

情報工学実験1

「基本ゲート回路」

実施日：2010年6月29日

学籍番号： 095707B

氏名： 大城佳明

締切日： 2010年7月6日

共同実験者： 095701B 青木史林

095703J 岩瀬 翔

095759D 諸見里拓治

1 実験目的

現代社会に書かすことのできないコンピュータは、大規模なデジタル回路として構成されている。本実験では、デ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、凡用ロジックICおよびブレッドボード、直流電源などの基本的な使用方法についても学ぶ。

2 実験概要

まず、NANDゲートのみを用いて、NOT,AND,OR,NOR,XORゲートを設計をする。カルノー図や真理値表などを利用して、回路図を描く。そうすることで、基本ゲートと論理演算の基礎を理解しているかがわかる。次に、4011Bの規格のIC(NANDゲートIC)を用いてブレッドボード上に実現する。5種類のゲートを作成することにより、実際の回路を学ぶことが出来る。直流電源、ブレッドボード、4011Bの規格のIC、ピニール線、LEDを使用して回路を作成する。LEDの点滅で実際に5種類の回路が作成出来たかを確認する。そうすることで、凡用ロジックICおよびブレッドボード、直流電源などの基本的な使用方法についても学ぶことが出来る。

3 実験結果

3.1 実験(1)の結果について

3.1.1 NOTゲート

1. 回路図



図 1: NOT

2. 説明

(a) 真理値表にて説明する(図1参照)

表 1: 図 1 の真理値表

A	F
0	1
1	0

(b) Aで入力された値はFに出力される

(c) 表1はNOTゲートの真理表と同じである

3. 経緯

表 2: NAND の真理値表

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

表 3: NOT の真理値表

A	F
0	1
1	0

- (a) 真理値表を使って求める
- (b) 表 2 の入力 A,B が 0,0 の時、出力 F は 1 である
- (c) 表 2 の入力 A,B が 1,1 の時、出力 F は 0 である
- (d) これは表 3 と比べると似ている
- (e) A,B の入力を一つにすることで NOT ゲートができることがわかる

3.1.2 AND ゲート

1. 回路図



図 2: AND

2. 説明

(a) 図 2 から真理値表作成

表 4: 図 2 の真理値表

A	B	F1	F2
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) A,B から入力された値は F1 に出力される

(c) F1 から出力された値さらには F2 に出力される

(d) 表 4 は AND ゲートの真理値表と同じである

3. 経緯

(a) AND の否定が NAND である

(b) つまり NAND の否定が AND である

(c) したがって、NAND ゲートの後ろに図 1 の NOT ゲートを加えれば良い

3.1.3 OR ゲート

1. 回路図

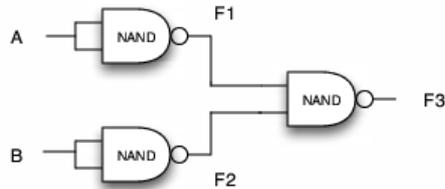


図 3: OR

2. 説明

(a) 図 3 から真理値表作成

表 5: 図 3 の真理値表

A	B	F1	F2	F3
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

- (b) A から入力された値は F1 に出力される
- (c) B から入力された値は F2 に出力される
- (d) F1, F2 から出力された値さらには F3 に出力される
- (e) 表 5 は OR ゲートの真理値表と同じである

3. 経緯

- (a) ド・モルガンの法則を利用して求める
- (b) OR ゲートは式で表すと、「 $A + B$ 」である
- (c) 式

$$\begin{aligned} A + B &= \overline{\overline{A + B}} \\ &= \overline{\overline{A} \cdot \overline{B}} \end{aligned} \tag{1}$$

- (d) A の否定と B の否定を NAND することがわかる

3.1.4 NOR ゲート

1. 回路図

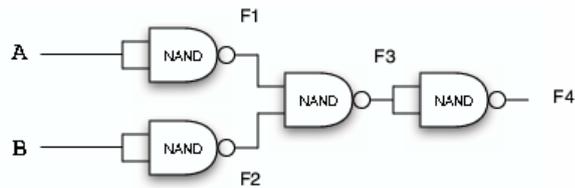


図 4: NOR

2. 説明

(a) 図 4 から真理値表作成

表 6: 図 4 の真理値表

A	B	F1	F2	F3	F4
0	0	1	1	0	1
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	1	0

- (b) A から入力された値は F1 に出力される
- (c) B から入力された値は F2 に出力される
- (d) F1, F2 から出力された値はさらに F3 に出力される
- (e) F3 から出力された値はさらに F4 に出力される
- (f) 表 6 は NOR ゲートの真理値表と同じである

3. 経緯

- (a) OR の否定が NOR である
- (b) したがって、図 3 の OR ゲートの後ろに図 1 の NOT ゲートを加えれば良い

3.1.5 XOR ゲート

1. 回路図

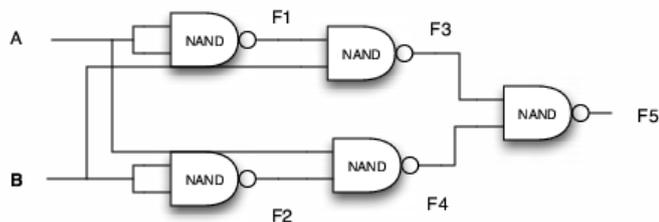


図 5: XOR

2. 説明

(a) 図 5 から真理値表作成

表 7: 図 5 の真理値表

A	B	F1	F2	F3	F4	F5
0	0	1	1	1	1	0
0	1	1	0	0	1	1
1	0	0	1	1	0	1
1	1	0	0	1	1	0

(b) A から入力された値は F1 に出力される

(c) B から入力された値は F2 に出力される

(d) B から入力された値と F1 から出力された値が F3 に出力される

(e) A から入力された値と F1 から出力された値が F4 に出力される

(f) F3 から出力された値と F4 から出力された値がさらに F4 に出力される

(g) 表 7 は NOR ゲートの真理値表と同じである

3. 経緯

(a) ド・モルガンの法則を利用して求める

(b) OR ゲートは式で表すと、「 $(\bar{A} \cdot B) + (A \cdot \bar{B})$ 」である

(c) 式

$$\begin{aligned}
 (\bar{A} \cdot B) + (A \cdot \bar{B}) &= \overline{\overline{(\bar{A} \cdot B) + (A \cdot \bar{B})}} \\
 &= \overline{\overline{(\bar{A} \cdot B)} \cdot \overline{(A \cdot \bar{B})}} \quad (2)
 \end{aligned}$$

(d) A の否定と B を NAND したやつと、A と B の否定を NAND したやつを NAND することがわかる

3.2 実験(2)の結果について

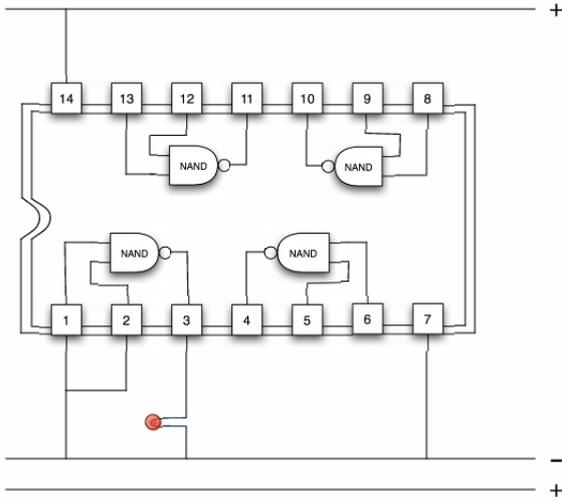


図 6: IC-NOT

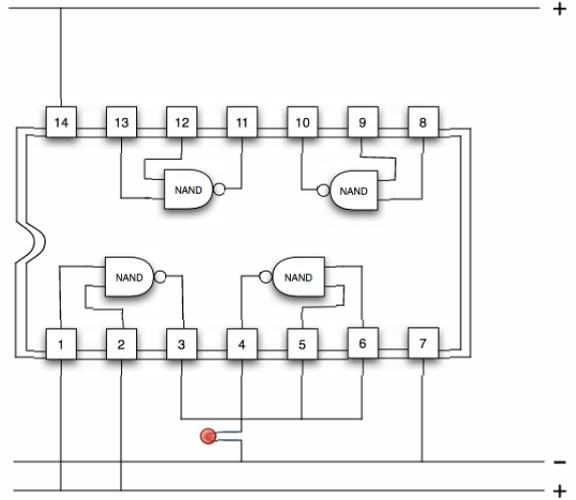


図 7: IC-AND

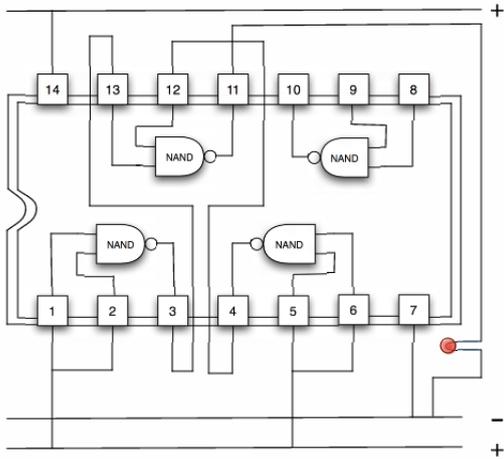


図 8: IC-OR

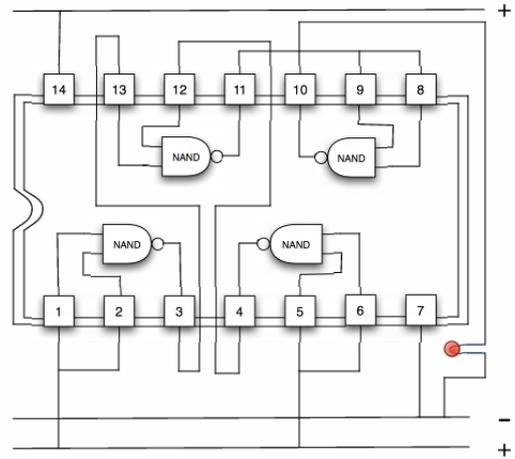


図 9: IC-NOR

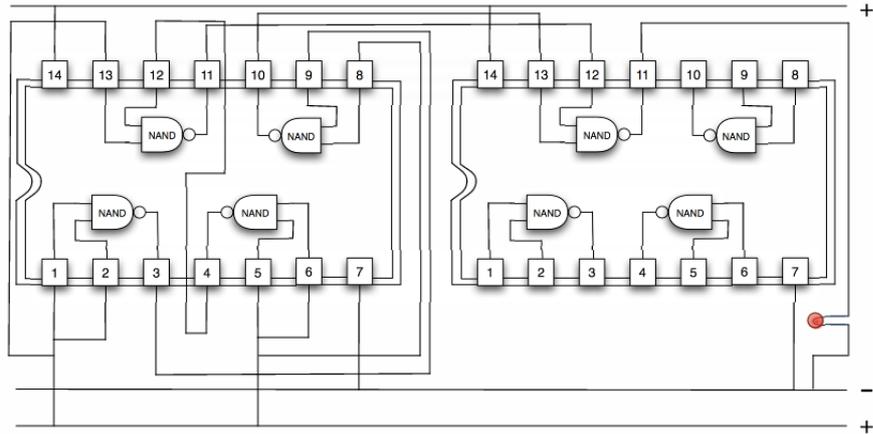


図 10: IC-XOR

上の図 6, 図 7, 図 8, 図 9, 図 10 のようにブレッドボードに接続した。それぞれ、NOT, AND, OR, NOR, XOR ゲートと等価な動作をした。

4 考察

4.1 実験(1)の考察について

上記の実験結果で報告した回路よりも、使用する NAND ゲートの数を減らすことが可能なのかを、NOT、AND、OR、NOR、XOR ゲートのそれぞれについて考察せよ。

1. NOT ゲートについて

(a) 使用しているゲートは1つなので減らす必要がない

2. AND ゲートについて

(a) 2つより減らすことができない

3. OR ゲート

(a) 3つより減らすことができない

4. NOR ゲート

(a) 4つより減らすことができない

5. XOR ゲート

(a) 図5を見ると、F1、F2を出力する NAND ゲートは論理反転するためのものである。

(b) NAND ゲートを1本にしているため、2つの NAND ゲートの入力が1本無駄である

(c) 2つの NAND を1つにすることで、NAND ゲートを4つに減らすことができる

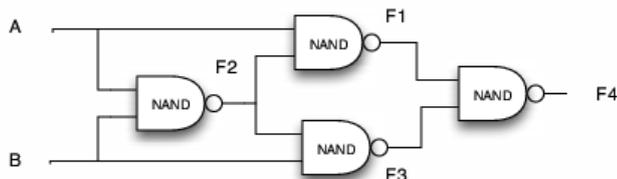


図 11: ゲート数4の XOR

(d) 図 11 から真理値表作成

表 8: 図 11 の真理値表

A	B	F1	F2	F3	F4
0	0	1	1	1	0
0	1	1	1	0	1
1	0	0	1	1	1
1	1	1	0	1	0

(e) XOR の真理値表と同じになった

(f) したがって、XOR ゲートは NAND ゲート4つでできる

4.2 実験（２）の考察について

4.2.1 配線数を減らすための工夫

1. 必要ない配線は取り除く

使わない電源配線部などに電気を流す必要がないので、配線を取り除く。電源配線部は必要最小限で使用する

2. ゲートの数を減らす

回路設計の時、ゲート数を最小にすることで、自然に配線を減らすことができる

-例-（回路のみの配線数）

図 5 の配線数は 11 本に対し、ゲート数を減らした図 11 は 9 本である

3. 使用するゲートの種類を増やす

使用するゲートの種類を増やすことにより、回路設計の時に最小の設計がしやすくなる

-例-（回路のみの配線数）

NAND ゲートを使って XOR ゲートを作った時の配線の数は最小 9 本であった。4030B のロジック IC の XOR ゲートをそのまま使用すると、入力 2 本と出力 1 本なので 3 本になる。

4.2.2 配線ミスを減らすための工夫

1. 配線をさす度に確認する

一番は大事なのは配線をちゃんとさすことである。なので、さす度に確認するのが効果的

2. さす場所を決める

部品配線部には IC をさす穴以外に 4 つの穴がある。その一つ一つに役割を割り当てるとミスが減り、確認も早くなる

4.3 その他の考察

AND ゲート、OR ゲート、NOT ゲートを使って完全系ができた。他にも完全系にはいろいろあったが、XOR ゲートだけや AND ゲート、OR ゲートの組合せなど完全系ではない組合せもある。その理由について考察する。

4.3.1 XOR ゲート

1. XOR ゲートだけでは完全系はできない
2. XOR ゲートでは NOT ゲートを再現できる
3. つまり XOR ゲートと NOT ゲートで AND,OR,NAND,NOR のいずれかが作ることができるなら完全系である

表 9: AND ゲートの真理値表

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

表 10: XOR ゲートの真理値表

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

表 11: NOT ゲートの真理値表

A	F
0	1
1	0

4. ここでは表 9 の AND ゲートについて考えてみる
5. 表 10 と表 11 の真理値表を組み合わせて考える
6. NOT ゲートは反転するゲートである
7. ここで表 10 を見ると出力「1」の数が偶数である
8. 表 9 の AND ゲートは出力「1」の数が奇数である
9. 偶数の XOR ゲートに反転する NOT ゲートをいくら組み合わせても「1」の数は偶数である
10. したがって、XOR ゲートと NOT ゲートでは AND ゲートは作れない
11. 同様に OR,NAND,NOR ゲートも「1」の数は奇数なので XOR ゲートは完全系ではない

4.3.2 AND ゲート、OR ゲート

1. AND ゲートに NOT ゲートを加えることができれば NAND ゲートとなり、完全系になる
2. つまり、AND ゲートまたは OR ゲートで NOT ゲートを再現できるなら完全系である

表 12: OR ゲートの真理値表

A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

3. ここでは表 12 の OR ゲートについて考える
4. 表 12 より OR ゲートはどちらかに「1」が入力されているのあら、出力が「1」となる。
5. 表 12 の入力 A が「0」の時、出力「1」にはできる
6. 表 12 の入力 A が「1」の時、出力「0」にはできない
7. つまり、入力「1」にした時、反転することができないのである
8. よって、OR ゲートでは NOT ゲートを作ることができない
9. 同様に AND ゲートについても考える
10. 表 9 の入力 A が「0」の時、出力「1」にはできない
11. つまり、AND ゲートも NOT ゲートを作ることができない
12. したがって、AND ゲートと OR ゲートは完全系ではない

5 調査課題

5.1 (a) 完全系

ある決まった論理演算の組を使用して任意の論理関数を表すことができるとき、そのような論理演算の組を完全系という。例えば、論理積否定 (NAND ゲート) だけを用いて任意の論理関数を表すことができるので、論理積否定は、それ一つだけで、完全系となっている。また、調査また、否定 (NOT)、論理積 (AND)、論理和 (OR) の組も完全系である。この他にもたくさんの完全系が存在する。これら以外の完全系を 2 組以上示し、それらが完全系である事を証明せよ。

1. NOT ゲートと AND ゲート

- (a) NAND ゲートは完全系である。
- (b) NAND ゲートを作ることができればそれは完全系であると言える
- (c) AND ゲート ($A \cdot B$) を否定したのが NAND ゲート ($\overline{A \cdot B}$) である



図 12: NOT,AND で NAND

- (d) したがって、NOT ゲートと AND ゲートは完全系である

2. NOT ゲートと OR ゲート

- (a) NAND ゲートを作ることができれば完全系である
- (b) 論理式を利用して証明する

ド・モルガンの法則より

$$(\overline{A \cdot B}) = (\overline{A} + \overline{B}) \quad (3)$$

- (c) つまり、A の否定と B の否定の論理和は NAND ゲートである

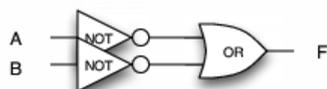


図 13: NOT,OR で NAND

- (d) したがって、NOT ゲートと OR ゲートは完全系である

3. NOR ゲート

- (a) NOT ゲートと OR ゲートは完全系である
- (b) NOT ゲートと OR ゲートを作ることができれば完全系である
- (c) OR ゲート $(A + B)$ の否定は NOR ゲート $\overline{(A + B)}$ である
- (d) よって、NOR ゲートを否定したらは OR ゲートができる

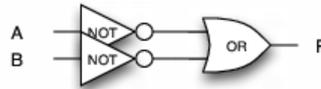


図 14: NOT,OR で NOR

- (e) つまり、NOR ゲートで NOT ゲートを作ることができれば完全系であると言える



図 15: NOR,NOT

- (f) 真理値表を使って証明する

表 13: 図 15 の真理値表

A	F
0	1
1	0

- (g) A で入力された値は F に出力される
- (h) 表 13 は NOT ゲートの真理表と同じである
- (i) NOR ゲートを使って NOT ゲートが作れた。したがって、NOR ゲートは完全系である

4. その他

- (a) XOR ゲートは完全系ではないが、XOR ゲートは NOT ゲートが作れる

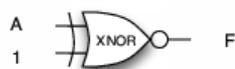


図 16: XOR で NOT

- (b) 真理値表を使って説明する

表 14: 図 16 の真理値表

A	1	F
0	1	1
1	1	0

- (c) 片方を常に「1」にする
(d) A と 1 の入力が F に出力される
(e) これは A,F 見ると NOT ゲートと同じ真理値表である
(f) つまり、XOR ゲートは NOT ゲートが作れる
(g) したがって、NOT ゲートと XOR ゲートを交換した完全系ができる
-例-
AND ゲートと XOR ゲート、AND ゲートと OR ゲートと XOR ゲート、... etc

5.2 (b) カルノー図

今回の実験で学んだように、論理関数の表現は一意ではなく複数（厳密には無数）ある。実際の回路を実現する場合は、それらの関数の中で、できるだけ簡単な表現を採用すべきである。そのような簡単な論理関数表現を求める際、カルノー図と呼ばれる図表が用いられることがある。カルノー図とはどのような図表か調査し報告せよ。また、カルノー図の使用方法を具体例を用いて説明せよ。

5.2.1 カルノー図とは

1. 論理式を簡略化することができる
2. 比較的楽にできる
3. 感覚的、視覚的な方式で簡略化できる
4. 2変数以上に使われる図である
5. 図の例

(a) 2変数の場合の図 17 の表と 4変数の場合の図 18 の表を作る

		A	
		0	1
B	0		
	1		

図 17: カルノー図 (2変数)

		AB			
		00	01	11	10
CD	00				
	01				
	11				
	10				

図 18: カルノー図 (4変数)

- (b) 図 17 のカルノー図は 2 : 2 の表である
- (c) 図 18 のカルノー図は 4 : 4 の表である
- (d) 数字の並びが 00,01,11,10 の順番である
- (e) このとき 11,10 を逆にしないように注意する

6. 論理関数を図 17 や図 18 を利用して簡略化することができる

5.2.2 具体例

$$f = \overline{A}B\overline{C}D + AB\overline{C}D + \overline{A}BCD \quad (4)$$

1. 論理式を数字にする

(a) $\overline{A}B\overline{C}D = 0101$

(b) $AB\overline{C}D = 1101$

(c) $\overline{ABCD} = 0111$

2. 図 18 に当てはまるセルに「1」を書く
3. そのあと2つ組で囲むと図 19 のようになる

		AB			
		00	01	11	10
CD	00				
	01		1	1	
	11		1		
	10				
	00				

図 19: カルノー図 (具体例)

4. 図 19 をまとめる
5. 組合せが 00,01 の場合は 0 だけとなる
6. まとめると以下の式ができる

$$f = B\overline{C}D + \overline{A}BD \quad (5)$$

7. カルノー図を使って論理関数を簡略化した

5.3 (c) 半加算器と全加算器

実用的なデジタル回路として半加算器および全加算器がある。これらの回路はどのような回路か調査し、それぞれの真理値表を示せ。また、各真理値表を用いて、それらの機能や特徴を説明せよ。さらに、それぞれの回路図の一例と、各回路図が実現している回路関数を示せ。

5.3.1 半加算器

1. 2 入力の 2 出力の論理回路である
2. 2 進数の同じ桁どうしを演算する
3. 半加算器の真理値表を書く

表 15: 半加算器の真理値表

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

4. 桁上がりは桁上げ出力 (Carry out) によって出力する。
5. 表 15 でわかるように、C は AND ゲート、S は XOR ゲートである

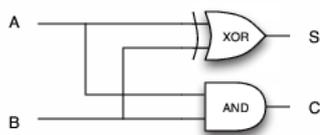


図 20: 半加算器の回路図

6. 図 20 を論理関数で表す

$$C = A \cdot B \quad (6)$$

$$S = (\bar{A} \cdot B) + (A \cdot \bar{B}) \quad (7)$$

5.3.2 全加算器

1. 3入力2出力の論理回路である
2. 2進数の最下位以外の同じ桁どうしの演算する
3. 下位からの桁上げ入力を含めて出力する。
4. 全加算器の真理値表を書く

表 16: 全加算器の真理値表

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

5. 表 16 より出力 C は A,B,X の多い方を出力している
6. つまり、多数決回路である
7. 表 16 より出力 S は A,B,X の奇数の数の値をを出力している
8. 全加算器は3つの入力を加算してるから、半加算器が2つと OR ゲートが必要である
9. 図 21 の回路図になる

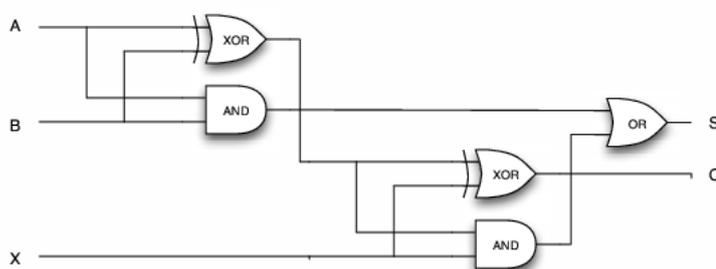


図 21: 全加算器の回路図

10. 表 16 を使って論理関数を求める

$$\begin{aligned}
 C &= \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC \\
 &= AB + BC + AC
 \end{aligned}
 \tag{8}$$

$$S = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC
 \tag{9}$$

6 感想

実験自体はとても楽しかったです。ブレットボードを言う初めてみる機器に触れ、いろいろ試したりすることができて良かったです。しかし、授業の説明がとても短すぎです。一体何をするのか全くわかりませんでした。自分で考えて結果的に出来ました。先生の話がほとんど意味がなかったように感じられました。でも、吉田先生の考えでは実験だけ自分で考えてやって欲しいのではないかと思います。しかし、いくら何でも少なすぎです。もっと説明して欲しかったです。

今回のレポートは今まで一番めんどくさかったです。いろいろな書き方や規制がかかっていて、とても時間がかかりました。でも、これはこれで4年次になった時に役に立つので、とても良かったと思います。しかし、調査課題の(b),(c)はどういう意味があったのか全く理解できません。(a)は完全系という言葉は初めて聞いたことなので、楽しかったです。(b),(c)に関しては1年次の復習です。たしかに、1年次の復習は大事ですが、この課題で出す必要があったのか疑問に思います。

今回のレポートで一番問題だったのは考察です。今までの考察は、自分で考え、それが正しいか実験し、なぜダメだったのか、なぜ成功したのかを考察してきました。しかし、今回のレポートでは確かめる方法が調べることができません。実際に回路を繋げたりすることができません。実験の良いところは、実際にやって、それについて考えることだと思います。しかし、この考察では考えても、図で書いたり、数式にすることしかできません。実際に回路に線を繋げたほうが良いアイデアが生まれたりするのではないのでしょうか？配線ミスが減らす工夫は何度も配線ミスをして、それを学習して工夫に工夫を重ねて、この考察がかけると思います。なので、もっと機器に触れる時間があれば良いと思いました。

参考文献

図の挿入

<http://hooktail.org/computer/index.php?%BF%DE%A4%CE%C1%DE%C6%FE>

NAND × 4ヶで、EXOR と等価な回路

<http://tamuro.gooside.com/guen/NandExor.html>

論理回路

<http://homepage2.nifty.com/ttoyoshima/Computer/Logic.htm>

加算器

<http://ja.wikipedia.org/wiki/加算器>