論理回路

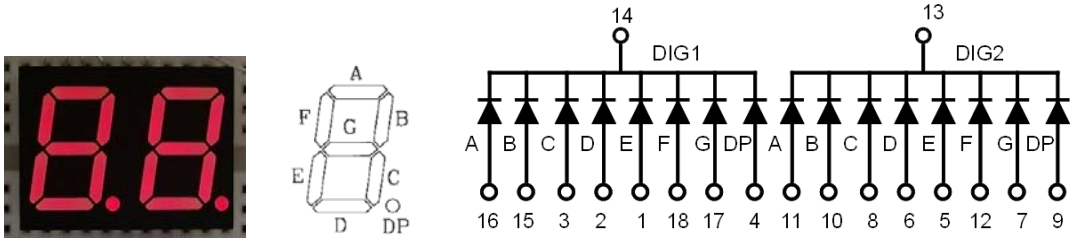
１．実験の目的

本実験では，ディジタル回路設計の基礎として**組み合わせ回路**と**順序回路**を学ぶ.

●組み合わせ回路の実験

家庭にある炊飯器等の電化製品はどれも電気で動作している．最近の電化製品は単純なものは少なく，「マイコン炊飯器」，「センサー付きエアコン」等，どれもが複雑な機能を持つ．このような複雑な機能を実現するためにディジタル回路は欠かせない.

たとえば，ＬＥＤ（発光ダイオード）は２つの状態「点灯」，「消灯」を表現できるが，その形状が棒状のＬＥＤを７つ組合せて８の字型に並べれば０から９までの数字を表すことのできる表示板になる（これを「**７セグメント表示器**」という．図１）．また，きれいに碁盤の目のように並べればコンピュータディスプレイ装置のようなドットマトリクス表示器にもなる．このような表示器は，多くのＬＥＤの「点灯」，「消灯」を制御して実現している.



(a)点灯時の概観　(b)表示部　(c)内部接続

図１：２桁の数を表示できる７セグメント表示器 C-552SRD

また，コンピュータの中も同様である．数値や文字を記憶するということは２進数を覚えることと同じである．２進数は「１」と「０」の文字によって表現される単純なもので，それらを用いて加減乗除算等の演算も実現できる.

このように，「ＬＥＤの点灯・消灯」や「コンピュータ内の２進数」は２つの状態のうちどちらであるか（２者択一）を表しており，それをたくさん保持すればするほど多くの機能を実現することができる．**論理回路**は，このような状態を１つの信号線（電線）に「電圧がかかっている（H:High)」，「かかっていない（L:Low)」という２つの値で表現し，それを組み合わせることでたくさんの論理動作を実現する回路のことをいう．また，その論理回路を実際の電子回路として実現したものを**ディジタル回路**という.

現代ではパソコンのようにCPU上で動作させるソフトウェアだけでは十分ではなく，ハードウェアも理解したソフトウェア技術者（逆に言えば，ソフトウェアも理解したハードウェア技術者）が社会から強く求められている．それはソフトウェアではできない次のようなハードウェア（電子回路）がたくさんあるからである．

・高速に動作（ハードウェアはソフトウェアと比べて少なくとも数１００倍以上高速）
・不要なものがないので単純（必要最低限のものしかない，つまり小型・軽量）
・集積化可能（多くの機能を１つのシステム上に実現）

たとえば，普段使っているパソコンは，さまざまな仕事を柔軟にこなすことができるが，ソフトウェアを中心にして動いているため動作が遅く，装置そのものが大きいのが難点である．一方，電子回路はトランジスタからＩＣ，ＬＳＩ，ＶＬＳＩと高集積化・多機能化（１つのパッケージの中に多くの機能を実現できる）が進んでおり，一旦動きはじめたら高速に動作し，なかなか壊れない．

このように，今までにない高速・小型・多機能な製品を実現するためには，ソフトウェアだけでは限界があり，どうしてもハードウェアの力を借りなければならない．一方，ハードウェアでは設計等に時間が必要であり，柔軟できめ細かな動作をする製品を実現するためにはソフトウェアの力を必要とする．つまり，ハードウェアの知識だけではダメ，ソフトウェアの知識だけでもダメ，ハードウェアとソフトウェアの両方を知り，それらをバランスよく使いわけることができて初めてよい製品を短期間で開発することができる．現代は，ハードウェアとソフトウェアの両方を使いこなせるエンジニアが求められている．

ディジタル回路（ハードウェア）の設計というと難しそうに思えるが，意外と簡単である．それはパソコンのソフトウェアを作りはじめる時に経験したのと同じように，機能はたくさんあるが，その一つ一つの基本機能は少なく，単純明解だからである．特に初心者には，ディジタル回路の仕組みは理解しているが，作ることに自信がなく難しく感じる人が多いかも知れない．しかし，一度慣れてしまえば，たくさんの機能を実現するディジタル回路を次々と作り出すことができるようになる．この実験では，簡単なディジタル回路の製作を通して，ハードウェアに慣れることを目的とする．

２．実験の原理

ディジタル回路の基礎

まずはハードウェアを実現するためのディジタル回路の基礎を学習しよう．ディジタル回路は，１つの信号線（電線）に「電圧がかかっている（H:High）」，「かかっていない（Ｌ:Low）」という２つの状態を作って動作する．以下では，H状態を「１」，Ｌ状態を「０」に割り当てる**正論理**でディジタル回路の動作を考えることにする（１と０を逆に割り当てる場合を**負論理**という）．つまり，回路上の信号線の状態と２進数の論理の間には，次の表１の関係があるものとする．

表１：電線の状態と２進数の論理の関係（正論理）

電線の状態	２進数の論理
電圧がかかっている（H:High）	1
かかっていない（L:Low）	0

通常，信号線のH状態は約＋５Ｖ，Ｌ状態は約０Ｖとする（**TTL レベル**という）．

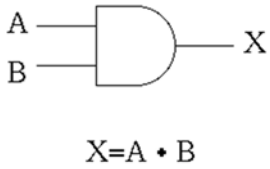
論理演算とゲート素子

２進数の和や積の計算では，ビットごとに演算を行なう．ディジタル回路では次のようなビット演算ができる基本素子が用意されている．これらの基本素子は，数本の入力信号線の論理値に応じて出力信号線の論理値が決まるという動作をするため，信号が門をくぐるというイメージから「**ゲート素子**」と呼ばれている．

それでは，主なゲート素子を紹介しよう．以下の各図で，左側が各ゲート素子の回路図上のシンボルマーク，その下側が動作を表す論理式，右側が具体的な論理入力に対する出力を示す動作表である．

AND 素子（論理積）

入力Ａ，Ｂが<u>ともに</u>１のときのみ出力Ｘが１になる（図２）．



X=A・B

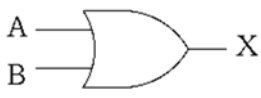
動作表

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

図２：ＡＮＤ素子

OR 素子（論理和）

入力Ａ，Ｂの<u>いずれか</u>が１のときのみ出力Ｘが１になる（図３）．



X=A+B

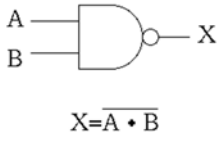
動作表

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

図３：ＯＲ素子

NAND 素子（論理積の否定）

入力Ａ，Ｂがともに１のときのみ出力Ｘが０になる（図４）．



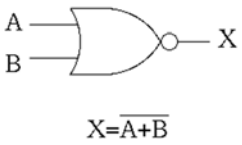
動作表

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

図４：ＮＡＮＤ素子

NOR 素子（論理和の否定）

入力Ａ，Ｂのいずれかが１のときのみ出力Ｘが０になる（図５）．



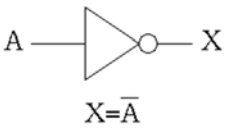
動作表

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

図５：ＮＯＲ素子

NOT 素子（否定）

入力Ａの値を反転させた値が出力Ｘに出力される．つまり，入力Ａが１のとき出力Ｘは０，入力Ａが０のとき出力Ｘは１になる（図６）．**インバータ素子**とも呼ばれる．



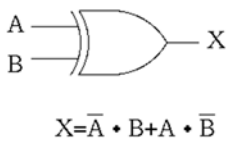
動作表

A	X
0	1
1	0

図６：ＮＯＴ素子

EXOR 素子（排他的論理和）

入力Ａ，Ｂのいずれか一方だけ１のとき出力Ｘが１になる（図７）．入力が両方とも１の時は，出力Ｘは１にはならない．つまり，入力Ａ，Ｂが一致する時のみ出力Ｘが０になる．エクスクルーシブＯＲと呼ぶ．



動作表

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

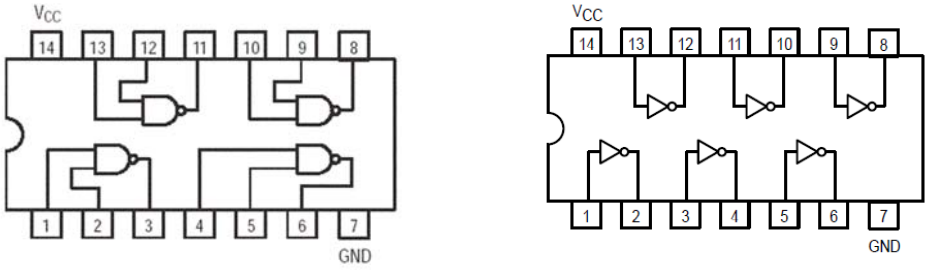
図７：ＥＸＯＲ素子

このようなゲート素子はディジタルＩＣで実現され，「**７４シリーズ**」と呼ばれる代表的なＩＣが使われている．ＩＣの型番が７４から始まることから，この名前が付いている．７４シリーズの詳しい仕様は各社がＷｅｂ等で公開している規格表（データシート）を参照すること．この実験で使用する主なゲート素子を表２に示す．

表２：74 シリーズの主なゲート素子

ゲート素子	**７４シリーズ型番**
AND	７４ＬＳ０８
OR	７４ＬＳ３２
NAND	７４ＬＳ００
NOR	７４ＬＳ０２
NOT	７４ＬＳ０４
EXOR	７４ＬＳ８６

これらのＩＣの多くは１４本のピン（電極）を持ち，電源として１４番ピンに＋５Ｖ（**ＶＣＣ**と書く），７番ピンに０Ｖ（**ＧＮＤ**と書く）を接続することで動作する．他のピンがどのような機能を持っているかは規格表（データシート）から調べること．例えば，７４ＬＳ００（ＮＡＮＤ４つ搭載）と，７４ＬＳ０４（ＮＯＴ６つ搭載）は切り欠き部分を左にして上面から見ると次の図８のピン配置になっている．



(a)７４ＬＳ００　　(b)７４ＬＳ０４

図８：７４シリーズＩＣのピン配置

加算器（足算器）の設計（その１）

ＣＰＵ等の内部や，一般的なディジタル回路で行なわれる演算のうち，最もよく行われるのは加算である．**加算器（足算器）**のディジタル回路の簡単な例として，２進数１桁の加算器を設計してみよう．

２進数１桁の加算器の動作は次の通りである．

「２つの入力Ａ，Ｂに０または１を入力すると，それらを加えた値をＳに出力する．もし桁上がりが発生した時には出力Ｃを１とする．」

次の図９は，人間が行なう２進数の加算演算と，２進加算器の出力の関係を示している．

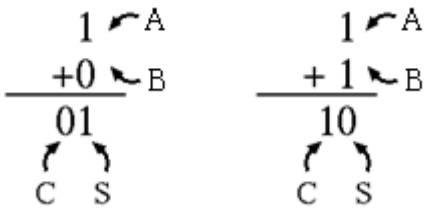


図９：２進数１桁の加算（半加算）

さて，和信号Ｓと桁上がり信号Ｃはどのような回路で実現できるだろうか．順を追って説明する．

加算器はＡとＢを入力の値とし，出力Ｓ，Ｃの値を一意に決める箱（ブラックボックス）と考える．このブラックボックスの動作が加算器の回路動作で，それを表で表してみる．この表を「動作表」という（図１０）．

２進１桁半加算器の動作表

入力		出力	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

図１０：半加算器の動作

この表をよく見てみると，出力ＳはＥＸＯＲと同じ動作，出力ＣはＡＮＤと同じ動作である．これによって，２進１桁の加算器が実現でき，「**半加算器(half adder)**」という（図１１）．

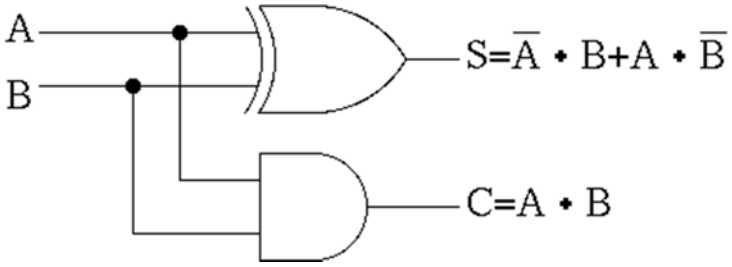


図１１：半加算器の回路図

加算器（足算器）の設計（その２）

半加算器の次は「**全加算器(full adder)**」を設計しよう．全加算器は，ひとつ下の桁からの桁上げも考慮した２進１桁の加算器である．つまり，入力が，Ａ，Ｂだけでなく，下の桁から桁上げ信号C’も考慮して設計しなければならない. 同様に設計してみよう（図１２）.

「下の桁から桁上げ信号C’も考慮し，２つの入力Ａ，Ｂに０または１を入力とし，それらを加えた値をＳに出力する．もし桁上がりが発生した時には出力Ｃを１とする.」

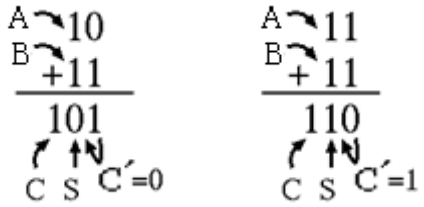
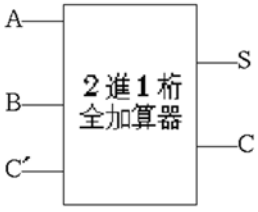


図１２：２進数２桁の加算（全加算）

まず，ＳとＣの結果を実現するための回路動作を示す動作表を書いてみよう．先ほどよりも入力の組合せが増えていることがわかるはずである．入力の選択肢が１つ増えると，組合せの数は２倍になること注意すること（図１３）.



２進１桁全加算器の動作表

入力			出力	
A	B	C´	S	C
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

図１３：全加算器の動作

次は，この動作表を満足する回路を設計するが，先ほどのようにゲート素子を単純に当てはめることはできない．そこで動作を表す論理式を導いた後，必要であれば回路の簡単化を行う．回路の簡単化は１年次の科目「情報基礎理論」で学んだ通りである．図１４に論理式とその簡単化の過程（カルノー図を使用する等）を記述したうえで，全加算器の回路図を実験開始時までに書き込んでおくこと（回路図は図１１を見本とせよ．信号が左から右に流れるように書くのが原則である．また，配線が交差する部分では，接続する場合には●で示さなければならない．一方，接続しない場合は●を記述してはいけない）

全加算器の回路図とそれを得るための途中過程も記述すること．この欄は実験開始前にチェックします．

チェック欄

図１４：全加算器の回路図とその導出過程

全加算器は２進数１桁分の加算回路であるが，これと同じ回路を複数接続することで，何桁の加算器でも作ることができる．次の図１５は４桁の２進加算器の例である．

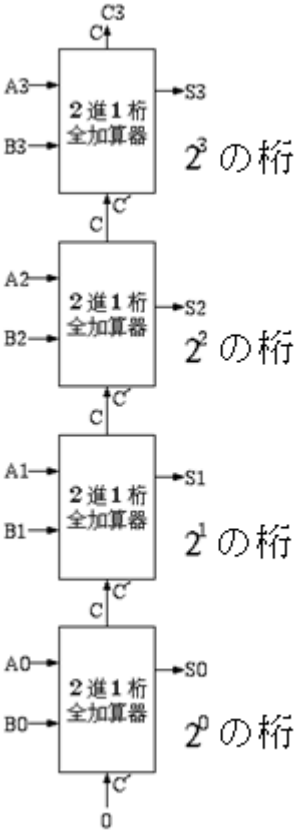


図１５：４桁の２進加算器

まとめ

ＣＰＵ等の複雑な回路は，ここで設計した加算器のような簡単で基本的な回路が多数組合わさって複雑な動作をしている．それは，各回路への入力信号の数を増やすことで，その動作の組合せの数が爆発的に増え，多くの動作を実現しているのである．先ほどの４桁の２進加算器が，２進１桁の全加算器を複数接続して実現できることがその例である．

どこに行っても電気製品が存在し，何を作るにしても電気を使う現代では，さらに小型で複雑な回路が必要とされ，その内部ではディジタル回路が活躍している（最近ではこのようなものを「**組み込みシステム**」と呼ぶことが多い）．現代では，これらの回路はすべて人間が設計しているのではなく，設計時間の短縮と，人間による設計誤りの混入防止のためコンピュータの力を借りて設計している．それが「**電子回路ＣＡＤ(Computer Aided Design)**」や「**ＥＤＡ(Electronic Design Automation)**」と呼ばれている分野である．電子回路ＣＡＤを用いることで，短時間で複雑な回路を間違えずに設計することができる．本実験では，電子回路のＣＡＤの一端として，物理学実験で行った「BSch3V を用いた回路図入力」を行い製作した回路をレポートする．電子回路の詳細な設計や，ＣＡＤを用いた自動設計については，別科目「**ディジタル回路と HDL**」で行うが，ここで得た回路設計・製作の体験が大きく役立つ．

本実験の前半で設計するディジタル回路は，どれも入力値の組み合わせがひとつ決まればそれに応じた値をいつも出力する回路である．このような回路を「**組合せ回路**」という．また，本実験の後半で設計するディジタル回路は，出力値がいつも同じ動作をしない回路で「**順序回路**」という．そのような回路は入力値の変化に応じて値が定まるメモリのような記憶回路である．

実際のディジタル回路の実現について

論理回路の動作についてひと通り理解したら，次は実際にディジタル回路として動かす方法について確認する．一般電化製品では，ＩＣへの電源供給や，素子間の電気的な接続は**プリント基板**を用いて配線する．プリント基板は，一般に緑色のガラス繊維（絶縁体）の板の表面に銅膜が貼ってあり，電線以外の不要な銅膜部分を化学反応で溶かして必要な配線（電線）を実現する（図１６）．このようなプリント基板が使われているのは，多くの配線を小さい面積で実現するためだけでなく，回路の安定動作が期待できるからである．

図１６：プリント基板を用いた配線例（ＲＯＭ，ＲＡＭの周辺配線）

本実験ではプリント基板を製作するには時間が不足するので，簡易にディジタル回路を実現できる「**ブレッドボード（試作用基板）**」を使用する（図１７）．

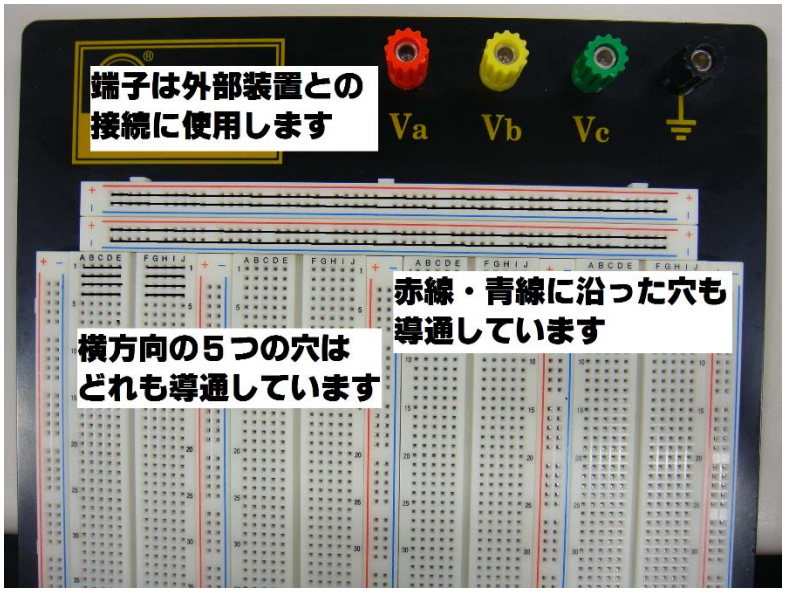


図１７：ブレッドボード

ブレッドボードは電線を差し込む穴がたくさんあいている配線板である．赤色または青色の線に沿った穴は内部で導通しており，主に電源配線用に用いる．赤色は＋５Ｖ，青色は０Ｖ（ＧＮＤ）とする．

また，何も書かれていないが，溝と垂直方向に並ぶ５個の横方向の穴も内部で導通している．ＩＣは溝をまたぐように置き，溝の両側の穴にＩＣのピンを挿入して使う．そして，ＩＣのピンの横方向に並んだ残りの穴に電線を差し込み，ＩＣのピン間を接続する．電線は，**電源の＋５Ｖに赤色，０Ｖ（ＧＮＤ）に黒色を使用**し，それ以外の信号線にその他の色を用いること．また，可能な限り短い電線を使い，できるだけブレッドボードの縦横の穴に沿って配線するときれいに配線できる．用途に応じて色が統一されていない電線や長い電線の利用は回路の理解を阻害するばかりでなく，回路動作を不安定にさせることもあるので避けること．一般に**「見た目できれいな配線は性能がよい配線」**と言われている．なお，ブレッドボード上部の端子は，安定化電源装置や波形発生器との接続に利用する．

１と０の論理値をディジタル回路に入力する方法

ゲート素子やフリップフロップ素子を動かすためには，入力信号線に１と０の値を与える必要がある．入力信号線に１と０の値を与えるには次のようにして行う．単純に，「１」は＋５Ｖ，「０」は０Ｖだから電源の＋５Ｖと０Ｖに接続すればよい，と考えたくなるが誤りである．７４シリーズをはじめ，一般的なＩＣは＋５Ｖをかけた時に信号線に電流が流れ過ぎて壊れてしまうことがある．そこで，図１８に示すスイッチと抵抗を使って１と０

の値を生成する．このとき使う抵抗（４．７ｋΩ）を「**プルアップ抵抗**」という．プルアップ抵抗の抵抗値については使用するＩＣによって異なるが，本実験では４．７ｋΩ の抵抗を使用する．

また，入力信号線をどこにも接続しないことは，その値が「１」なのか「０」なのか不明であり，ディジタル回路にとって思いもよらない動作をさせてしまうことがある．よって，使用しない入力信号線は「０」（常に０Ｖのところ．**グラウンド，ＧＮＤ**と書く）に接続するか，４．７ｋΩ のプルアップ抵抗を通して＋５Ｖに接続し「１」にしておく．本実験で使用する TTL IC は入力信号線をどこにも接続しないと「１」と認識する IC であり，プルアップする必要はないが，明確に値を設定することで安定した動作が期待できる．

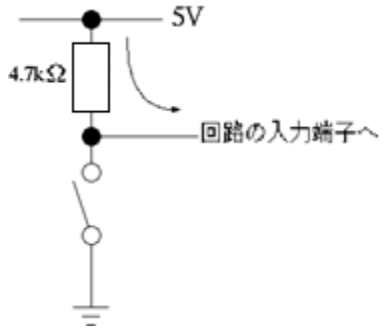


図１８：回路入力とプルアップ抵抗

出力の１と０の論理値を調べる方法

ディジタル回路では，各信号線の論理値が「１」なのか「０」なのかを人間が見てわかるようにＬＥＤ（発光ダイオード）を使用して表示することが多い．ＬＥＤはダイオードの一種で，一方向に電圧を加えたときのみ電流が流れ発光する半導体素子である．電球と違って，電流が流れる時はＬＥＤに無制限に電流が流れてしまい，ＬＥＤを壊してしまう．そこで，そのようなことが起こらないように図１９のように抵抗（330Ω）を直列に挿入して流れる電流を制限する．このような抵抗を「電流制限抵抗」という．

このように回路出力にＬＥＤを接続することで，消灯のとき「１」，点灯の時「０」の論理値であることがわかる（直感的に負論理）．もし点灯イメージとあわせて点灯のとき「１」，消灯の時「０」としたいなら（直観的に正論理），ＮＯＴ素子で出力端子の信号を論理反転すればよい．

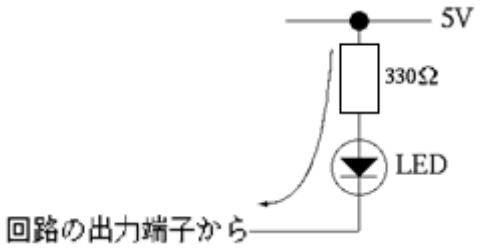


図１９：ＬＥＤによる信号線の状態表示

電源，クロックについて

７４シリーズの TTL IC を動かすには，＋５Ｖの電源が必要である．一般的な電池１本の端子電圧は１．５Ｖなので，それを３本直列に接続して４．５Ｖの電源を作り代用することもできるが，実験では本格的な「**実験用安定化電源装置**」を使用する．実験用安定化電源装置は，出力電圧や最大電流を自由に設定することができ，万一回路がショートした時でもやけどや火災がおきないように保護装置が入っている（図２０(a)）．

また，クロック信号のような＋５Ｖと０Ｖの方形波形は「**波形発生器（シグナルジェネレータ)**」を使用する（図２０(b)）．波形発生器はさまざまな波形の信号を発生させることができるが，ディジタル回路では，「ＴＴＬ／ＣＭＯＳ」と書かれた＋５Ｖと０Ｖのいずれかの方形波形を出力する端子から信号を作製した回路に加える．

(a)安定化電源 PR18-3A　　(b)波形発生器 FG-274

図２０：安定化電源装置（直流電源）と波形発生器（シグナルジェネレータ）

３．実験機材

組み合わせ回路の実験に必要な部品を表３に示す．

表３：組み合わせ回路の実験に必要な部品

品名	**規格・型番**	**個数**
７４シリーズＩＣ	適切な品番	適切な数
抵抗	３３０Ω（１／８W）	適切な数
抵抗	４．７ｋΩ（１／８W）	適切な数
ＬＥＤ	赤	適切な数
電線	各色	適切な数

注意：抵抗は，帯色によって抵抗値が示されている．使用する前に必ずその値を確認してから使用すること．返却するときも必ず帯色によって抵抗値を確認し，違う値の場所に返却しないこと．

４．実験方法

実験１～５のディジタル回路（組み合わせ回路）を次の順で製作せよ．なお，初心者は，急がず焦らず，落ち着いてこの手順に従うこと．この手順に従わないとプロでもブレッドボード上に回路を作るのは難しい．うまく動作せず，どこが間違っているのかわからず，完成するまで時間がかかりとても効率が悪いので絶対に避けること.）

１．ブレッドボード上に IC を回路図中の番号（U1，U2,...）の順に並べて配置する
２．各 IC の電源ピン（主に１４番ピン）を赤色の電線で＋５Ｖ（ＶＣＣ）に接続する
３．各 IC の GND ピン（主に７番ピン）を黒色の電線で０Ｖ（ＧＮＤ）に接続する
４．その他の部品（抵抗，ＬＥＤ，スイッチ等）をブレッドボード上に適当な間隔をあけて配置する
５．＋５Ｖ（ＶＣＣ）に接続された残りの配線を赤色の電線ですべて接続する
６．０Ｖ（ＧＮＤ）に接続された残りの配線を黒色の電線ですべて接続する
７．回路図に記載された各 IC のピン番号を参考にして，他の配線を赤色・黒色以外の電線ですべて接続する
８．他の部品間も赤色・黒色以外の電線ですべて接続し，完成させる
９．回路図上のすべての配線が電線で接続されているかどうかをチェックする

注意

- 本実験では，理論値（真理値表または動作表の値）と実際に測定した値（LED の点灯または消灯）が正しいか否かを確認し，実験ノートに表として記録すること．
- 回路の正しい動作が確認できたら，各実験終了時にスタッフからその動作のチェックを受けること．チェックを受けずに次の実験に進むことのないようにすること．
- 所望の動作が確認できないときは，机上のディジタルマルチテスターを使用して各信号線の値を調べるとよい．テスターのマイナス（黒）端子をＧＮＤに接続し，プラス（赤）端子を調べたい信号線に接続すると，その信号線の値が０Ｖまたは５Ｖで値が０なのか１なのか判断できる．
- 連続する実験では共通部分が多く存在している．実験ごとにすべての回路を作り直すのではなく，流用できる部分は残し新規部分のみ追加して実験時間を短縮せよ．

レポートは, 各実験でブレッドボード上に作製した回路を, 回路図入力 CAD「BSch3V」で入力した回路図として直接印刷して報告するのみとする（２０１７年度から）．BSch3V は,「ファイル」メニュー→「印刷オプション」→「ページフィット」にチェックを入れてＡ４用紙（横）に印刷すること．

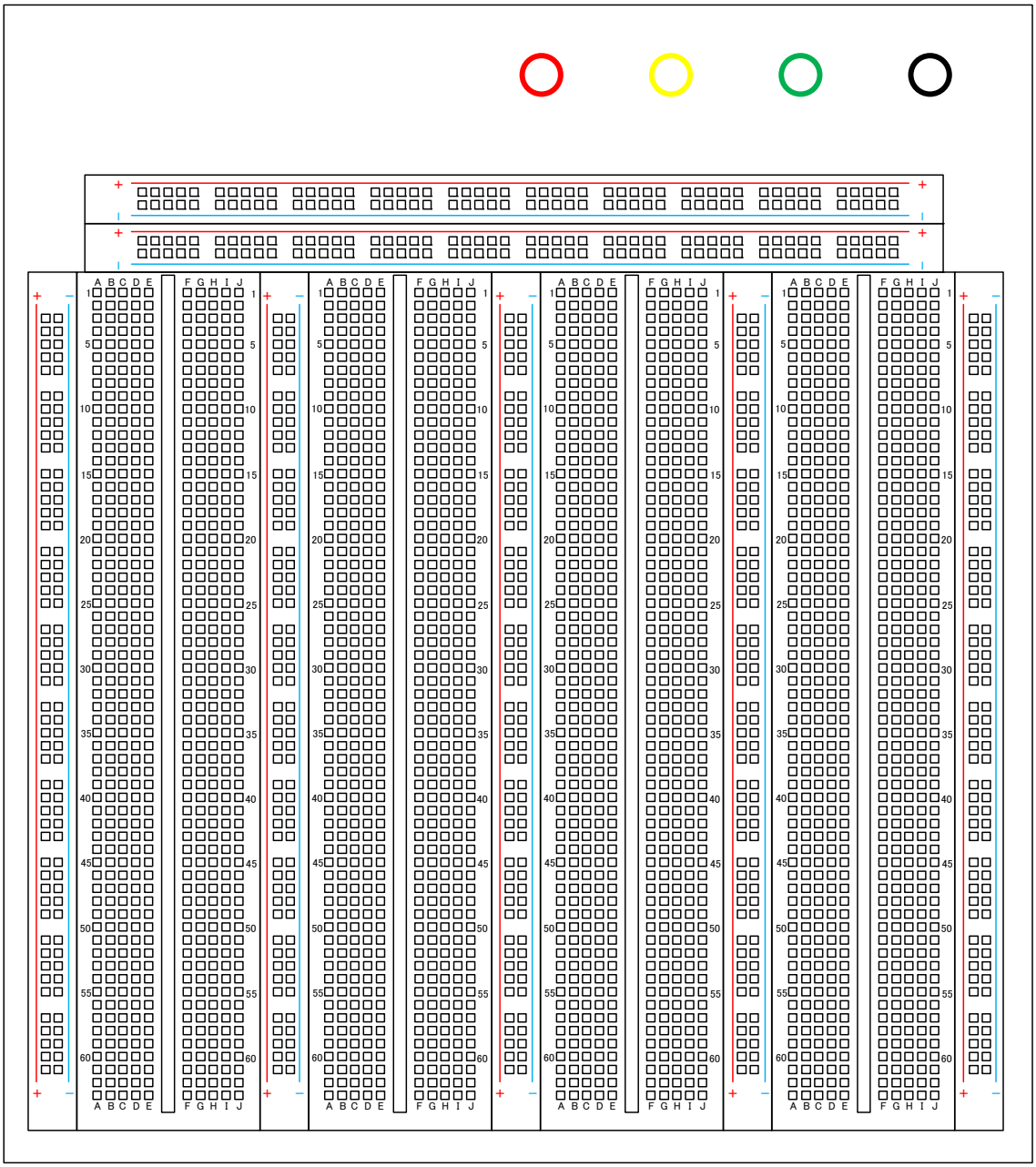
「BSch3V」（水魚堂） http://www.suigyodo.com/online/schsoft.htm

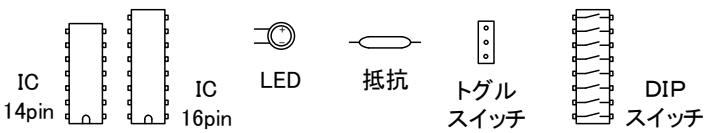
（参考）「BSch3V」はフリーソフトである．上記アドレスにアクセスして各自のパソコンにインストールすることができる．また，電気電子工学実験室のパソコン，情報センターのパソコンにはすでに「BSch3V」がインストールされている．

参考

製作した回路を記録するためのブレッドボードの図を次に用意した．コピーするか，次のところからダウンロードして使用するとよい．

http://www.ee.secu.chukyo-u.ac.jp/





実験１

ＮＡＮＤ（７４ＬＳ００）の動作を確認する回路を作製し，その動作を確認せよ．

（回路の作製手順）

1. ＮＡＮＤ素子１つに，２つの入力用スイッチおよび１つの出力確認用 LED を接続した回路図が次の場所に保存されているので，実験開始前に入手せよ．

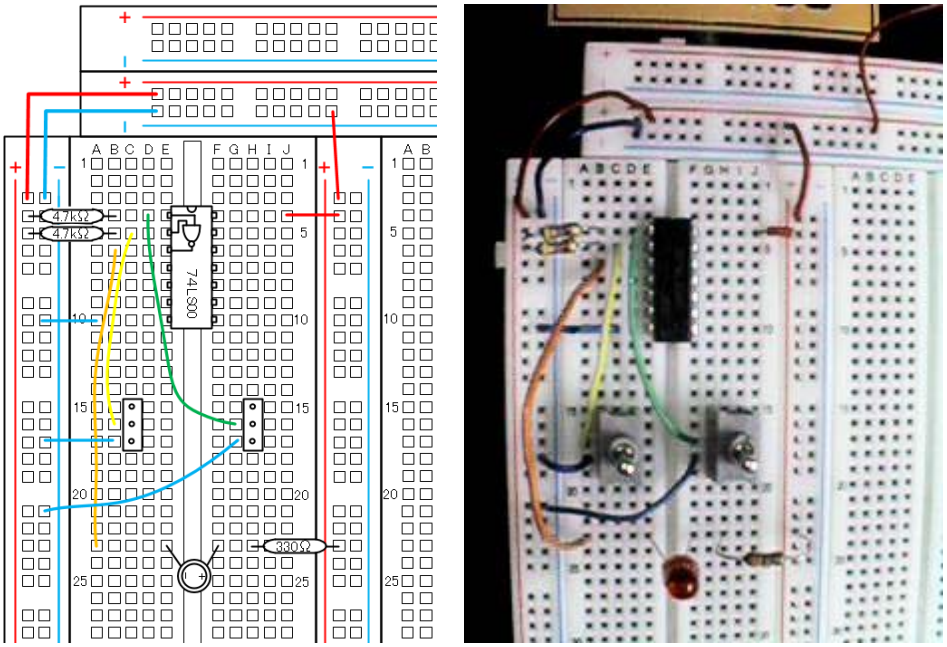
   http://www.ee.secu.chukyo-u.ac.jp/

ここに入手した回路図を貼り付けるか，印刷された回路図を提示すること．

図２１：７４ＬＳ００動作確認回路の回路図

2. 入手した回路図を参考にして，前述の実験方法の手順に従ってブレッドボード上に回路を製作せよ（図２２）．このとき，ブレッドボード上部の赤色・黒色端子を＋５Ｖ（赤）とＧＮＤ（黒）として安定化電源装置との接続に使用すること．このとき，配線作業を集中して行なわないと人間は配線ミスを起こすことに注意すること．
3. 安定化電源装置とブレッドボードの上部端子を接続する前に，安定化電源装置単独で＋５Ｖが出力されるように電圧計を見ながらVOLTAGEつまみとFINEつまみを回して調整する．一旦安定化電源装置のスイッチを切り，＋端子と－端子を安定化電源装置の＋出力端子とＧＮＤ端子をバナナジャック付き電線等で接続(短絡)し，CURRENT つまみで最大電流を１Ａ程度に調整する．調整が終わったら，安定化電源装置のスイッチを切った後，短絡していた電線を取り除く．これで作製した回路内で短絡（ショート）していても大電流が流れずＩＣや回路が壊れにくくなる．
4. ブレッドボード上部の＋５Ｖ端子に赤色の電線で，ＧＮＤ端子に黒色の電線で安定化電源と接続する．

5. 安定化電源のスイッチを入れる．（もしＣＣと書かれた赤ランプが点灯したり，電流計が１Ａ程度振れたりした場合は，作製した回路内で短絡（ショート）しているのですぐに安定化電源装置のスイッチを切って回路を確認すること）
6. 作製した回路の入力信号線に０または１の状態を与えるために２つのスイッチを動かし，そのときのＮＡＮＤ回路の出力信号線の状態をＬＥＤにより観測し，スイッチの位置とＬＥＤの点灯・消灯の関係をすべて表に記録する（表４）．（スイッチはレバーを倒した方と逆側の２端子間が接続（短絡）されることに注意すること）
7. 記録した表から，動作が正しいかどうか確認する．



(a)実態配線図　　(b)実装した結果

図２２：ブレッドボードへの実装例

表４：ＮＡＮＤ回路の動作結果

スイッチの位置（上下左右等を記入）		スイッチの位置に相当する入力論理値（０または１を記入）		ＬＥＤの状態（点灯または消灯を記入）	ＬＥＤの状態に相当する出力論理値（０または１を記入）	ＮＡＮＤの動作確認（○または×を記入）
A	B	A	B			

この動作結果表が NAND の動作と合っているかを確認すること．確認ができたら，スタッフに提示してチェックを受けること．

（以下の実験も，実験１に習って，同様の手順で回路製作と報告を行うこと）

実験２

ＥＸＯＲ（７４ＬＳ８６）の動作を確認する回路を作製し，その動作を確認せよ．

実験３

２進数１桁の全加算器回路を作製し，その動作を確認せよ．

実験４

「２つのスイッチで２桁の２進数を表わす回路」を２つ使用することで，両者の２進数が一致したとき LED が点灯する回路を作製し，その動作を確認せよ．

実験５

３個のスイッチのうち２個だけが同時にHの時，LED が発光する回路を作製し，その動作を確認せよ．

●順序回路の実験

ここまでに扱った「**組合せ回路**」は入力信号の値が決まるとそれに応じて出力信号の値も一意に決まる回路であった．それに対して「**順序回路**」は過去の状態を覚えておき（メモリ），その状態と入力値をもとに値を出力する回路のことである．つまり，同じ入力値に対して出力値がいつも同じ値とは限らない回路のことである．

フリップフロップ

メモリは「**フリップフロップ（ＦＦ）**」と呼ばれる１ビットを記憶することのできる素子の集合である．フリップフロップには多くの種類があるが，ここでは，「ＲＳ型フリップフロップ」，「D型フリップフロップ」，「ＪK型フリップフロップ」そして「T型フリップフロップ」と呼ばれる記憶素子を紹介する．

ＲＳ型フリップフロップ

ＲＳ型フリップフロップ（ＲＳ－ＦＦ）は，出力の値Ｑを Ｈレベル（１レベル）に設定する$\overline{\text{Ｓ}}$端子と，Ｌレベル（０レベル）に設定する$\overline{\text{Ｒ}}$端子の２つ入力端子からなる基本的なフリップフロップである．$\overline{\text{Ｓ}}$端子を０レベルにすると出力が１になり，１に戻してもそのまま出力は１を保持し続ける．また$\overline{\text{Ｒ}}$端子を０レベルにすると出力が０になり，１に戻してもそのまま出力は０を保持し続ける．（「$\overline{\ }$」は，否定を表す．記号が使えない時は記号の後に「’」をつけて表すこともある．Ｓは「セット」，Ｒは「リセット」の意味である．）

ＲＳ型フリップフロップ回路の内部はＮＡＮＤ素子を２つ組合せ，お互いの出力を他方の入力とするような回路で構成されている．出力端子$\overline{Q}$はＱの逆の値を出力する（図２３）．

７４シリーズのＴＴＬでは７４２７９がＲＳ－ＦＦを４つ搭載している．

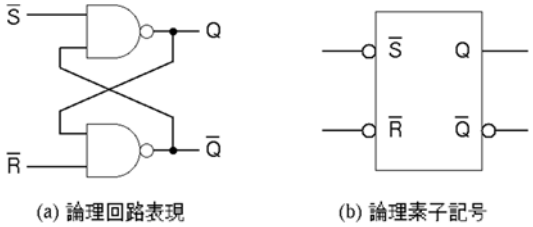


図２３：ＲＳ－ＦＦ素子（正論理表現）

また，ＲＳ－ＦＦを負論理表現すると，図２４のようになる．

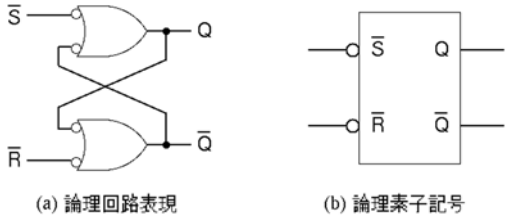


図２４：ＲＳ－ＦＦ素子（負論理表現）

ＲＳ－ＦＦは表５のように$\overline{S}$端子と$\overline{R}$端子の信号を変化させた瞬間に値を記憶する．

表５：ＲＳ－ＦＦ素子の機能表（Qoは直前に覚えていた値を表す）

$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$
0	0	－	－
0	1	1	0
1	0	0	1
1	1	Qo	$\overline{Qo}$

D型フリップフロップ

Ｄ－ＦＦは，クロック信号入力端子（ＣＬＫ）が０から１に変化する瞬間のＤ端子の信号状態を記憶して出力端子Ｑに出力するフリップフロップ素子である．出力端子$\overline{Q}$はＱの逆の値を出力する．

このようにクロック信号が０から１に変化した瞬間の値を記憶するＦＦ素子を「**ポジティブエッジトリガＦＦ**」という（逆に，クロック信号が１から０に変化した瞬間の値を記憶するＦＦ素子を「**ネガティブエッジトリガＦＦ**」という）．

また，ＣＬＲ端子がある場合は，出力端子Ｑの値を強制的に０に設定（**クリア**）することができる．その逆にＰＲ端子は，出力端子Ｑの値を強制的に１に設定（**プリセット**）することができる（図２５）．

７４シリーズのＴＴＬでは７４７４が２つ，７４２７３が８つのＤ－ＦＦを搭載している．

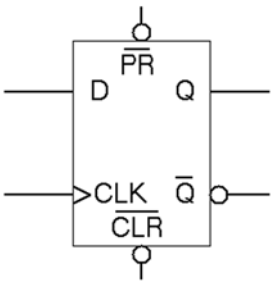


図２５：Ｄ－ＦＦ素子

Ｄ－ＦＦの動作は表６に従う．

表６：Ｄ－ＦＦ素子の機能表

（Qo は直前に覚えていた値を表す．*は動作を保証していない）

$\overline{PR}$	$\overline{CLR}$	CLK	D	Q	$\overline{Q}$
0	1	×	×	1	0
1	0	×	×	0	1
0	0	×	×	1*	1*
1	1	↑	1	1	0
1	1	↑	0	0	1
1	1	0	×	Qo	$\overline{Qo}$

ＪＫ型フリップフロップ

クロック信号の**立ち下がり（負エッジ）**の瞬間に，端子ＪおよびＫの値によって定まる値を記憶し，端子Ｑに出力する．出力端子$\overline{Q}$はＱの逆の値を出力する．また，ＣＬＲ端子は，次のクロックの立ち下がりで強制的に出力の値をＬレベル（０レベル）に設定する．ＰＲ端子は，次のクロックの立ち下がりで強制的に出力の値をＨレベル（１レベル）に設定する（図２６）．

７４シリーズのＴＴＬでは７４１１２がＪＫ－ＦＦを２つの搭載している．

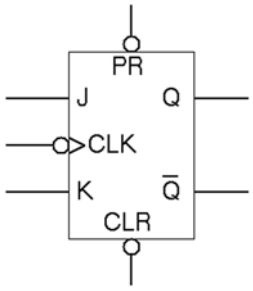


図２６：ＪＫ－ＦＦ素子

ＪＫ－ＦＦが記憶する値は表７に従う．

表７：ＪＫ－ＦＦ素子の機能表（Qo は直前に覚えていた値を表す）

PR	CLR	CLK	J	K	Q	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H*	H*
H	H	↓	L	L	Qo	$\overline{Qo}$
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	TOGGLE	
H	H	H	X	X	Qo	$\overline{Qo}$

Ｔ型フリップフロップ

Ｔは**トグル(toggle)**の意味で，トリガ（クロック）パルスが入力されるたびに出力 Q が反転するＦＦである．Ｔ－ＦＦは入力端子としてＴ端子とＣＬＫ端子を持ち，Ｔ端子がＨレベルの状態でＣＬＫ端子に負エッジトリガが入力されると出力が反転する．Ｔ端子がＬレベルの時，出力は反転しない（図２７）．

Ｔ－ＦＦはＤ－ＦＦやＪＫ－ＦＦを使用して間単に作製できるので，７４シリーズには用意されていない．

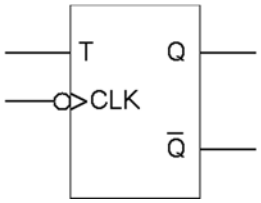


図２７：Ｔ－ＦＦ素子

Ｔ－ＦＦが記憶する値は表８に従う．

表８：Ｔ－ＦＦ素子の機能表
（Qo は直前に覚えていた値を表す）

CLK	T	Q	$\overline{Q}$
↓	L	Qo	$\overline{Qo}$
↓	H	$\overline{Qo}$	Qo

ここまで説明したフリップフロップ素子を使うことで，過去に設定した値（状態）によって，出力が多彩に変化する回路を実現することができる．

シフトレジスタ

フリップフロップを応用した回路の一つに，「**シフトレジスタ**」がある．シフトレジスタは，ＲＳ－ＦＦ回路の出力を次段のＦＦ回路の入力とすることにより，記憶している値を次段のＦＦへ次々と受け渡すことができる回路である．

７４シリーズのＴＴＬでは７４ＬＳ１６４が８つのＦＦでシフトレジスタを構成しており，「**８ビットシフトレジスタ**」と呼ばれている．

７４ＬＳ１６４には，通常のＲＳ－ＦＦと違って次段に値を受け渡すタイミングを与えるためのクロック端子があり，クロック信号が入るたびに次のＦＦに値を移動させる．またＦＦのＲ，Ｓ端子が正論理（１のとき有効）になっている．つまり，前段の出力端子Ｑと次段の入力端子Ｓ，前段の出力端子$\overline{\mathbf{Q}}$と次段の入力端子Ｒを接続するだけでＦＦ間のデータ受渡しが行われている（図２８，２９）．

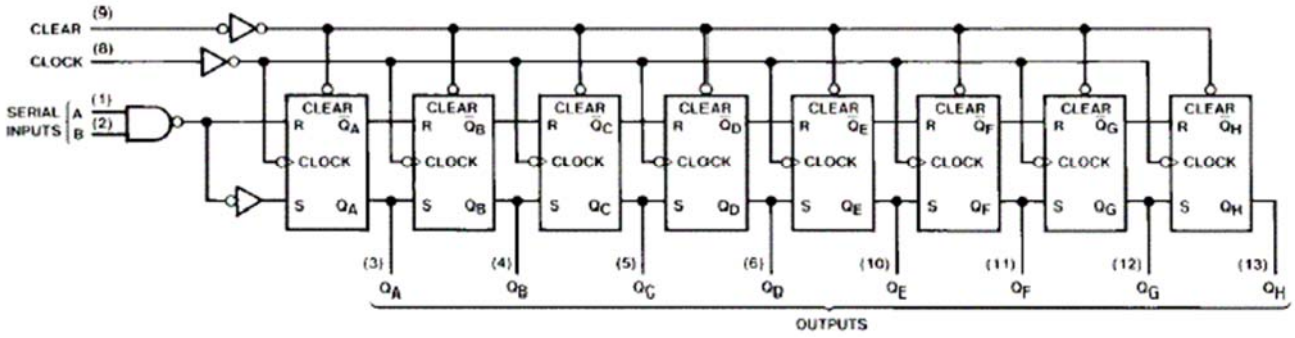


図２８：74LS164 の内部構造

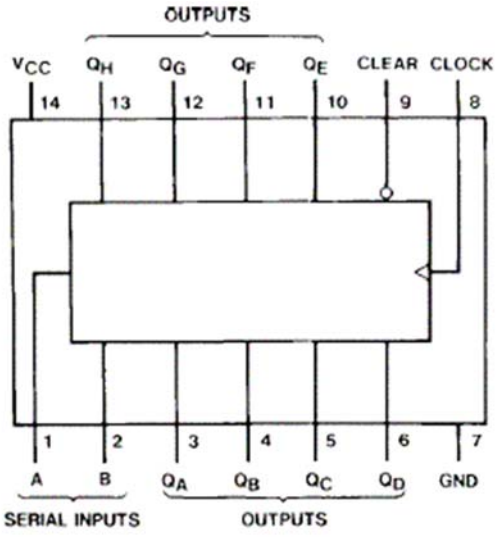


図２９：74LS164 のピン配置

非同期２進カウンタ

ＦＦの出力値がＨレベル（１レベル）からＬレベル（０レベル）になるとき，次段のＦＦの値が反転するようにＦＦを接続すると，ＦＦに２進数の各桁の値（ビット）を記憶させることができる．クロックを負エッジとすることで，２進カウンタ（計数器）が構成できる．ここで「**非同期**」とあるのは，各ＦＦへの信号伝搬には時間がかかり，クロックの瞬間に同時にすべてのＦＦの値が定まらない（同期しない）ためである．

この非同期カウンタは，０から（２のビット乗－１）までの値を計数することができ，最大値までくると次は再び０に戻る（実際には桁あふれが起こっている）．任意の値で０に戻るようにしたい時は，クリアしたい値になったらＬレベル（０レベル）を出力する組合せ回路を作製し，この出力信号をすべてのＦＦのクリア端子に入力すればよい．このようにすると，目的の値になった瞬間にすべてのＦＦはクリアされ，出力値はすべてＬレベル（０レベル）になる．この結果，あたかも目的の値の１つ前の値の次にリセットがかかったように動作する．桁が複数あるときには，この出力信号を桁上げ信号としても利用できる．

７セグメント数字表示器

ＦＦに記憶された２進数はLED等で出力を確認できるが，人間にとっては非常にわかりにくいものである．そこで，人間にも容易にわかるように，数字で表示させることを考える．数字は８の字に配置した７つの線分（セグメント）状のLEDを点灯させることで表示可能である．このようなLEDを**７セグメント数字表示器**という．７セグメント表示器はLEDのカソード側が１本にまとめられた**カソードコモン**タイプと，アノード側がまとめられた**アノードコモン**タイプがある．

本実験で使用するC-552SRDはカソードコモンタイプの７セグメント表示器である．概観とピン配置は図１を参照すること．

BCD７セグメントデコーダ

入力として２進化１０進数（BCD）の４つの信号$A(2^0)$,$B(2^1)$,$C(2^2)$,$D(2^3)$を入力し，７セグメント表示器の７つのセグメントＬＥＤを制御する機能をもつ回路を**BCD７セグメントデコーダ**という．

７４ＨＣ４５１１はカソードコモンタイプの７セグメント表示器をドライブすることができる組合せ回路が構成されているICである．７４ＨＣ４５１１を使用するときは，それぞれのセグメントのＬＥＤに対して電流制限用抵抗を接続することを忘れないこと．

表９はよく使用するフリップフロップとそれを応用した回路を搭載したＩＣをまとめたものである．ＩＣのピン配置や内部構成は規格表（データシート）をWeb等で参照すること．

表９：フリップフロップとその応用回路としてよく使用する主な７４シリーズＩＣ

素子名	７４シリーズ型番
ＲＳ型ＦＦ（４個入）	７４ＬＳ２７９
D型ＦＦ（２個入）	７４ＬＳ７４
ＪＫ型ＦＦ（２個入）	７４ＬＳ１１２
D型ＦＦ（８個入）	７４ＬＳ２７３
８ビットシフトレジスタ	７４ＬＳ１６４
２進＋５進（１０進），２進＋８進（１６進）カウンタ	７４ＬＳ９０/９３
２進＋８進（１６進）カウンタ	７４ＬＳ２９３
ＢＣＤ７セグメントデコーダ	７４ＨＣ４５１１

チャタリング

有接点スイッチをオンからオフに切替えると，スイッチレバーのわずかな振動等のため，接点の断接がわずかな間に何度も切り替わる現象が発生する．この現象を**チャタリング**といい，スイッチのオン・オフによって１つのパルスをディジタル回路に入力したつもりでも実際には不特定多数のパルスが入力され誤動作の原因になる．特に順序回路のクロック信号では，チャタリングは絶対に避けなければならない．このようなチャタリング現象を防止して確実に１つのパルスを発生させるためにＦＦを用いる方法がある．つまり，一度でもＨレベルになったらどのような入力が入っても出力を変化しないように過去の状態をＦＦで記憶すればよい．ＪＫ－ＦＦを用いたチャタリング防止回路の例を図３０に示す．

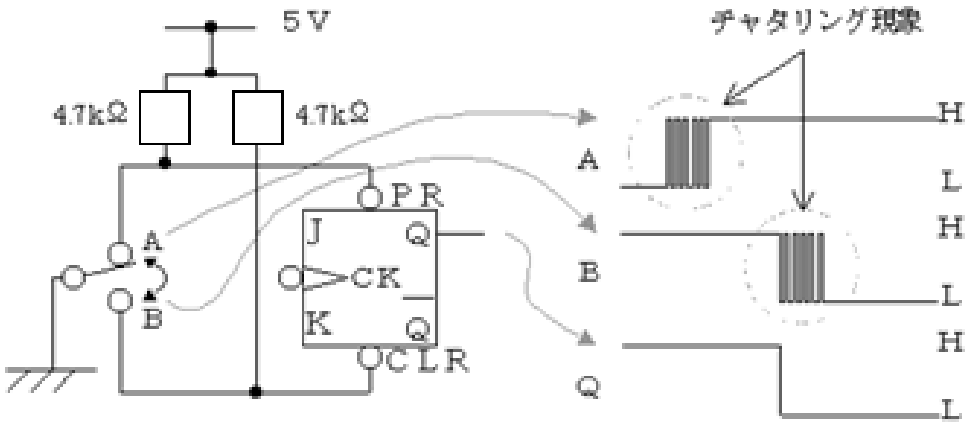


図３０：チャタリング防止回路

５．実験機材

実験に必要な部品を表１０に示す．

表１０：後半の実験に必要な部品

品名	規格・型番	個数
７４シリーズＩＣ	適切な品番	適切な数
抵抗	ＬＥＤ電流制限用３３０Ω（１／８W）	適切な数
抵抗	プルアップ用４．７ｋΩ（１／８W）	適切な数
ＬＥＤ	赤	適切な数
７セグメント表示器	赤	適切な数
スイッチ	トグル型，ＤＩＰ型	適切な数
電線	各色	適切な数

６．実験方法

これまでと同様に，実験６～１２のディジタル回路（順序回路）をこの順で設計せよ．

実験６

トグルスイッチ１つとＪＫ－ＦＦ（７４ＬＳ１１２の中の１つ）を使ってチャタリング防止回路を作製し，その動作をＬＥＤの点灯・消灯によって確認せよ．このとき，スイッチをどちらに倒したらチャタリング防止回路の出力がＨレベル，またはＬレベルになる時の点灯・消灯を記録せよ（表１１）．

表１１：実験６の動作結果

スイッチの位置	スイッチの位置に対する入力論理値	ＬＥＤの点灯/消灯	ＬＥＤに対応する出力論理値

実験７

７４ＬＳ２７３（８回路ポジティブエッジトリガＤ－ＦＦ，図３１）の８つのＤ－ＦＦを２進数における各桁の記憶素子と考え，教員から指定された０から１２７までの１０進数を入力し，記憶させる回路を作製し，その動作を確認せよ．（表１２）．

表１２：教員から指定された値

１０進数	２進数

1. 各 D-FF の D 端子への２進数入力信号を DIP スイッチ（１列に端子が並んだ小さなスイッチ群）とプルアップ抵抗を使用せよ．
2. 入力するクロック信号はトグルスイッチとプルアップ抵抗による信号に実験６で作製したチャタリング防止回路を接続した回路として構成せよ．
3. まず，$\overline{\text{CLR}}$端子に接続されたリセットスイッチを操作し，８つのＤ－ＦＦに一旦０を記憶させよ．（ＬＥＤがすべて点灯）
4. 次に，チャタリング防止回路のトグルスイッチを動かすことでクロック信号を入力し，各 FF に記憶されている値を LED の点灯・消灯で確認せよ．

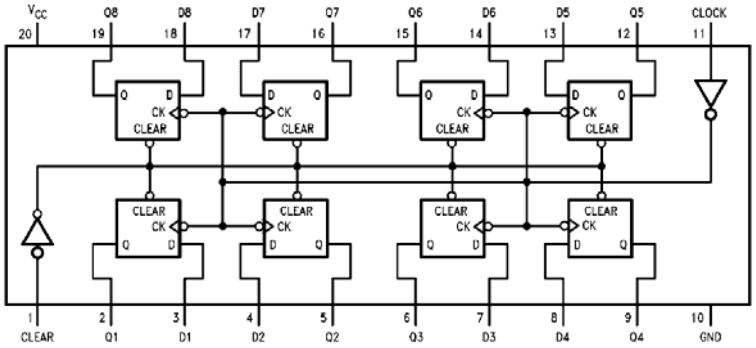


図３１：74LS273 のピン配置

実験８

ネガティブエッジトリガＪＫ－ＦＦ　７４ＬＳ１１２を使ってＴ－ＦＦを実現する回路を作製し，その動作を確認せよ．クロックには実験６で作製したチャタリング防止回路の出力を用いよ．

実験９

非同期式カウンタ７４ＬＳ２９３を使用して１６進カウンタを作製し，その動作を７セグメント表示器 C-552SRD で値を表示させて確認せよ（７４ＬＳ２９３は７４ＬＳ９３のピン配置を変えたものである．図３２）．

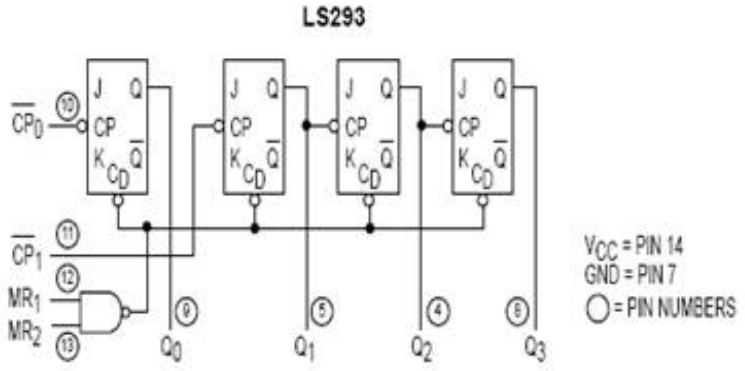
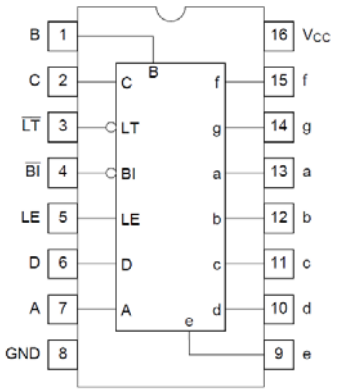


図３２：74LS293 のピン配置

- カウンタの４ビットBCD出力に７セグメントデコーダ７４ＨＣ４５１１(図３３)を接続し，７セグメント表示器C-552SRDの右側１桁に数字１文字を表示する回路を作製せよ．(７４ＨＣ４５１１の入力端子LEは０，$\overline{\text{BI}}$は１，$\overline{\text{LT}}$は１に固定せよ)
- カウンタのクロックに実験６で作製したチャタリング防止回路の出力を接続し，１６進カウンタの動作を確認せよ．
  (特に，１６進数ＡからＦまではどのような表示になるかも記録せよ)
- チャタリング防止回路の代わりに直接トグルスイッチを使ったオンオフパルスを与えると回路にチャタリング現象が発生する．トグルスイッチの出力を直接クロック信号に接続し，チャタリングがどのように起こったか記録せよ．(チャタリング防止回路のPR端子またはCLR端子は直接トグルスイッチから出力されるオンオフパルスなので，チャタリングが含まれた入力信号としてそのまま流用できる)
- また，チャタリング防止回路の代わりに，ファンクションジェネレータのＴＴＬ出力を接続し，高速に動作させた場合でも回路が安定して動作することを確認せよ．



入力							出力							
LE	$\overline{\text{BI}}$	$\overline{\text{LT}}$	D	C	B	A	a	b	c	d	e	f	g	表示
X	X	L	X	X	X	X	H	H	H	H	H	H	H	8
X	L	H	X	X	X	X	L	L	L	L	L	L	L	ブランク
L	H	H	L	L	L	L	H	H	H	H	H	H	L	0
L	H	H	L	L	L	H	L	H	H	L	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	H	L	H	2
L	H	H	L	L	H	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	L	H	H	L	L	H	H	4
L	H	H	L	H	L	H	H	L	H	H	L	H	H	5
L	H	H	L	H	H	L	L	L	H	H	H	H	H	6
L	H	H	L	H	H	H	H	H	H	L	L	L	L	7
L	H	H	H	L	L	L	H	H	H	H	H	H	H	8
L	H	H	H	L	L	H	H	H	H	L	L	H	H	9
L	H	H	H	L	H	L	L	L	L	L	L	L	L	ブランク
L	H	H	H	L	H	H	L	L	L	L	L	L	L	ブランク
L	H	H	H	H	L	L	L	L	L	L	L	L	L	ブランク
L	H	H	H	H	L	H	L	L	L	L	L	L	L	ブランク
L	H	H	H	H	H	L	L	L	L	L	L	L	L	ブランク
L	H	H	H	H	H	H	L	L	L	L	L	L	L	ブランク
H	H	H	X	X	X	X	(注)							(注)

【注】 LE＝Lのときに印加されたBCDコードによって決まる。

図３３：７４ＨＣ４５１１のピン配置と機能表

実験１０

実験９で作製した１６進カウンタに回路を追加して，１０進カウンタを作製し，その動作を確認せよ．（１０までクロックパルスを数えたらすぐに全ての記憶を０にクリアする回路を追加する）

実験１１

実験１０で作製した１０進カウンタを２つ作製し，非同期ＢＣＤ６０進カウンタを作製せよ．また，同様にして非同期２４進カウンタを作製し，その動作を確認せよ．

実験１２

実験１０で作製した非同期ＢＣＤ６０進カウンタで「分」を，同様に作製した非同期ＢＣＤ２４進カウンタで「時」を表すことで，非同期２４時間時計（時分表示）を作製し，その動作を確認せよ．

７．応用課題（時間がある場合に挑戦するとよい）

１２時間時計を実現する回路の回路図を示し，実現せよ．(24 時間時計と同様の設計では動作しない．ヒント：12 時の次は 0 時ではなく 1 時である．)