情報工学実験I
実験1-ゲート回路-

レポート作成者: 055702B 池野谷克俊

共同実験者: 055730H 新垣大志
055752J 比嘉安史
035710D 内間新祐

実験実施日: 2006年 6月 2日 金曜日
提出日: 2006年 6月 9日 金曜日

1 実験目的

現代社会に欠かすことのできないコンピュータは, 大規模なディジタル回路によって構成されている. 本実験では, ディジタル回路の構成要素である基本ゲート回路と論理演算の基礎を習得することを目的とする. また本実験では, NAND ゲートを用いて他のゲート回路 (NOT,AND,OR,NOR,XOR) を構成することによって, 汎用ロジック IC およびオシロスコープ, 直流電源, 発振器などの測定機器の基本的な使用法についても学ぶ.

2 実験概要

NAND ゲートを用いて NOT,OR などの回路を設計するために, まずオシロスコープ, 直流電源などの機器の使い方を学んだ, 例えばケーブルの接続や各ボタンの説明, 直流電源の設定など. そのあと, 実際に設定するまえに,NAND ゲートを用いて NOT,AND,OR,NOR,XOR ゲートを紙の上で設計した.
それから, NAND 回路が 4 つ集まってできた IC を使って, 実際にブレッドボード上に回路を設計した. その際に, ブレッドボードは縦の列や, 横の列で繋がっていることを学び, 直流電源を使ってボード上に電源となる列を決め, アースとなる列を決めた. それから IC の NAND ゲートを使って, さきほど紙の上で設計した回路を見ながら,NOT,AND,OR,NOR,XOR を設計した. 最後に設計した回路をオシロスコープを使いちゃんと回路ができているかを確かめた.

3 実験結果

1. 実験 (1) (5) の結果について
   本実験の結果に関しては, 報告を省略します.

2. 実験 (6) の結果について

3.1 回路図

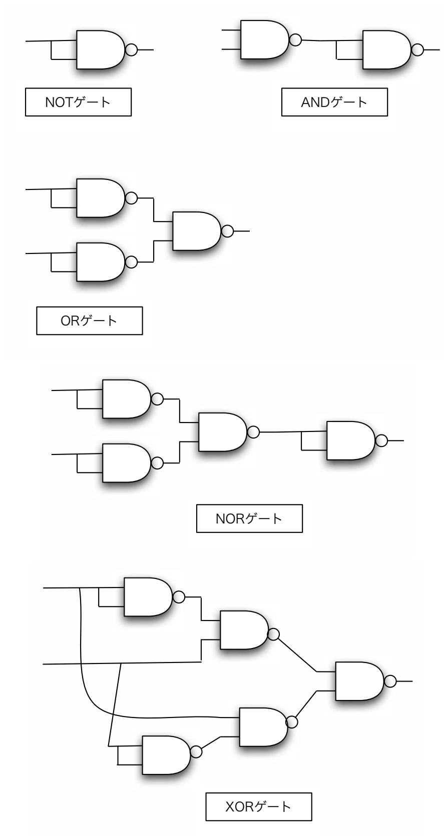

図 1: NOT,AND,OR,NOR,XOR 回路

3.2 上記の回路が, それぞれゲートの機能を実現しているか証明

- NOT
  入力を A, 出力を F とする

$$F = \overline{A \cdot A} = \overline{A} + \overline{A} = \overline{A} \tag{1}$$

- AND
  入力を A,B 出力を F とする

$$F = \overline{\overline{A}{\cdot}\overline{B}} = A \cdot B \tag{2}$$

- OR
  入力を A,B 出力を F とする

$$F = \overline{\overline{A \cdot B}} = \overline{\overline{A}} + \overline{\overline{B}} = A + B \tag{3}$$

- NOR
  入力を A,B 出力を F とする

$$F = \overline{\overline{\overline{\overline{A}} \cdot \overline{\overline{B}}}} = \overline{\overline{\overline{A}} + \overline{\overline{B}}} = \overline{A + B} \tag{4}$$

- XOR
  入力を A,B 出力を F とする

$$F = \overline{\overline{\overline{\overline{A} \cdot B} \cdot \overline{A \cdot \overline{B}}}} = \overline{\overline{\overline{A} \cdot B}} + \overline{\overline{A \cdot \overline{B}}} = \overline{A} \cdot B + A \cdot \overline{B} \tag{5}$$

3.3 上記の各回路図が導かれた経緯の説明

- NOT 回路
  NAND 回路の真理値表から求める.
- AND,OR,NOR,XOR 回路
  ド・モルガンの法則, 加法標準形を用いて, 式を変形していく.

3.4 実験 7 の結果

NAND ゲート IC を用いてブレッドボード上に設計した回路は各回路 (NOT,AND,OR,NOR,XOR) と等価な動作をしました. ブレッドボードに設計した回路の構成などは省略します.

4 考察

4.1 実験 6 の考察

- NOT
  NAND 回路 1 つで構成されているので, これ以上減らすことはできない.
- AND
  AND 回路を NAND 回路 1 つで構成することはできないので, これ以上減らすことはできない.

- OR
  ド・モルガンなどをもちいて色々計算したが, これ以上減らすことはできなかった.

- NOR
  OR に NOT を付けて NOR にするので, これ以上減らすことはできない.

- XOR
  以下のような回路への書き換えができた.

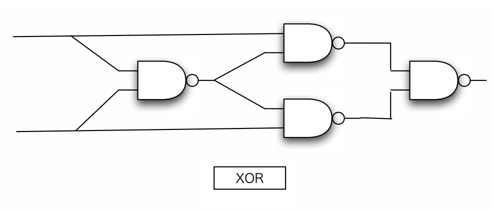

図 2: XOR 回路

NAND 回路 5 個から,NAND 回路 4 個に減らすことができる事が判明した.

4.2 実験 7 の考察

- 配線数を減らす工夫
  ブレッドボードは縦あるいは横の列で繋がっているので, それを用いれば配線数を減らすことが可能. 繋がっている列に配線を多用するのは, 配線を不用意に増やすだけである.

- 配線ミスを減らす工夫
  ブレッドボードはあまり大きいものではないため, 複雑な回路を設計する場合に配線ミスを起こしてしまう場合がある. そのため, 配線ミスを減らすためには常に回路を見やすく設計することが大事になってくる. 自分でブレッドボードに決められた役割をするエリアを作って, 分かりやすくしたり, 配線の密度を減らすことが配線ミスを減らす工夫となる.

4.3 その他の考察

今回の実験を通して得られた新たな知見は, 大きく分けて 2 つである.1 つはオシロスコープについてである. 今回初めてオシロスコープを見たので, 電源の付け方, コードの接続, 画面の調整など色々な事を知った. まだすべての

操作を覚えてはいないが,
信号を観測するためのプローブを接続する端子
電圧軸部
時間軸部
などについては色々操作してみて, だいたい分かった. 2つ目がブレッドボードについてである. ブレッドボードは縦, 横のいずれかの列でつながっていることを学んだ.

5 調査課題

(a) 2変数の論理関数について

- 2変数の論理関数が16種類になる理由

  2変数の論理関数の入力の組み合わせは,(0,0),(0.1),(1.0),(1,1) である. これらの入力に対する出力は1か0である. それぞれの入力の組み合わせに対する出力の種類は2種類であるので,2変数の論理関数の入力の組み合わせは 2^4 で16種類となる.

- 2変数の論理関数の列挙

  $A+B \qquad \overline{A}+B \qquad A+\overline{B} \qquad \overline{A}+\overline{B}$

  $A\cdot B \qquad \overline{A}\cdot B \qquad A\cdot\overline{B} \qquad \overline{A}\cdot\overline{B}$

  $A\cdot\overline{A}+B\cdot\overline{B} \qquad A+B\cdot\overline{B} \qquad A\cdot\overline{A}+B \qquad \overline{A}\cdot B+A\cdot\overline{B}$

  $A\cdot B+\overline{A}\cdot\overline{B} \qquad \overline{A}\cdot\overline{B}+A\cdot\overline{B} \qquad \overline{A}+B\cdot\overline{B} \qquad A+B+\overline{A}+\overline{B}$

- 上の論理関数を NOT,AND,OR を用いて表す

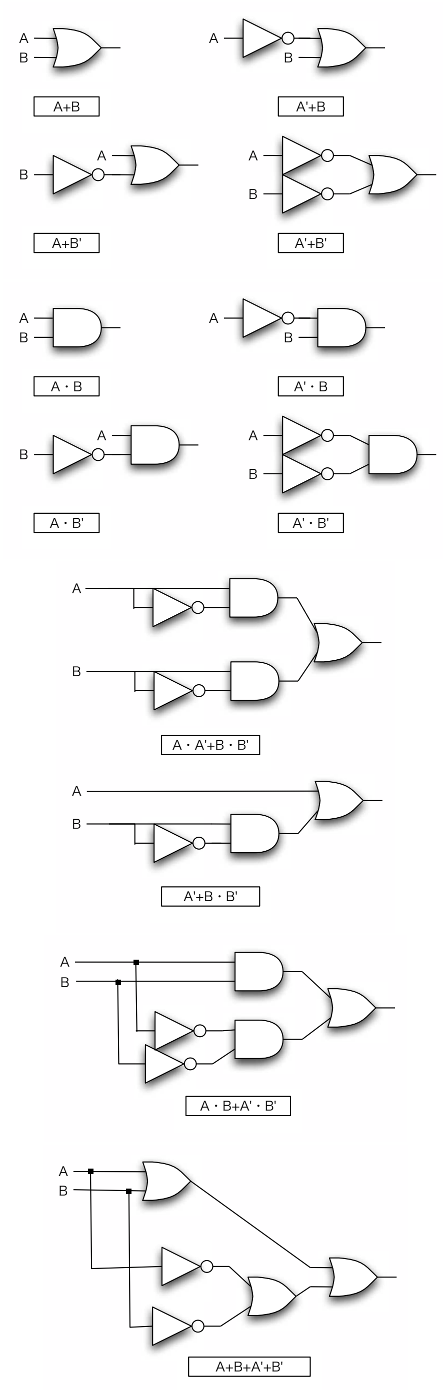

図 3: 上記の論理関数の回路図

(b) 完全系となる論理関数

- NOT,AND で完全系をなす.
  証明:NOT と AND で NAND が作れる.NAND は完全系であるから NOT,AND の 2 つで完全系をなす.
- NOT,OR で完全系をなす.
  証明:NOT と OR で NAND が作れる.NAND は完全系であるから NOT,OR の 2 つで完全系をなす.

(c) 組み合わせ回路と順序回路の違い

組み合わせ回路と順序回路の違いは内部に状態を保持するか, しないかということである. 組み合わせ回路は内部に状態を保持しないが, 順序回路は保持する.

(d) 半加算器, 全加算器について

- 反加算器
  半加算器とは 2 個の 1 桁の 2 進数を加算する論理回路である. 桁上がりは桁上げ出力によって出力.

  回路図

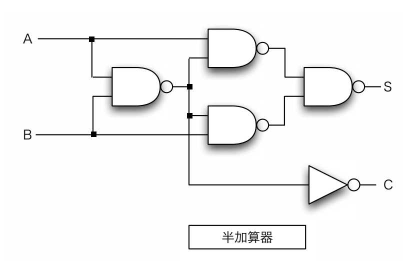

図 4: 半加算器

真理値表

A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

論理関数
$S = \overline{A} \cdot B + A \cdot \overline{B}$
$C = A \cdot B$

- 全加算器
全加算器は、2 進数の 1 つの桁を演算し、下位からの桁上げ入力を含めて出力する。下位の桁上げ出力を上位の桁上げ入力に接続することにより、任意の桁数の 2 進数の加算が可能となる.

回路図

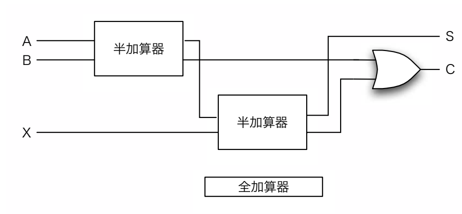

図 5: 全加算器

真理値表

A	B	X	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

論理関数
$S = \overline{A} \cdot \overline{B} \cdot X + \overline{A} \cdot B \cdot \overline{X} + A \cdot \overline{B} \cdot \overline{X} + A \cdot B \cdot C$
$C = \overline{A} \cdot B \cdot X + A \cdot \overline{B} \cdot X + A \cdot B \cdot \overline{X} + A \cdot B \cdot C$

6　感想

今回の課題は図の作成が多く大変だった. また実験の授業の形態としてグループ毎に実験を行うというのは, お互いの意見を交換できたりして, 良いと思う. 次の実験, 課題も頑張っていきたい.

参考文献

[1] Microsoft PowerPoint - Logic3.ppt

`http://66.102.7.104/search?q=cache:Iie7BlbEcvEJ:w1.cs.miyazaki-u.ac.jp/users/yamamori/H18`
論理関数　完全系`&hl=ja&ct=clnk&cd=10&client=safari`

[2] 加算器-Wikipedia-
http://ja.wikipedia.org/wiki/加算器