- 48 -
6 . その他	- 50 -
6 . 1 . 反省点	- 50 -
6 . 2 . まとめ	- 50 -
7 . 謝辞	- 52 -
8 . 参考文献	- 51 -

1. 概要

現在、エレクトロニクス分野の技術は目覚ましい速度で発展し、さらなる技術向上のため日々努力されています。今日の私たちの生活はその技術の成果の上に成り立っており、これ無しでは日常生活が成り立たないくらいにまでなっています。それを支えているのが、半導体技術とそれをを用いた電子回路の設計技術です。

半導体は「産業の米」と呼ばれ、現代社会が存続していく上でなくてはならないものとなっています。また、半導体の上に開花した IC (集積回路) は、いまやありとあらゆる電子機器に組み込まれており、むしろ IC が入っていない機器を探すほうが困難なくらい普及しています。

十数年前は IC チップ一つ数万円もしていましたが、半導体技術の信頼性の向上、大量生産により、その値段は驚くほど安くなりました。また、色々な品種の IC が開発され市販されており、高性能かつ小型の IC を手軽に入手できるようにまでなっています。その技術向上とともに、IC を用いた電子回路の設計技術も品質向上、小型化を目指し、技術の進歩はまだまだ続くでしょう。

このような状況の中で、私は今後、IC やそれをを用いた電子回路の設計の知識無しには身の回りの電気製品を理解することも新製品を開発することもできないと思い、具体的な例をとりあげて、電子回路の設計、製作、評価の一連の流れを習得し、最近の CAD 技術を習得することを卒業研究のテーマとして選びました。

具体的には、デジタル時計をとりあげて、デジタル回路の基礎知識習得、CAD 技術を習得し、全体回路の設計、部分回路の詳細設計を進め、それを標準 IC を用いた回路製作し、評価を行う手順で進めました。

以下には、この手順にしたがって、本報告書をまとめています。

2 . 全体回路設計

2 . 1 . 何を作るのか

製作の目標を明確にさせないといけない。何を、どういう製品を作りたいのか、考えたことを以下にまとめる。

- ・ コンセプト

利用価値があり、実際に使用できる機能を実現する。

- ・ 製品名

秒, 分, 時, を LED 表示するデジタル時計。

- ・ 特徴

持ち運びにはこだわらないが、全体的に小型。電源電圧は 5V とし、正確な時間を刻み、初期設定が可能。

2 . 2 . どうやって実現させるか

具体的な目標を決めたら、どのような方法で実現させるか考えなければいけない。また、どのように設計し、どのように動作させるか、大まかなイメージを持つことが大切である。

図 2-1 に初期設定可能なデジタル時計の概要を設計した結果を示す。

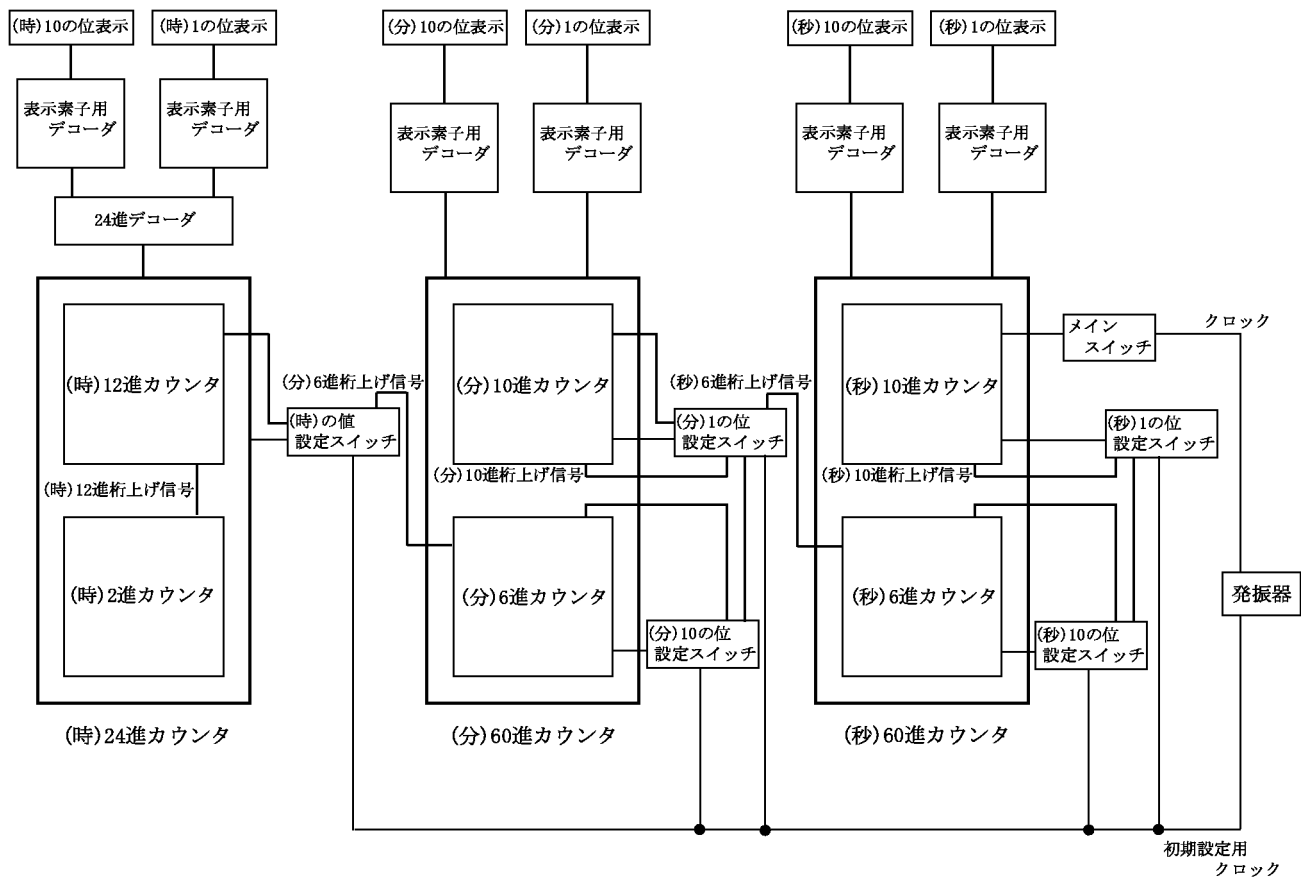


図2-1 全体のイメージ図

発振器の各出力は、時を刻むベースとなるクロック信号と初期設定用のクロック信号を必要とし、それぞれのスイッチのON、OFFで初期設定を可能とした配置となっている。各表示素子用デコーダは、(秒)60進カウンタ、(分)60進カウンタ、(時)24進カウンタの各出力信号を効率よく変換し、秒、分、時を表示するための表示素子へと接続される。24進カウンタには表示素子用デコーダの前に24進デコーダが必要である(詳しくは26ページの3・2・4・2・24進デコーダで説明する)。

以上のことよりデジタル時計動作のためには、60進カウンタ、24進カウンタの動作を得ることが必要不可欠だと分かった。この機能実現のために、まずVHDLで設計し、シミュレーションを行うことにする。

2.3. 回路シミュレーション (VHDL)

2.3.1. VHDL とは

HDL(Hardware Description Language)とは、日本語ではハードウェア記述言語と言われ、ハードウェアを設計するためのプログラミング言語です。HDLにも様々な種類があるが、代表的なものとして、Verilog-HDLとVHDLがある。Verilog-HDLは、米国ゲートウェイ社(現 Cadence 社)で開発されたHDLで半導体設計ではもっとも多く使われている。

一方、VHDLはVHSIC-HDLの略で、VHSICとは米国国防省のVHSIC(Very High Speed IC)プロジェクトで開発されたHDLで、システム開発にもっとも多く用いられている。

ここではVHDLを使用することにする。

2.3.2. 60進カウンタの機能実現

60進カウンタは図2-1から分かるように、10進カウンタと6進カウンタを組み合わせることによって実現することから、まず10進カウンタ、6進カウンタを記述し、各出力をシミュレーションする。

2 . 3 . 2 . 1 . 10進カウンタ

数を0から9まで10数える回路(D10)を以下のように記述する。入力信号として回路の動作をコントロールするCLK信号、ONにすることによって回路が動作するRST信号を使用する。また出力信号はQ0、Q1、Q2、Q3、COとする。

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity D10 is
port (
    CLK,RST: in std_logic;
    COUNT: buffer std_logic_vector(3 downto 0);
    Q0,Q1,Q2,Q3,CO: out std_logic
);

end D10;

architecture RTL of D10 is

signal COUNT_IN :std_logic_vector(3 downto 0);
signal Q0in,Q1in,Q2in,Q3in : std_logic;

begin

COUNT <= COUNT_IN;

process( RST,CLK,CO ) begin

    if(RST = '0') then
        COUNT_IN <= "0000";
        CO <= '0';
    elsif(RST = '1') then
        if (CLK'event and CLK = '1') then
            COUNT_IN <= COUNT_IN + '1';
            if(COUNT_IN = "1001") then
                COUNT_IN <= "0000";
                CO <= '1';
            else
                CO <= '0';
            end if;
        end if;
    end if;

end process;

Q0in <= COUNT(0);
Q1in <= COUNT(1);
Q2in <= COUNT(2);
Q3in <= COUNT(3);
CO <= CO;

end RTL;
```

シミュレーションした結果を図 2-2 に示す。

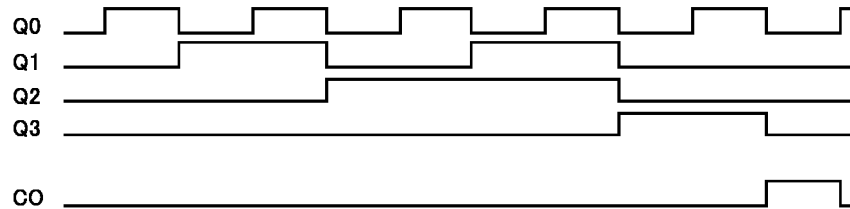


図2-2 10進カウンタ(シミュレーション)

Q0 が 2^0 、Q1 が 2^1 、Q2 が 2^2 、Q3 が 2^3 を示す。この各信号が表示素子用デコーダの入力信号となる(詳しくは 24 ページの 3 . 2 . 4 . 1 . 表示素子用デコーダで説明する)。CO が桁上げ信号です。図 2-1 では(秒)10 進桁上げ信号、(分)10 進桁上げ信号を指している。

図 2-2 の結果から、10 進カウンタとしての機能、9 から 0 に変わる時の桁上げ信号を得たことから、満足のいく結果となった。

ここでの HDL での記述は RTL(Register Transfer Level) と呼ばれ、回路設計では比較的上位の階層に位置する。この段階で動作のシミュレーションを行うことができるが、基本的にゲート遅延(信号がトランジスタを通過するのにかかる時間)は考慮できないので注意が必要です。

2 . 3 . 2 . 2 . 6進カウンタ

数を0から5まで6数える回路(D6)を以下のように記述する。入力信号として回路の動作をコントロールするCLK信号、ONにすることによって回路が動作するRST信号を使用する。また出力信号はQ0、Q1、Q2、COとする。

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity D6 is
  port (
    CLK,RST: in std_logic;
    COUNT : buffer std_logic_vector(2 downto 0);
    Q0,Q1,Q2,CO: out std_logic
  );
end D6;

architecture RTL of D6 is

  signal COUNT_IN : std_logic_vector(2 downto 0);

begin

  COUNT <= COUNT_IN;

  process( RST,CLK ) begin
    if(RST = '0') then
      COUNT_IN <= (others => '0');
      CO <= '0';

    elsif(RST = '1') then
      if (CLK'event and CLK = '1') then
        COUNT_IN <= COUNT_IN + '1';

        if(COUNT_IN = "101") then
          COUNT_IN <= "000";
          CO <= '1';
        else
          CO <= '0';
        end if;
      end if;
    end if;
  end process;

  Q0 <= COUNT(0);
  Q1 <= COUNT(1);
  Q2 <= COUNT(2);

end RTL;
```

シミュレーションした結果を図 2-3 に示す。

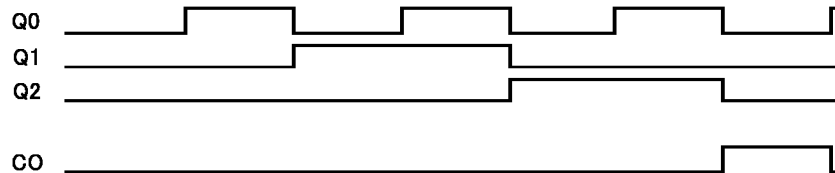


図2-3 6進カウンタ(シミュレーション)

Q0が 2^0 、Q1が 2^1 、Q2が 2^2 を示す。COが桁上げ信号です。図 2-1 では(秒)6進桁上げ信号、(分)6進桁上げ信号を指している。

図 2-3 の結果から、6進カウンタとしての機能、5から0に変わる時の桁上げ信号を得たことから、満足のいく結果となった。

2.3.2.3. 60進カウンタ

この60進カウンタ(D60)の特徴として、6進カウンタ(D6)と10進カウンタ(D10)で記述したプログラムを呼び出している。60進カウンタは10進カウンタの桁上げ(D10_CO)を6進カウンタのクロック信号とすることによって、動作を表現することができる。

数を0から59まで60数える回路(D60)を以下のように記述する。入力信号として回路の動作をコントロールするCLK信号、ONにすることによって回路が動作するRST信号を使用する。また出力信号はQ0、Q1、Q2、Q3、D10_CO、Q00、Q11、Q22、D6_COとする。

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity D60 is
  port (
    CLK,RST: in std_logic;
    Q0,Q1,Q2,Q3,Q00,Q11,Q22,CO: out std_logic);
end D60;

architecture RTL of D60 is

  component D10
  port (
    CLK,RST: in std_logic;
    Q0,Q1,Q2,Q3,CO: out std_logic);
end component;

  component D6
  port (
    CLK,RST: in std_logic;
    Q0,Q1,Q2,CO: out std_logic);
end component;

  signal D10_CO,D6_CO : std_logic;

begin
  CO <= D6_CO;

  D0 : D10 port map (CLK,RST,Q0,Q1,Q2,Q3,D10_CO);
  D1 : D6 port map (D10_CO,RST,Q00,Q11,Q22,D6_CO);

end RTL;

```

シミュレーションした結果を図 2-4 に示す。

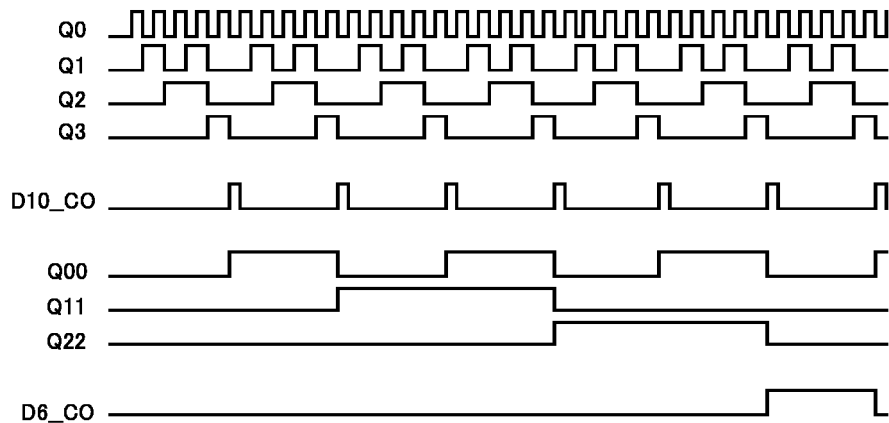


図2-4 60進カウンタ(シミュレーション)

1の位としてQ0が 2^0 、Q1が 2^1 、Q2が 2^2 、Q3が 2^3 を示す。D10_COが10の位の桁上げ信号です。10の位としてQ00が 2^0 、Q11が 2^1 、Q22が 2^2 を示す。D6_COが10の位の桁上げ信号です。

図 2-4 の結果から、60進カウンタとしての機能、59から0に変わる時の桁上げ信号を得たことから、満足のいく結果となった。

2 . 3 . 3 . 24進カウンタの機能実現

24進カウンタは図 2-1 から分かるように、12進カウンタと2進カウンタを組み合わせることによって実現することから、まず12進カウンタ、2進カウンタを記述し、各出力をシミュレーションする。

2 . 3 . 3 . 1 . 12進カウンタ

数を0から11まで12数える回路(D12)を以下のように記述する。入力信号として回路の動作をコントロールするCLK信号、ONにすることによって回路が動作するRST信号を使用する。また出力信号はQ0、Q1、Q2、Q3、COとする。

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity D12 is
  port (
    CLK,RST: in std_logic;
    COUNT: buffer std_logic_vector(3 downto 0);
    Q0,Q1,Q2,Q3,CO: out std_logic
  );
end D12;

architecture RTL of D12 is
  signal COUNT_IN :std_logic_vector(3 downto 0);
  signal Q0in,Q1in,Q2in,Q3in : std_logic;

begin

COUNT <= COUNT_IN;

process( RST,CLK,CO ) begin
  if(RST = '0') then
    COUNT_IN <= "0000";
    CO <= '0';
  elsif(RST = '1') then
    if (CLK'event and CLK = '1') then
      COUNT_IN <= COUNT_IN + '1';
      if(COUNT_IN = "1011") then
        COUNT_IN <= "0000";
        CO <= '1';
      else
        CO <= '0';
      end if;
    end if;
  end if;
end process;

Q0in <= COUNT(0);
Q1in <= COUNT(1);
Q2in <= COUNT(2);
Q3in <= COUNT(3);
CO <= CO;

end RTL;
```

シミュレーションした結果を図 2-5 に示す。

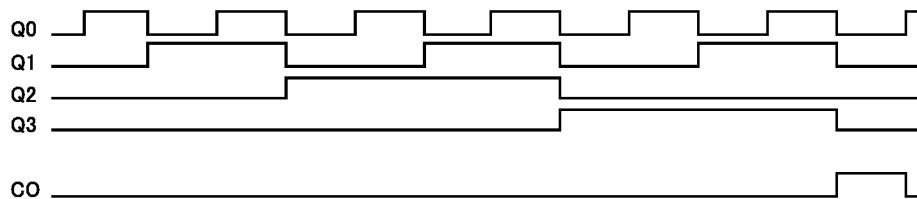


図2-5 12進カウンタ(シミュレーション)

Q0 が 2^0 、Q1 が 2^1 、Q2 が 2^2 、Q3 が 2^3 を示す。CO が桁上げ信号です。図 2-1 では (時)12 進桁上げ信号を指している。

図 2-5 の結果から、12 進カウンタとしての機能、11 から 0 に変わる時の桁上げ信号を得たことから、満足のいく結果となった。

2 . 3 . 3 . 2 . 2 進カウンタ

数を 0 から 1 まで 2 数える回路 (D2) を以下のように記述する。入力信号として回路の動作をコントロールする CLK 信号、ON にすることによって回路が動作する RST 信号を使用する。また出力信号は Q0 とする。

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity D2 is
  port (
    CLK,RST: in std_logic;
    COUNT: buffer std_logic_vector(0 downto 0);
    Q0: out std_logic
  );
end D2;

architecture RTL of D2 is

  signal COUNT_IN :std_logic_vector(0 downto 0);
  signal Q0in : std_logic;

begin

  COUNT <= COUNT_IN;

  process( RST,CLK ) begin

    if(RST = '0') then
      COUNT_IN <= "0";
    elsif(RST = '1') then
      if (CLK'event and CLK = '1') then
        COUNT_IN <= COUNT_IN + '1';
        if(COUNT_IN = "1") then
          COUNT_IN <= "0";
        end if;
      end if;
    end if;

  end process;

  Q0in <= COUNT(0);
  Q0 <= Q0in;

end RTL;

```

シミュレーションした結果を図 2-6 に示す。



図2-6 2進カウンタ(シミュレーション)

Q0 が 2^0 を示す。

図 2-6 の結果から、2 進カウンタとしての機能を得たことから、満足のいく結果となった。

2 . 3 . 3 . 3 . 24 進カウンタ

この 24 進カウンタ (D24) の特徴として、12 進カウンタ (D12) と 2 進カウンタ (D2) で記述したプログラムを呼び出している。24 進カウンタは 12 進カウンタの桁上げ (D12_CO) を 2 進カウンタのクロック信号とすることによって、動作を表現することができる。

数を 0 から 23 まで 24 数える回路 (D24) を以下のように記述する。入力信号として回路の動作をコントロールする CLK 信号、ON にすることによって回路が動作する RST 信号を使用する。また出力信号は Q0、Q1、Q2、Q3、D12_CO、Q00 とする。


```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity D24 is
  port (
    CLK,RST: in std_logic;
    Q0,Q1,Q2,Q3,Q00 : out std_logic);
end D24;

architecture RTL of D24 is

  component D12
  port (
    CLK,RST: in std_logic;
    Q0,Q1,Q2,Q3,CO : out std_logic);
  end component;

  component D2
  port (
    CLK,RST: in std_logic;
    Q0 : out std_logic);
  end component;

  signal D12_CO : std_logic;

begin

  D0 : D12 port map (CLK,RST,Q0,Q1,Q2,Q3,D12_CO);
  D1 : D2 port map (D12_CO,RST,Q00);

end RTL;

```

シミュレーションした結果を図 2-7 に示す。

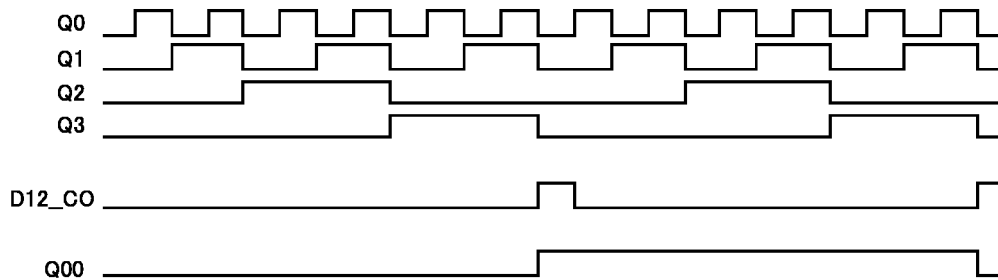


図2-7 24進カウンタ(シミュレーション)

1 の位として Q0 が 2^0 、Q1 が 2^1 、Q2 が 2^2 、Q3 が 2^3 を示す。D12_CO が 1 の位の桁上げ信号です。10 の位として Q00 が 2^0 を示す。

図 2-7 の結果から、24 進カウンタとしての機能を得たことから、満足のいく結果となった。

2 . 3 . 4 . 時計の機能実現

ディジタル時計では、発振器のクロック信号を (秒)10 進カウンタの入力信号とし、秒の 60 カウントの桁上げ信号 (図 2-1 の (秒)6 進桁上げ信号) を (分)10 進カウンタの入力信号とする。

そして分の 60 カウントの桁上げ信号 (図 2-1 の (分)6 進桁上げ信号) を (時)12 進カウンタの入力信号とすることによって実現できる。

この時計の特徴として、60 進カウンタ (D60) と 24 進カウンタ (D24) で記述したプログラムを呼び出している。時計は (秒)60 進カウンタの桁上げを (分)60 進カウンタのクロック信号として用いて、(分)60 進カウンタの桁上げを (時)24 進カウンタのクロック信号とすることにより、動作を表現することができる。

時計を表現する回路を以下のように記述する。入力信号として回路の動作をコントロールする CLK 信号、ON にすることによって回路が動作する RST 信号を使用する。また出力信号は Q0、Q1、Q2、Q3、Q00、Q11、Q22、(秒)D60_CO、Q_0、Q_1、Q_2、Q_3、Q_00、Q_11、Q_22、(分)D60_CO、Q000、Q111、Q222、Q333、Q444 とする。

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity TOKEI is
  port (
    CLK,RST: in std_logic;
    Q0,Q1,Q2,Q3,Q00,Q11,Q22,Q_0,Q_1,Q_2,Q_3,Q_00,Q_11,Q_22,
    Q000,Q111,Q222,Q333,Q444: out std_logic);
end TOKEI;

architecture RTL of TOKEI is

  component D60
  port (
    CLK,RST: in std_logic;
    Q0,Q1,Q2,Q3,Q00,Q11,Q22,CO: out std_logic);
  end component;

  component D24
  port (
    CLK,RST: in std_logic;
    Q0,Q1,Q2,Q3,Q4: out std_logic);
  end component;

  signal D0_CO,D1_CO : std_logic;

begin

  D0 : D60 port map (CLK,RST,Q0,Q1,Q2,Q3,Q00,Q11,Q22,D0_CO);
  D1 : D60 port map (D0_CO,RST,Q_0,Q_1,Q_2,Q_3,Q_00,Q_11,Q_22,D1_CO);
  D2 : D24 port map (D1_CO,RST,Q000,Q111,Q222,Q333,Q444);

end RTL;
```

シミュレーションした結果を図 2-8 に示す。結果は見やすいように秒、分、時とまとめた。

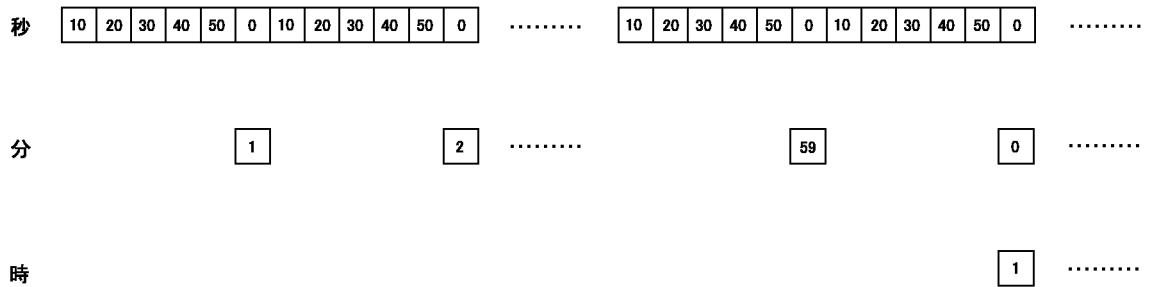


図2-8 時計シミュレーション

秒の 60 カウントと同時に発生する秒の桁上げ信号により、分のカウントがはじまる。そして分の 60 カウントと同時に発生する分の桁上げ信号により、時のカウントがはじまる。

図 2-8 の結果から、時計としての機能を得たことから、満足のいく結果となった。

以上の回路シミュレーション (VHDL) から、時計としての満足のいく出力を得ることができた。このことから市販されている標準 IC を用いての詳細設計を行う。

3 . 部分回路の詳細設計

3 . 1 . 標準 IC を用いて設計するための基礎知識

3 . 1 . 1 . IC パッケージ

IC は通常、パッケージに収まっている。幾つかあるが、デュアルインラインパッケージ (DIP dual inline package) (図 3-1) を最も多く見かける。その他小型にする目的で作られたスモールアウトラインパッケージ (SOP small outline package) (図 3-2) のようなタイプもある。

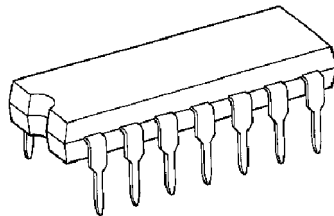


図3-1 DIP

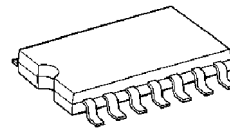


図3-2 SOP

パッケージの材質はプラスチックかセラミックです。セラミックの場合は高価であるが、より厳しい環境でも使用が可能である。

デジタル時計全体の大きさを抑えるという意味で同じ性能なら SOP を使いたいところだが、今回はプリント基板を用いるということで DIP を使用することにする。

3 . 1 . 2 . IC の性能

IC 一つを取り上げてもその性能は様々で、扱いやすい複合ゲートされた IC もある。小型化するにはうってつけだが、できるだけゲートレベルから製品を作りたいと思う。

どんな集積化された IC も結局は全てゲートレベルから構築されている。その意味をよく知り、考えるためにも図 3-3 のような基本的な IC から製作を行うことにする。

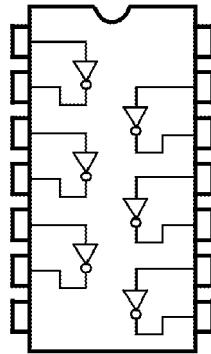


図3-3 ゲートレベルのIC例

また電気製品の頭脳として働いているマイコン(マイクロコンピュータ)(図3-4)などがある。

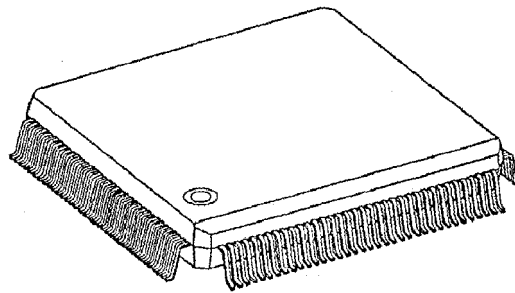


図3-4 マイコン(例)

3 . 1 . 3 . どのICを使用するか

3 . 1 . 3 . 1 . 主とするIC

3 . 1 . 2 . ICの性能でも多少触れたが、今回の製作はできるだけゲートレベルのICから製作したいと思った。

しかしただ漠然に AND,OR,NOT などの IC を使うのではなく、できるだけ NAND の IC にこだわりたいと考えている。

その理由として、

- * 1 . 同じ種類の IC を使うことによって、可能な限り小型化する。
- * 2 . AND,OR,NOT などの組み合わせを NAND のみで表現させる技術を習得する。

以上のことより、おもに使用するIC(図3-5)として、東芝製 TC74HC00 , 74HC10 , 74HC20 を使用することにした。

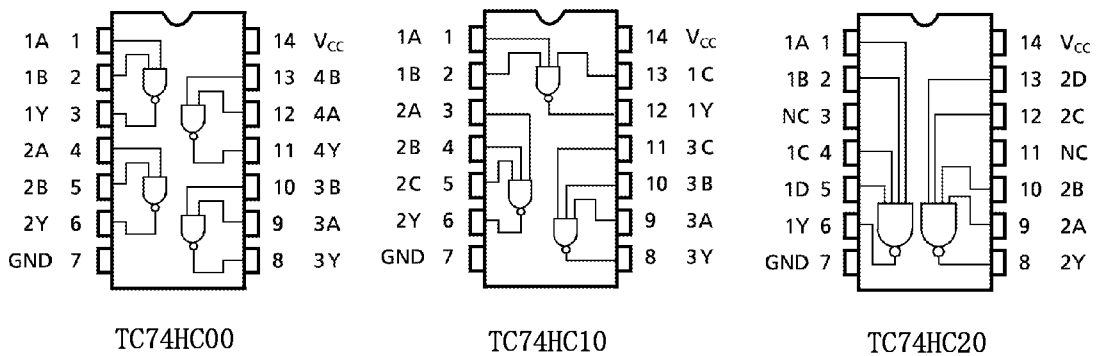


図3-5 主とするIC

3 . 1 . 3 . 2 . フリップフロップ
 3 . 1 . 3 . 2 . 1 . ラッチの基本

デジタル時計は今の時間を記憶 (保持) しなければ次の時間へと続けることができない。そこで記憶する回路としてラッチ (信号を一時的に保持する回路) またはフリップフロップの基本構成を知る必要がある。

基本的なラッチの一例が図 3-6 である。

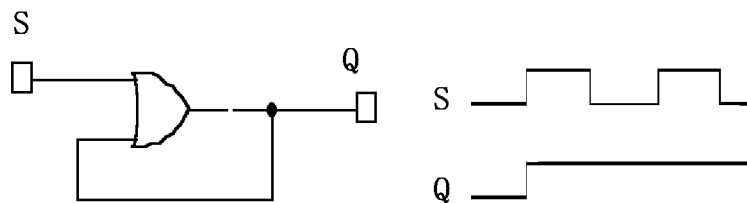


図3-6 ラッチの基本

OR ゲートの出力が入力に帰還しており、入力 S (セット) がいったん H (ハイ) レベルになると出力 Q が H になり、それが入力に戻ってくるのでその後入力 S が L (ロウ) レベルに戻っても、出力は H レベルを保持し続ける。つまり入力を記憶するということになる。

以上が基本的なラッチの動作です。ただ図 3-6 のような回路は、一度出力 Q が H レベルになると、2 度と L レベルにはなってくれない。これでは使い物にならない。

そこで、この回路にさらに R (リセット) のゲートを付け加えて、図 3-7 のように出力を L にも H にもできるようにする。これを RS フリップフロップという。

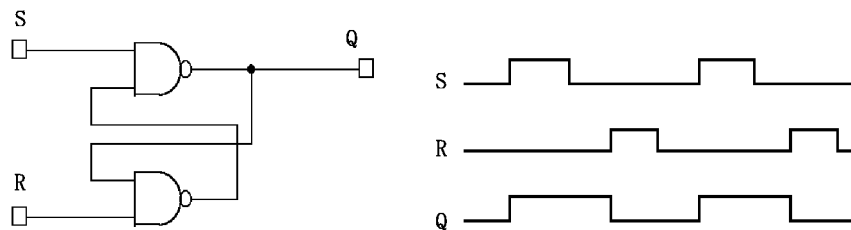


図3-7 RSフリップフロップ

3 . 1 . 3 . 2 . 2 . Dフリップフロップ

図 2-7 を発展させたものとして、RS フリップフロップの前にデータ記憶用ゲートを 2 個追加させた回路を D フリップフロップ (図 3-8) と呼ぶ。

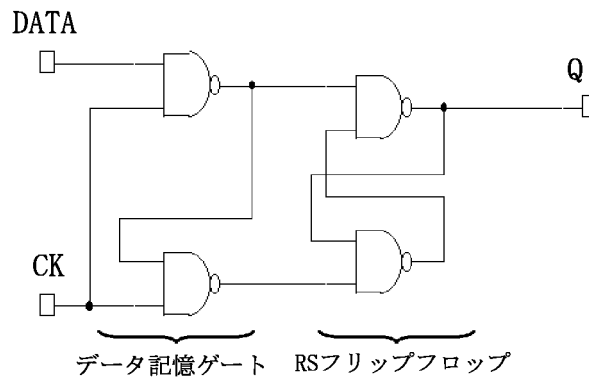


図3-8 Dフリップフロップ

CK (クロック) はコントロール信号とも呼ばれ、この回路はクロックの H、L によってデータを RS フリップフロップに記憶するかどうかを決めるようになっている。

D フリップフロップの動作を図 3-9 に示す。

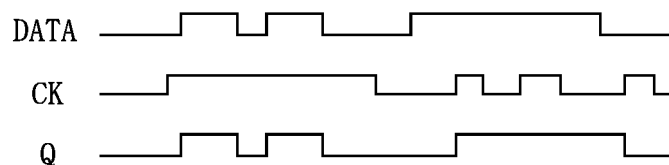


図3-9 Dフリップフロップの動作

D フリップフロップの回路構成は図 3-8 のようになるので図 3-5 で紹介した IC で表現することができるが、可能な限り回路全体の大きさを縮小したいので TC74HC74 (図 3-10) を使用する。

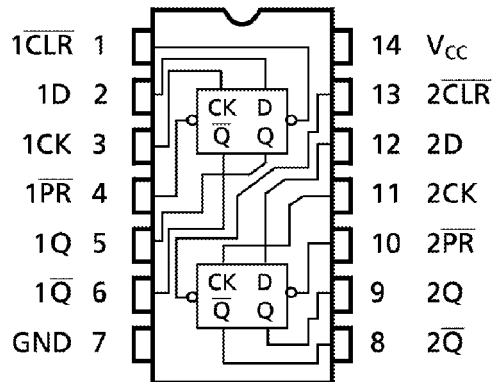


図3-10 TC74HC74

以上より、使用する IC 等、必要部品をまとめる。

・使用する IC

プリント基板を使用するので IC パッケージは DIP 型 (東芝製ではその表現として IC 名の最後に AP と表示)。おもに使用する IC は東芝製 TC74HC00AP, 74HC10AP, 74HC20AP, 74HC74AP。また必要に応じて IC を使用する。

・その他

セグメント表示器、水晶発振器、抵抗等も必要に応じて使用する。

使う部品が決まったので、それら部品を用いた詳細設計をする。部品をどう配置すればよいか、まずシミュレーションをして、それから回路製作を行うことにする。

3 . 2 . Multisim による詳細シミュレーション

3 . 2 . 1 . Multisim とは

2 . 3 . 回路シミュレーション (VHDL) では最も重要な動作である 60 進カウンタ、24 進カウンタのみの動作シミュレーションを行った。

今度はその動作をゲートレベル (NAND 等) で設計すると共に、発振器、表示素子用のデコーダ、24 進デコーダ、スイッチ等のサブ回路等も設計する。

その作業を行うソフトの一例が Multisim である。

3 . 2 . 2 . カウンタ

3 . 2 . 2 . 1 . 60 進カウンタ

数を 0 から 59 まで 60 数える回路を以下のようなゲートレベルで表現できる (図 3-11)。回路の中で表される Subcircuit の中身は図 3-12 で示す。

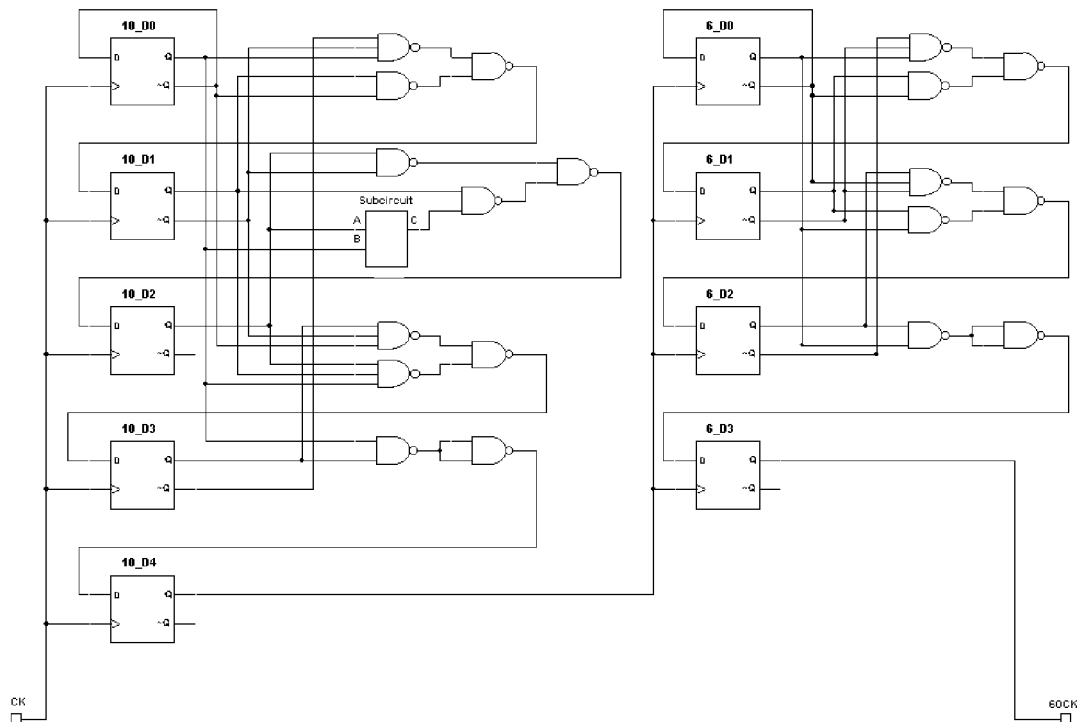


図3-11 60進カウンタ (Multisim)

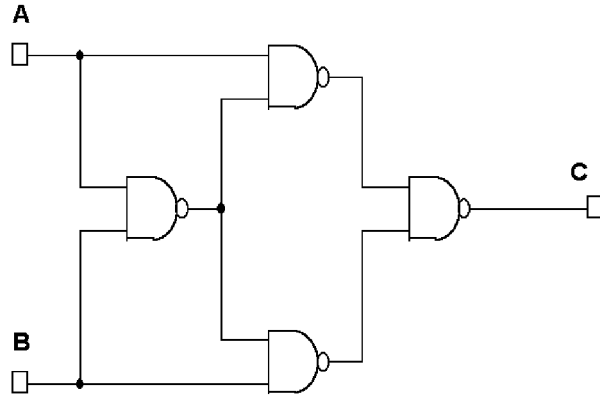


図3-12 Subcircuitの中身

動作は図 3-13 に示す。

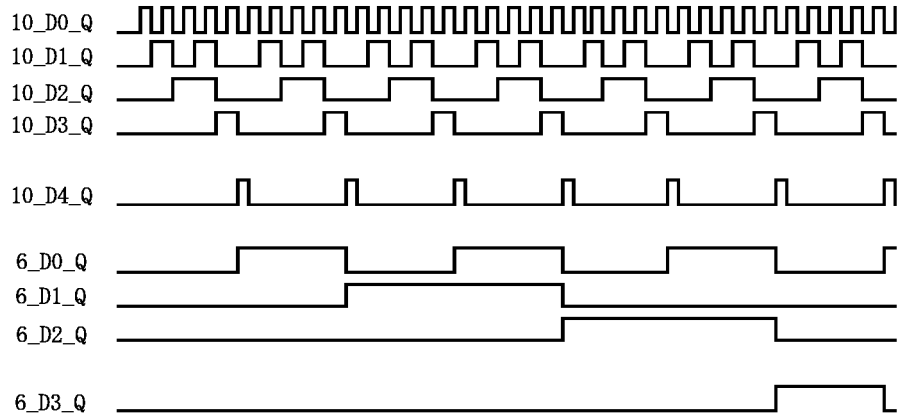


図3-13 60進カウンタ (動作)

3 . 2 . 2 . 2 . 24進カウンタ

数を0から23まで24数える回路を以下のようなゲートレベルで表現できる (図3-14)。

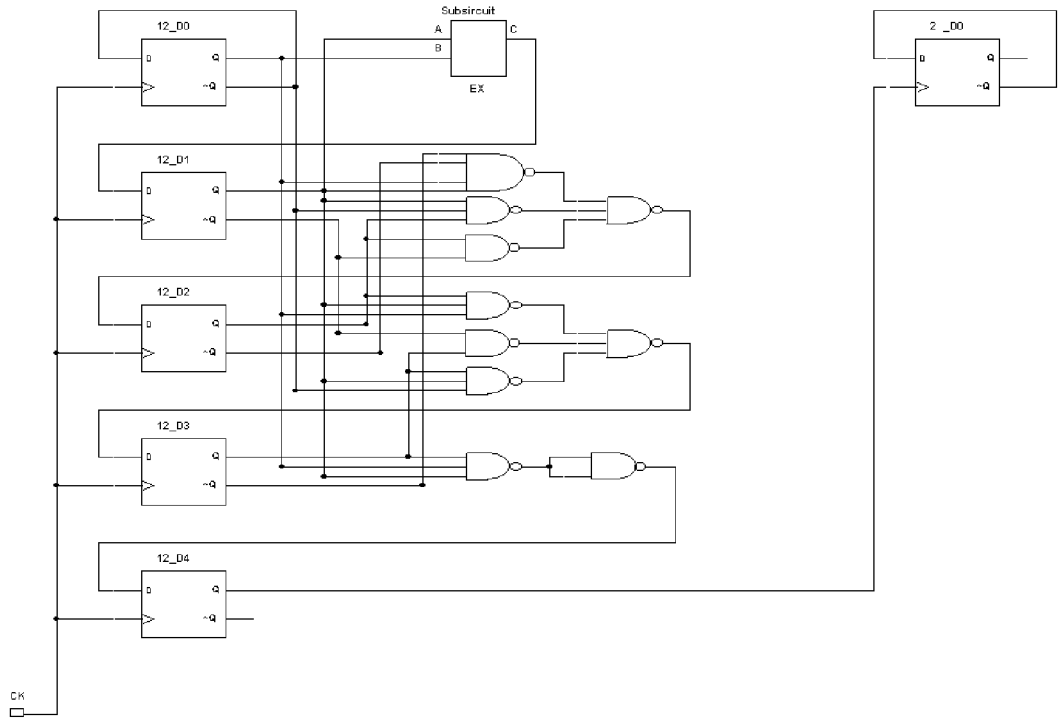


図3-14 24進カウンタ (Multisim)

動作は図 3-13 に示す。

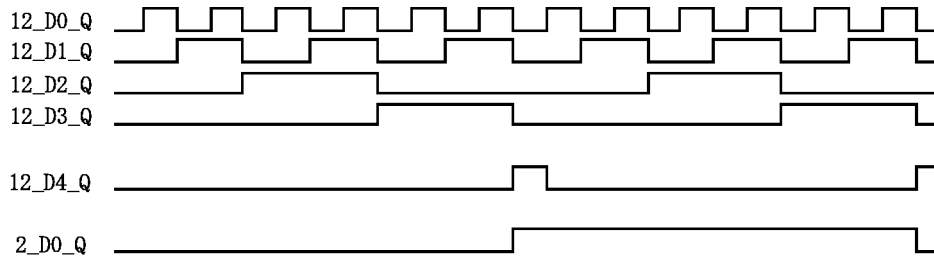


図3-15 24進カウンタ (動作)

3 . 2 . 3 . LED 数字表示素子

回路でカウントした結果の2進数を10進数表示するには、LED 数字表示素子 TLR336S (図 3-16) を使用することにした。

7個の発光ダイオードが8の字状に配置されている。それぞれ a ~ g までのセグメントとして記号がつけられており、ここに電流を流すことによって発光表示することになっている。

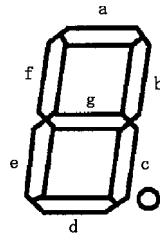


図3-16 TLR336S

3 . 2 . 4 . デコーダ

3 . 2 . 4 . 1 . 表示素子用デコーダ

図 3-16 を使用するために、どのような状態でどのように光らなければならないか指定しないとイケない。真理値表を表 3-1 に示す。

表3-1 入力と出力の関係

入力				出力						
D3_Q	D2_Q	D1_Q	D0_Q	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	0	0	1	1

これを実現させた回路が図 3-17 である。

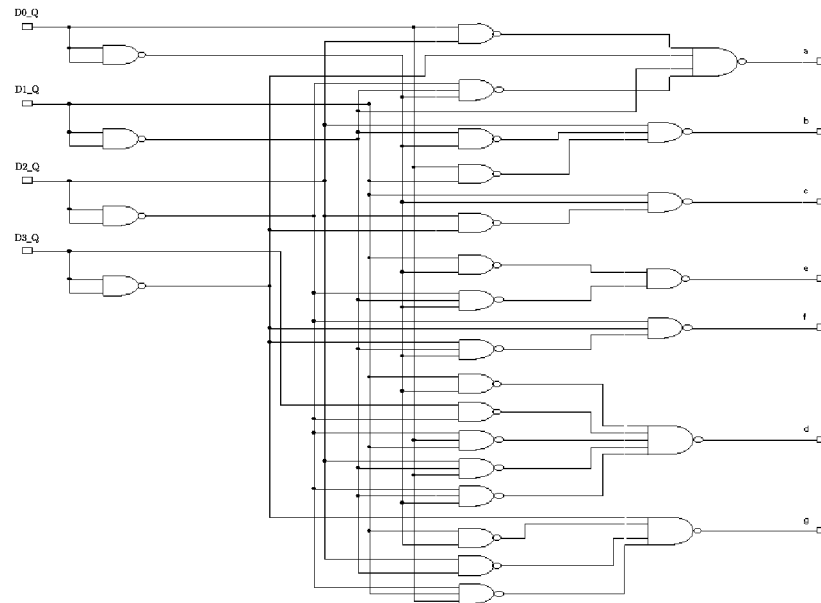


図3-17 LED表示用回路

秒、分、時を人が見てわかるように表示するので、例えば秒では60カウントまでであるので1の位用と10の位用の2つの表示素子と表示用回路が必要である(図2-1参照)。同様に分、時も2つずつ必要だから、全体では計6つの表示素子と表示用回路が必要となる(図2-1参照)。これを図3-17のまま実現すれば明らかに回路全体が大きくなる。

そこでこの部分の回路に市販されている4543BP(図3-18)を使用することにした。

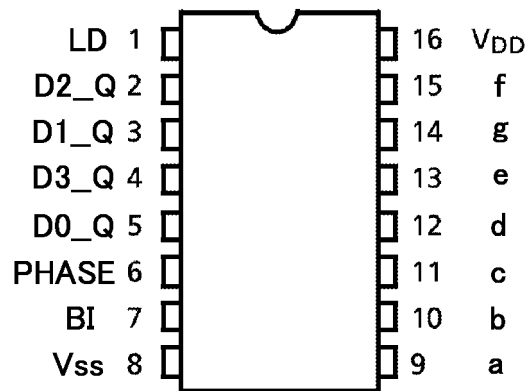


図3-18 TC4543BP

3 . 2 . 4 . 2 . 24進デコーダ

60進カウンタは、10進カウンタの桁上げ信号を6進カウンタのクロック信号とすることで動作する。この動作は表示素子用デコーダにとって大変都合がよい。

10進数で0~59をカウントする動作は、「1の位が10カウントすると、桁上げにより10の位が動作する、それを59まで行う」。この動作は60進カウンタの「1の位の10進カウンタが10カウントすると桁上げ信号により、10の位の6進カウンタが動作する、それを59まで行う」と同じことである。

つまり、60進カウンタの動作はそのまま表示素子用デコーダを使うことにより、表示素子を10進数で表示することが可能である。

しかし、12進カウンタの桁上げ信号を2進カウンタのクロック信号とすることで動作する24進カウンタはどうであろう。この動作は「1の位の12進カウンタが12カウントすると桁上げ信号により10の位の2進カウンタが動作する、それを23まで行う」ということである。表示素子用デコーダは0~9までのカウントしかないので、1の位がうまくカウントできない。

そこで、24進カウンタの動作を「1の位の10進カウンタが10カウントすると桁上げ信号により10の位の3進カウンタが動作する、それを23まで行う」。そういった信号に変換する回路を表示用デコーダの前に作る必要がある(図2-1参照)。それを実現した回路が図3-19である。

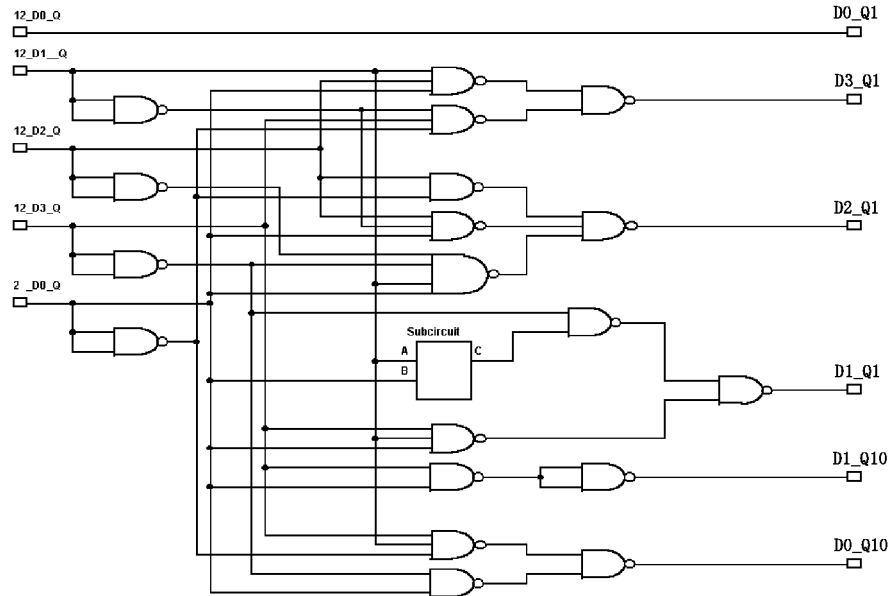


図3-19 24進デコーダ

図3-19を使用することで、図3-15の動作を図3-20のようになる。

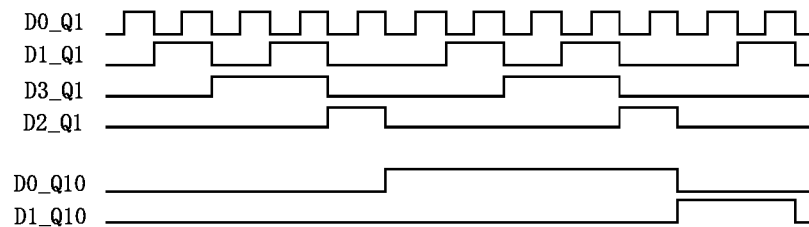


図3-20 24進デコーダの動作

3 . 2 . 5 . 発振器

3 . 2 . 5 . 1 . 発振器とは

時を表示させる回路には、分からの桁上げ信号をクロックとして用いる。同様に分の場合は秒からの桁上げ信号を用いる。

では秒はどこからの信号を用いればよいか。その基本となる短いパルスを発振する回路を作らなければならない。

3 . 2 . 5 . 2 . 水晶発振器

基本となる信号は正確でなくてはならない。とくに今回のデジタル時計は正確さを大事にしたい。それを可能としたのが、水晶（クリスタル）を用いた発振器である。なんととっても周波数の安定度がその特徴である。

3 . 2 . 5 . 3 . 分周器

さて、水晶発振を用いて秒の基準となる 1Hz のクロック信号を使用したいところであるが、水晶発振器は非常に高い周波数で発振している。それを自分の利用したい周波数まで落とすことを分周という。

分周とはフリップフロップなどの技術を用いて周波数を $\frac{1}{2}$ にすることなので、うまく分周後の周波数が 1Hz になるような水晶を選ばなければいけない。そこでちょうど 2 の累乗に等しい周波数発振する水晶を選ぶので

- 3.2768kHz
- 4.194304MHz

の 2 つが実際に市販されている。前者なら 15 回、後者なら 22 回分周すれば 1Hz になる。水晶の発振周波数が高いほど、1Hz は正確になるので、ここでは 4.194304MHz を使うことにした。

しかし、いくら何でもフリップフロップを 22 個、つまり 74HC74AP を 11 個使うのは効率がよくないので、ここでは 4521BP (図 3-21) を使うことにした。

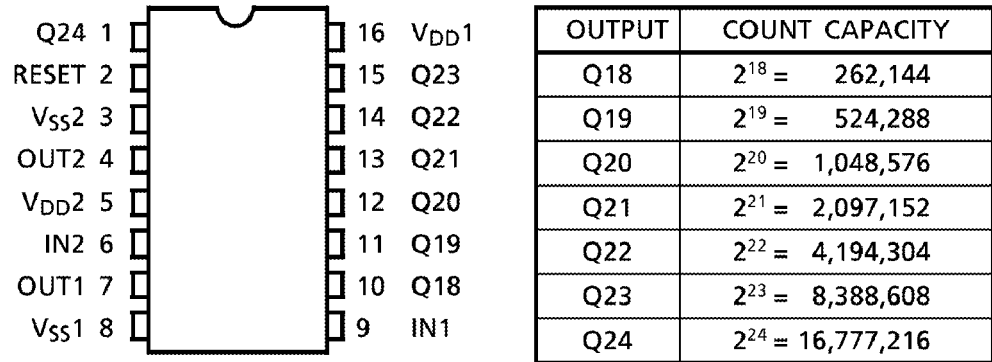


図3-21 4521BPの仕様

4521BP の出力 Q22 を用いることによって 1Hz の波形を得ることが可能となる。

ところで、私の作りたいデジタル時計は初期設定を可能としている。初期設定を決定するときは 1Hz だと少し遅く感じるので、出力 Q20 を用いることにより、4Hz を得ることができる。この波形を初期設定用の周波数として利用することにした。

3 . 2 . 6 . スイッチ

初期設定用にスイッチ動作をシミュレーションしたかったが、私の知る限りでは Multisim にはいくつかの簡単なスイッチしかなかった。不本意だが、ここではシミュレーションせず、実際にできた回路から組み立てることにする。

4 . 回路製作

4 . 1 . 回路製作時の注意点

3 . 詳細設計で考えた回路を実際に製作する。注意点を以下にまとめます。

1 . 手に入れやすいICが事前にチェックしておく

今回のデジタル時計で使用したICは全て東芝製で東芝セミコンダクター社 (<http://www.semicon.toshiba.co.jp/>) でそのIC情報を入手することが可能である。ただ、HPに載ってるからといって必ずしもそのICが市場に出回ってるわけではない。中には入手困難なICもある。

今回は2つのICを得るのに苦労した。TC4543BPとTC4521BPである。前者は全部で6つ必要だが、あまり在庫が無いとのこと。しかも製造中止だったので入手に苦労した。そして後者にいたっては四国では入手困難だったのでホームページで関東の業者から在庫を探しあてた。

部品は先に入手してから、回路設計を行う。それが後々に良いとわかった。

2 . 部品を知る

3 . 詳細設計を行った結果、全ての回路はNANDゲートから構築できることが分かった。だからといって図2-5に示す基本的なICから全てを構築していくと、それはとてつもなく大きな回路になってしまう。

どんな部品が市場に出回り、どんな部品が存在するのか、それらを把握することによって回路全体の大きさも変わる。

またICだけではなく、スイッチ、押しボタンの種類、基板等も知ることも大事である。そうすることにより全体のレイアウトを描くことも可能である。

例えば、スイッチと一言でこれだけの種類がある。

- ・トグルスイッチ … レバーを操作してON/OFFする、工作機械等でよく見られるスイッチである。丸穴を開けるだけでケースに取り付けられるので重宝するが、やや高価。接点数の違いや、ハネ返りの有無で、さまざまな種類がある。

- ・スライドスイッチ … いわゆるスライドスイッチである。割と安価だが、角穴を開けないと取り付けられないのが面倒。基板に直接取り付けられる小型タイプもある。
- ・波動スイッチ … 部屋の照明スイッチ等によく見られる、シーソーのようなスイッチ。安価で見栄えも良いが、取り付けは面倒である。LED内蔵型もある。
- ・プッシュスイッチ … これもいわゆるプッシュスイッチである。押すと ON になる一般的なもののほかに、OFF になるのや、ON でロックされ、もう一度押すと OFF に戻るオルタネート型もある。
- ・マイクロスイッチ … ロボットの壁面検出等によく使われるスイッチである。レバーが針金等でできていて長いので、軽い力で ON/OFF できるのが特徴です。
- ・ディップスイッチ … DIP 型の IC と同じ大きさでできているスイッチである。デジタル回路で使いやすいように、何個ものスイッチがある。これと似たものにピアノスイッチや、シーソースイッチがある。
- ・タクトスイッチ … 基板取り付け専用のスイッチである。構造はプッシュスイッチと同じだが、クリック感があり、キーボードのスイッチ等に使われる。
- ・ロータリースwitch … ボリュームと同じような形をし、回転させることで接点の ON/OFF する。

今回はトグルスイッチを使った。使用したスイッチの特徴は 37 ページの図 4-4 に示す。

4 . 2 . 回路設計と回路製作の違い

4 . 2 . 1 . IC ソケット

IC は図 3-1 の DIP 型を用いたが、そのままハンダ等の作業を行うとピンが剥き出しのため、ハンダの熱で IC の性能が狂ってしまう可能性がある。それを防ぐのが IC ソケットです。

最初はソケットのみをハンダごてし、そのソケットの中に IC を挿入する。

注意点として、各種 IC によってピンの数が若干異なる（図 3-5 の IC のピンが全部で 14 ピンに対し、図 3-18 は 16 ピン）。その IC のピンにあった、IC ソケットを用いるようにする。

4 . 2 . 2 . 表示素子までの配線

4 . 2 . 2 . 1 . 実際に接続した配線

回路設計の段階では単純に表 3-1 からの信号を参考にし、その動作を行う図 3-18 の IC を使えばよいが、その信号の電圧、電流等を検討しなくてはいけない。今回のデジタル時計は 5V で動作すると最初の段階で決めたので、それに見合った配線を考えなければならない。

この場合、5V の電圧を加え、この表示素子の標準電流は 10mA なので

$$5.00 = R \times 10 \times 10^{-3}$$

$$R = \frac{5}{10 \times 10^{-3}} = 500$$

つまり 500 Ω の抵抗を表示素子の入力部分の前に入れる必要がある。実際は 500 Ω がなかったので近い値の 470 Ω を使用した。

論理上で配線を考えるのが回路設計。その論理と IC の特性、表示素子の特徴等の物理的な調和を考えた上での配線方法が回路製作とわかる。

4 . 2 . 2 . 2 . つなぐべき配線

実際に接続した抵抗値等は接続した後に間違いと分かった。ここでは本当になくべき抵抗値をもう一度計算しなおす。

全体に5Vの電圧を加え、このLED数字表示素子には順方向電圧として標準電圧2Vかかる。標準電流は10mA。なので表示素子の前の抵抗に $5-2=3$ Vの電圧がかかることになる。つまり、

$$3 = R \times 10 \times 10^{-3}$$

$$R = \frac{3}{10 \times 10^{-3}} = 300$$

300Ωの抵抗を表示素子の入力部分の前に入れる必要性があった。実際に使用した470Ωは少し高すぎたことになる。実際に470Ωを使用した場合、

$$3 = 470 \times I$$

$$I = \frac{3}{470} = 6.38\text{mA}$$

となり標準電流よりも少なくなることがわかった。

4 . 2 . 3 . 入力信号の注意

4 . 2 . 3 . 1 . ノイズ対策

また入力信号にも注意が必要である。CMOS回路の入力は外部からのノイズ等に非常に敏感で、ノイズで回路が不安定になってしまうことも十分考えられる。これを防ぐのがコンデンサである。コンデンサの一方には入力部分、そしてもう一方はグランドに接続する(図4-1)。

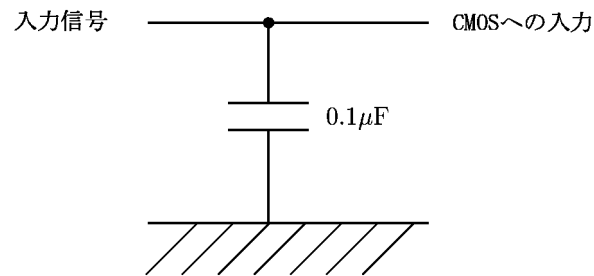


図4-1 ノイズ対策

図4-1では $0.1\mu\text{F}$ を使用している。もちろん別に $1\mu\text{F}$ や $10\mu\text{F}$ でも問題はない。

ただ、使うコンデンサは高周波ノイズ対策のためにも周波数特性の良いものを使うので、電解コンデンサは使えない。またあまり容量が小さいと逆に電気をためてノイズをなまらせる効果が薄くなるので、その中間点が $0.1\mu\text{F}$ ぐらいがベストではないかと思う。

今回のデジタル時計回路の入力部分には $0.1\mu\text{F}$ のコンデンサを使用することにした。

4 . 2 . 3 . 2 . HとLを明確にするには

シミュレーションの場合はH信号、L信号とハッキリと明確に表すことができるが、現実の場合にはその信号がどちらかハッキリ分からない状態にならないと限らない。

そこで、図4-2のような回路を配置する。

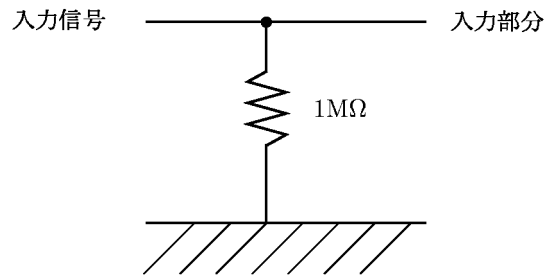


図4-2 入力信号を明確にした回路図

入力信号が H 信号と L 信号のそれぞれの状態を検証する (図 4-3)。

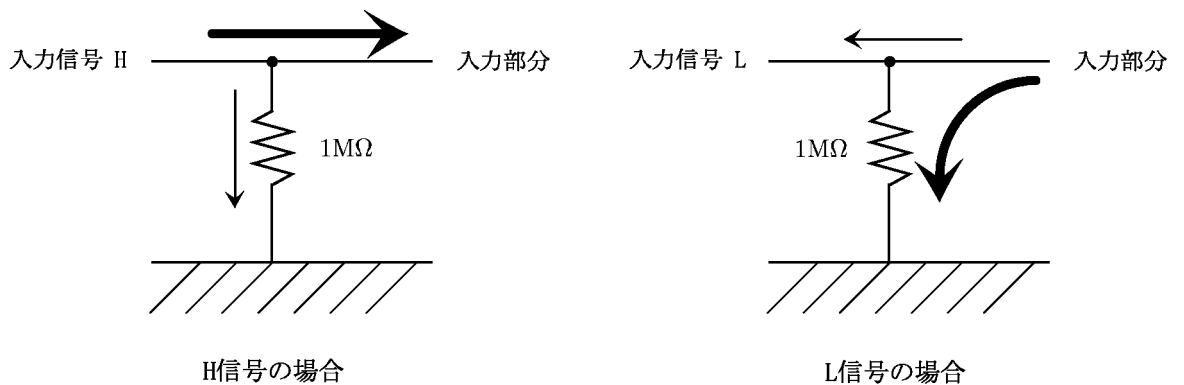


図4-3 HとLの入力状態

それぞれの矢印の向きは電流量を表している。H 信号の場合、電流は流れやすいほうへ流れる性質があるので抵抗へはあまり流れず、入力部分に多くの電流が流れ込む。

一方 L 信号の場合は、入力信号がハッキリしないような L レベルだとしても、抵抗の先がグランドに接続されているから、必然的に入力部分には L レベルになる。

4.3. スイッチ

4.3.1. スイッチ配置

初期設定実現のため、トグルスイッチを用いた。実際に用いたスイッチの簡単な動作を図4-4に示す。

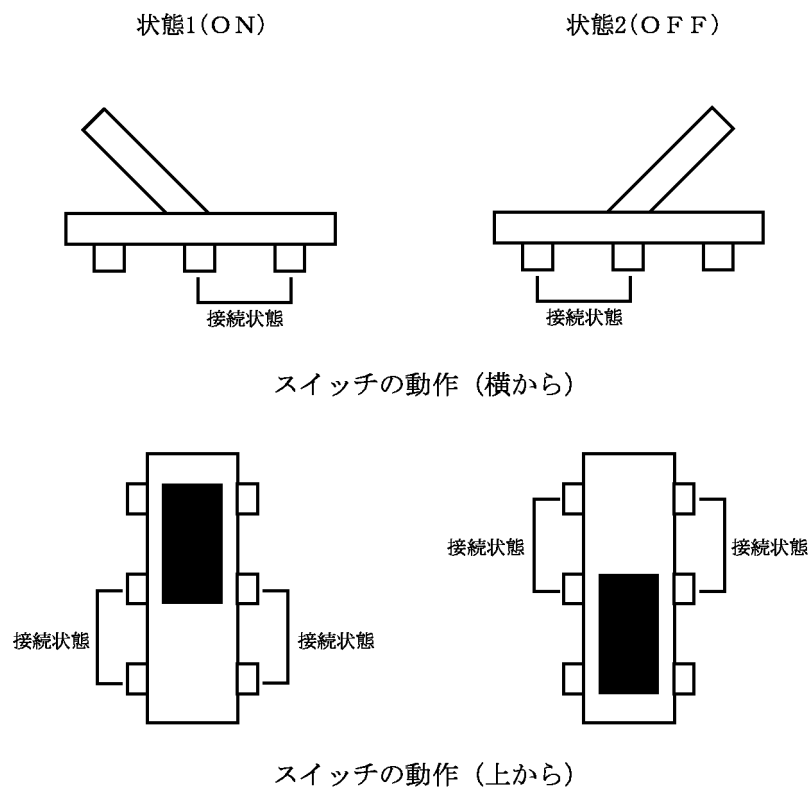


図4-4 スイッチの基本動作

図4-4のスイッチを使って、初期設定を可能にした回路図を図4-5に示す。

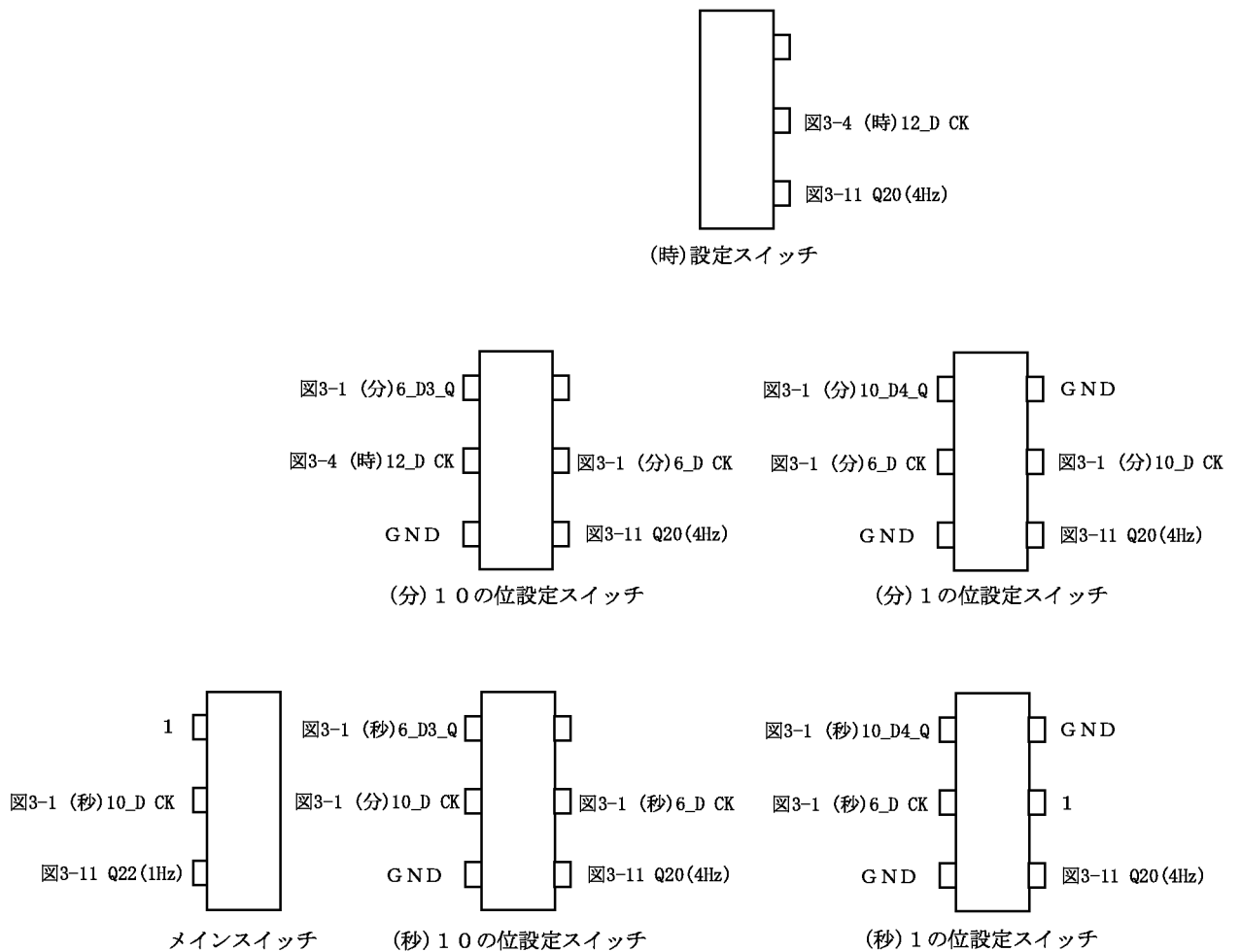


図4-5 初期設定スイッチ配置図(上から)

図4-5の1の表示はそこが接続されていることを示す。スイッチには何も入力されていない所がいくつかあるが、この場合は特に回路に影響は与えない。詳しい配置は図2-1と見比べればよく分かる。

スイッチそのものの初期状態として、メインスイッチがON(図4-4の状態1の時)の状態以外、他のスイッチは全てOFF(図4-4の状態2の時)になっている。

この状態では(秒)10_D CKにTC4521BPの出力、Q22からの1Hzが入り、通常どおり、デジタル時計としての役割を果たしている。

初期設定をしたい場合は、メインスイッチをOFFにする。そうすると、(秒)10_D CKをGNDに接続することによって、表示素子の今表示されている数字を止めることができる。そこで、表示素子の数字を変更したい所のスイッチをONにすれば、そこだけを変更する事が可能となる。

例えば、分を表している表示素子の10の位を変えたいのであれば、(分)10の位設定スイッチのみをONにすることによって、初期値を変更することができる。

この利点としては、利用者が時計の時間を合わせる時に、手間なく、すぐに時計としての利用価値を得ることができる。

4 . 3 . 2 . スイッチ利用の注意点

スイッチを利用するにあたり、注意すべき点がある。1回しかスイッチを切り返してないのに、数字がいくつもカウントされ、でたらめな出力になったりすることがある。

これは、スイッチの接点は押してから完全につながるまで短い周期でON/OFFを繰り返すからである。この現象をチャタリングという。

この現象を防ぐのが、シュミット・トリガーです。シュミット・トリガーはいったんHからL(もしくはLからH)になったとき、再びH(L)に戻すには、いま変化した電圧以上の電圧を加えないと元の状態に戻らないようになっている。つまりはノイズに強いということだ。だからスイッチが押されて一旦Lになると、Hには戻りにくいので好都合というわけである。

TC74HC14AP(図4-6)がシュミット・トリガーの役割を果たす回路なので、スイッチの部分に利用することにする。

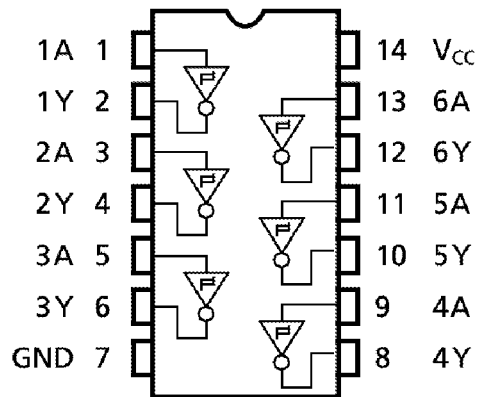


図4-6 TC74HC14AP

製作した回路を上から撮影したのを図 4-7 に示す。

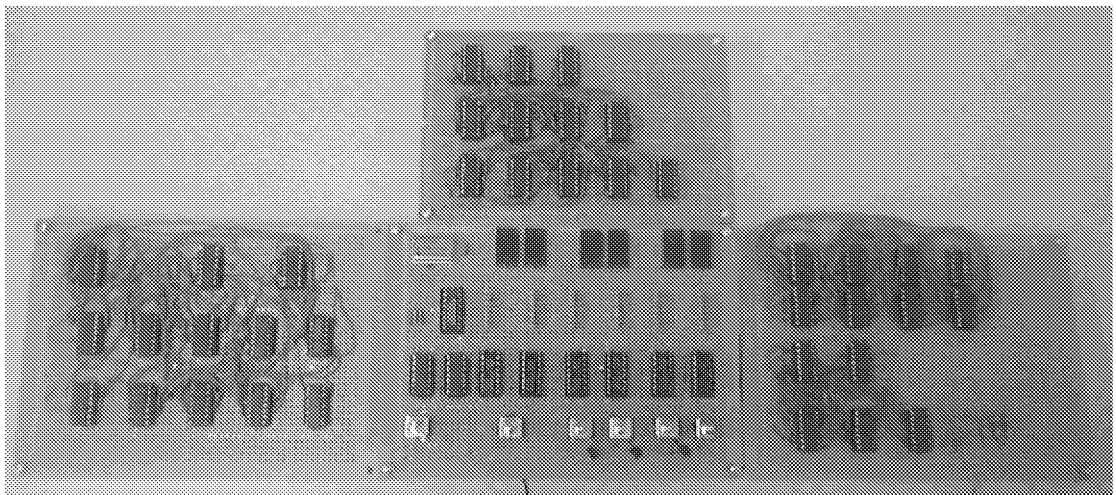


図4-7 初期設定可能デジタル時計（上から）

製作した回路を下から撮影したのを図 4-8 に示す。

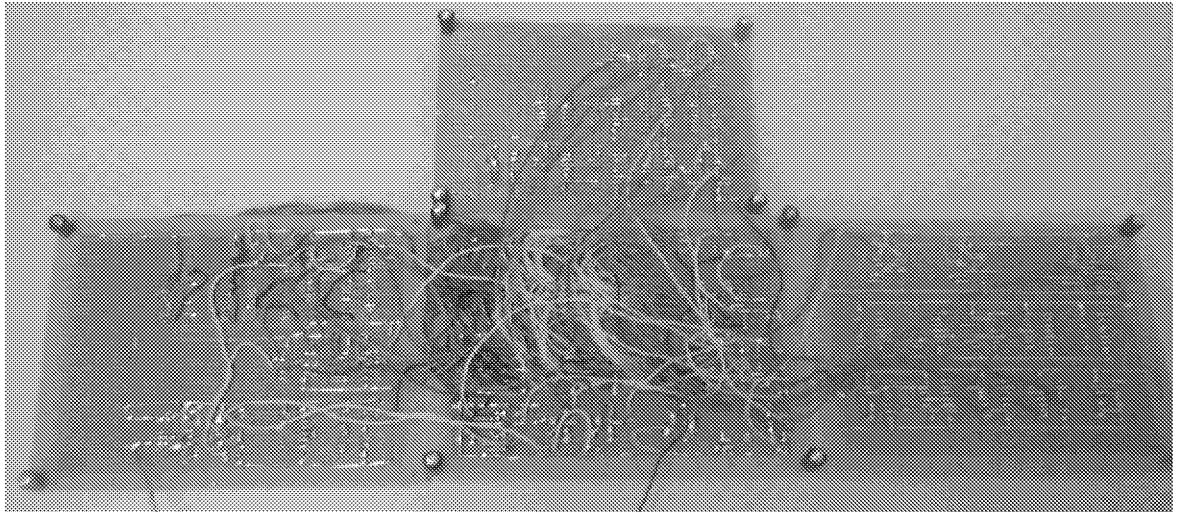


図4-8 初期設定可能デジタル時計（下から）

表示素子がある所は上からの見た目を綺麗にしたかったので、配線等を極力下に持っていった結果、下から見るとその部分の配線が相当汚くなってしまった。次にやる機会があればこの反省も含め、もう少しその部分の配線にも気を使いたい。

5 . 評価

5 . 1 . 機能確認

電源を入れて、機能を確認した。まず、初期設定からできることを確認してから、5日間の動作を見た。その間、秒、分、時を正しくカウントしていることを確認できるため、時計としての機能をはたしていることを確認できた。

5 . 2 . 時計としての精度

メインスイッチを OFF の状態から、現在の時間に初期状態を設定し、5日間放置して測定したところ、1秒以上の誤差はなかった。

$$5 \text{ 日} = 60 \times 60 \times 24 \times 5 = 432000 \text{ 秒}$$

$$\frac{1}{432000} = \frac{1}{4.32 \times 10^5} = 2.3 \times 10^{-6}$$

1秒の誤差が 2.3×10^{-6} 秒以下の精度があると確認できた。

5 . 3 . 電源電圧

最初の段階で動作は 5V と決めて、5V で動くデジタル回路を製作した。もちろんその通りの回路ができたので、とてもうれしく思う。

しかし今回使用した IC は 2~6V で動作すると保証されている。つまりその間の電圧であれば IC としての機能は果たすわけである。

ここで、3V で動かしてみた。時計としての精度は別として、一応時計としての機能ははたした。しかし、LED 数字表示素子で問題が起こった。

34 ページの 4 . 2 . 2 . 表示素子までの配線の中では 5V 用に抵抗を考えた。
しかしそれが 3V になったので、

$$3 - 2 = 470 \times I$$

$$I = \frac{1}{470} = 2.13 \times 10^{-3}$$

2.13mA になると、表示素子が見えにくく、使えない。
仮に 3V を使用すると最初に決めていれば、

$$3 - 2 = R \times 10 \times 10^{-3}$$

$$R = \frac{1}{10 \times 10^{-3}} = 100$$

100 Ω を使用すれば、電源電圧 3V で LED 数字表示素子は使用可能になる。

5 . 4 . 電源電流

5 . 4 . 1 . 電源の電流値

電源電圧 4.86V で電源電流を測定したところ、回路全体で 70 ~ 210mA の電流が流れた。IC 一つあたりの消費電力は微弱であるが、回路全体では大きくなる。

図 5-1 に示すように、

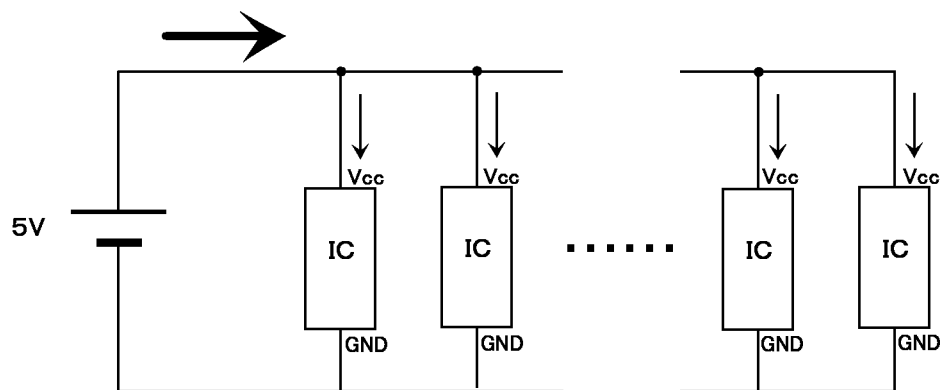


図5-1 電源電流が高い理由

IC 一つ一つに必要な電流は小さいが、数多くの IC を使う回路であればあるほど全体の電流は大きくなる。今回のデジタル時計は計 47 個の IC を使用した。それ以外にも LED 表示素子、水晶発振器等が V_{CC}、GND を共通して図 5-1 のように接続されている。

以下、なぜこのような大きな電流になるのか、またなぜ時間によって 70 ~ 210mA と変化するのか、定量的に検討してみる。

電源の電流値は表示素子の光り方によって 70 ~ 210mA と変化することが、測定結果から分かった。以下に表示素子の電流値を検討する。

5 . 4 . 2 . 表示素子の電流値

1セグメントを光らせるための入力電圧は測定したところ4.68Vであった(電源電圧は4.86Vを使用しているが配線による電圧降下か)。LED表示素子に加わっている電圧は2.3 ~ 2.4Vなので、表示素子の1セグメントに流れる電流は4.68mA ~ 5.11mA。

電源電流が一番低い表示素子の状態を図5-2に示す。

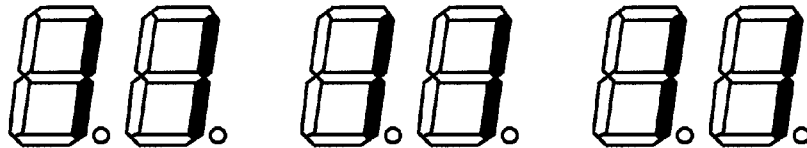


図5-2 一番電源電流が低い時の表示素子状態

この時の電源電流の値は70 ~ 90mA。全部で表示素子は12セグメントが点灯し、その一つに流れる電流は先ほどの計算結果から約5mAなので、図5-2の状態を光らせるのに必要な電流は

$$5 \times 12 = 60\text{mA}$$

となる。

1セグメントを光らせるのに必要な電流は4.68mA ~ 5.11mAのばらつきがあるのだから、12セグメントを点灯させるときに必要な電流はさらにばらつく。表示素子が止まっている状態なのに、70 ~ 90mAのばらつきがあるのはこのためと考えられる。

また、電源電流が一番高い表示素子の状態を図 5-3 に示す。



図5-3 一番電源電流が高い時の表示素子状態

この時の電源電流の値は 160 ~ 210mA。全部で表示素子は 39 セグメントが点灯しているので、図 5-3 の状態を光らせるのに必要な電流は

$$5 \times 39 = 195\text{mA}$$

となる。

点灯するセグメントの数が増えれば増えるほど、それに比例して電源の電流値が高くなっていることから、表示素子と電源電流は密接な関係があることがわかった。また、電源電流の値はほとんどが表示素子を光らせるのに必要な電流であることがわかった。

今回極力ゲートからの製作にこだわったが、最近色々な機能を一緒にしたワンチップ化というのが重要視されている。その理由が回路全体の大きさの縮小と共に、配線等による電圧降下や回路全体の消費電力の低減というのがよく理解できた。

5 . 5 . 製作初期の問題点

スイッチを ON、OFF にすることにより初期設定を可能にしたが問題があった。図 3-13 から分かるように、1 の位の桁上げ信号である 10_D4_Q の信号によって、10 の位が動作する。

つまり、初期設定の時に 10 の位の表示を 0 で止めてしまえば、桁上げ信号は H 状態を維持するので、10 の位に初期設定用の信号を送っても、桁上げ信号が邪魔をし、動作しないのである。

これを解決するには、時表示、10 の位の分表示、1 の位の分表示、10 の位の秒表示、1 の位の秒表示の流れで、初期設定を行えばよい。

また、0 で表示止めると、桁上げ信号が H 状態になることから、次の桁が 1 カウントされてしまう。それを頭に入れて、初期設定を行わなければならない。

例えば、9 時 00 分 00 秒に初期設定をあわしたいなら、まずスイッチで図 5-4 に示すような状態にする。



図5-4 初期状態を9時00分00秒にする前の状態

図 5-4 の状態から秒の 1 の位をスイッチで 0 にすると図 5-5 に示すような状態になる。

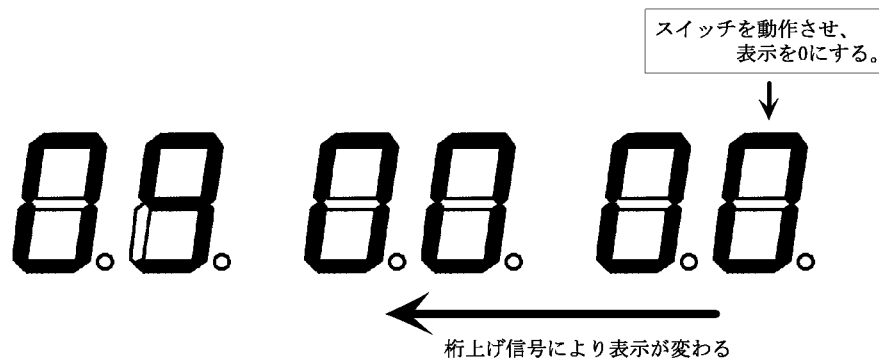


図5-5 初期状態を9時00分00秒にした状態

このような動作で初期状態を9時00分00秒にすることができる。

6 . その他

6 . 1 . 反省点

一連の作業でよく分かったのが、何を作るかの段階で全てをイメージし、まとめておくということの重要性を知った。後々で分かっても、その段階ではもう手遅れになっていることは避けられない。そうなると今まで製作したモノを白紙に戻し、もう一度最初からという二度手間になる。

またICを組み立てるのをはじめてということで仕方がない部分もあったが、もう少しIC間の間隔を縮める努力もしたかった。またICの配線、スイッチの有効な使い方等の知識も不足していたと感じた。

配線を終えた後で分かったことだが、デジタル回路というのはやはり部品の寿命等も当然あるので、後でメンテナンスをしやすい回路製作を心がけなくてはいけないと思った。

今回のデジタル時計は、衝撃などで表示素子までの配線が途絶えてしまい、LEDが消えやすくなってしまう。図4-8のような配線では切れた所だけを修復すればいいのに、修復中に別の所がおかしくなることがある。実際、そういう手間がかかってしまい、修復するたびにストレスを感じた。

後々にメンテナンスしやすい回路製作という重要性も感じた。

6 . 2 . まとめ

今回はデジタル時計という身近で簡単な回路をテーマとして選んだが、全体回路設計、VHDLの使い方、電子回路の設計、標準ICを用いてのデジタル回路の製作、そして評価の一連の流れを経験し、最近のCAD技術も習得することができた。

この卒業研究で得た知識、経験は、今後の自分の人生に関わるとても重要なものだと思っている。

7 . 謝辞

今回の研究をするにあたり、参考文献として掲げた多くの著書、文献を参照しました。これらの著者の方々に感謝すると共に、有益なご助言と多大な協力をいただいた原 央教授と、原研究室、矢野研究室、橘研究室の先輩方々に心から感謝の意を表したいです。どうもありがとうございました。

8 . 参考文献

- ・ デジタル回路設計 中村 次男 著
- ・ デジタル IC 回路の基礎 松田 勲/井原 充博 著
- ・ 論理回路の基礎 田丸 啓吉 著
- ・ やさしいデジタル回路の実験 白土 義男 著
- ・ VHDL によるハードウェア設計入門 長谷川 祐恭 著