

基本ゲート回路

035760A 横田敏明

2004/05/30

1 実験状況

1.1 共同実験者

村山正嗣

宮城大輔

1.2 実験器具シリアルナンバー

IC : TC4011BP 電源 : 04535

2 目的

現代社会に欠かすことのできないコンピュータは、大規模なデジタル回路によって構成されている。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。

3 実験

- NAND ゲートのみを用いて、NOT,AND,OR,NOR,XOR ゲートを設計せよ。
- 上の実験で設計した各ゲートを実際に NAND ゲート IC を用いて実現し、それらの動作を確認せよ。

4 報告

4.1 ゲートの設計

実験中に設計したゲート設計を以下に記す。IC による実験も、成功した。IC に関しては難易度の高い XOR のみ記す。

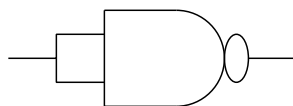


図 1: NOT 回路

NOT 真理値表

入力	出力
0	1
1	0

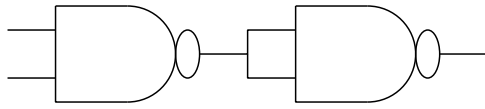


图 2: AND 回路

AND 真理值表

入力 1	入力 2	出力
0	0	0
0	1	0
1	0	0
1	1	1

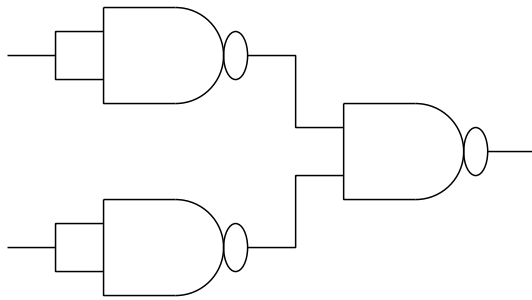


图 3: OR 回路

OR 真理值表

入力 1	入力 2	出力
0	0	0
0	1	1
1	0	1
1	1	1

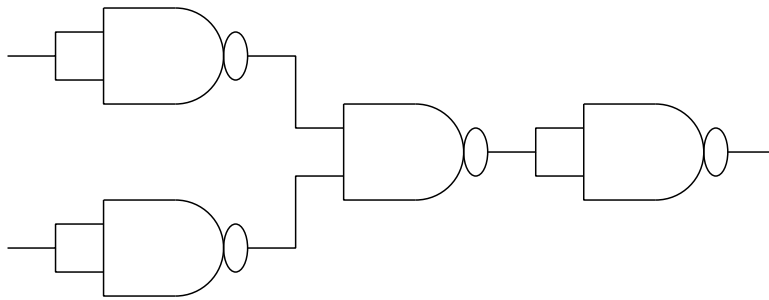


图 4: NOR 回路

NOR 真理值表

入力 1	入力 2	出力
0	0	1
0	1	0
1	0	0
1	1	0

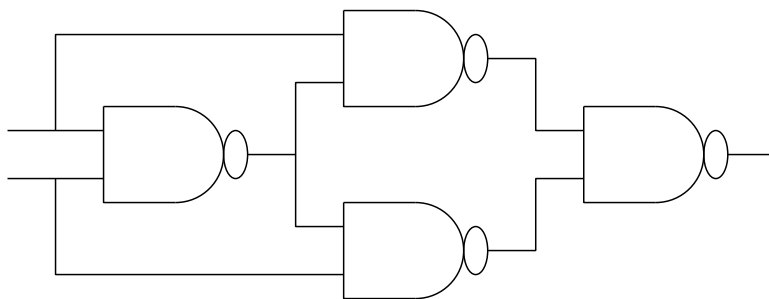


图 5: XOR 回路

XOR 真理值表

入力 1	入力 2	出力
0	0	0
0	1	1
1	0	1
1	1	0

4.2 2変数の論理関数

2変数の論理関数がなぜ16種類あるのか。

出力だけで考えると「0」と「1」の二種類あり，入力の組合せは4種類である．よって，入力と出力の全ての組合せは $2^4 = 16$ となる．

すべての論理関数の出力のみを以下に列挙する．入力は(A,B)とし，(00,01,10,11)の順

- $(0,0,0,0) = \overline{A + \bar{A} + B + \bar{B}}$
- $(0,0,0,1) = A \cdot B$
- $(0,0,1,0) = A \cdot \bar{B}$
- $(0,0,1,1) = A$
- $(0,1,0,0) = \bar{A} \cdot B$
- $(0,1,0,1) = B$
- $(0,1,1,0) = (A \cdot \bar{B}) + (\bar{A} \cdot B)$
- $(0,1,1,1) = (A + B)$
- $(1,0,0,0) = \overline{(\bar{A} + \bar{B})}$
- $(1,0,0,1) = (A \cdot B) + (\bar{A} \cdot \bar{B})$
- $(1,0,1,0) = \bar{B}$
- $(1,0,1,1) = \bar{A} \cdot \bar{B}$
- $(1,1,0,0) = \bar{A}$
- $(1,1,0,1) = \overline{A \cdot \bar{B}}$
- $(1,1,1,0) = \overline{A \cdot B}$
- $(1,1,1,1) = A + \bar{A} + B + \bar{B}$

4.3 1種類使用のゲート回路

NANDゲートオンリー回路のように，1種類だけでNOT,AND,OR,NAND,NOR,XORゲートを表せるゲート回路の，具体例を示す．NOR回路で実現できる．

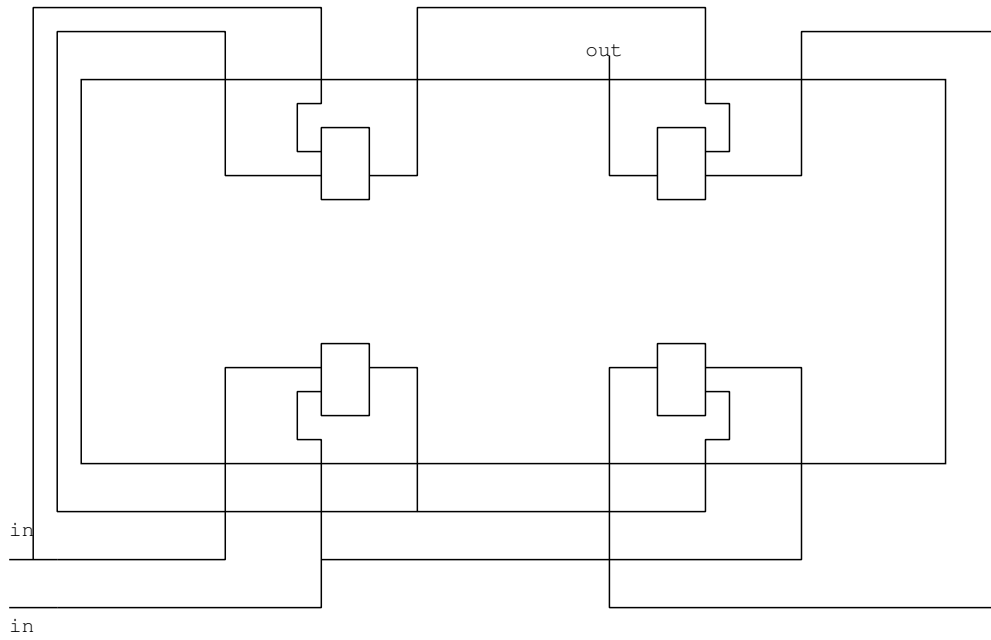


図 6: IC による XOR 設計

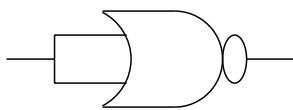


図 7: NOT 回路

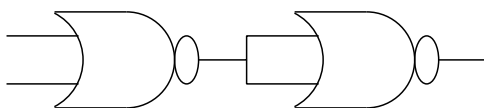


図 8: OR 回路

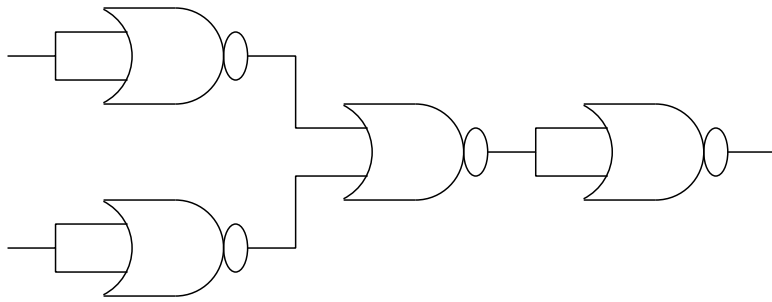


图 9: NAND 回路

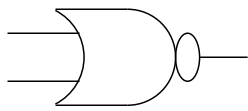


图 10: NOR 回路

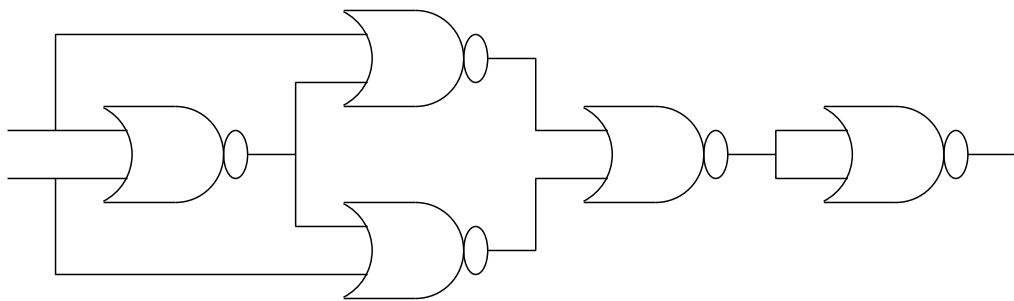


图 11: XOR 回路

4.4 2種類のゲート回路

2種類のゲート回路を用いて各ゲートを設計せよ。以下に具体例を示す。

4.4.1 ANDとNOTの組合せ・ORとNOTの組み合わせ

NANDのみで回路設計ができるのならば、ANDとNOTを組み合わせれば、全て作ることができる。同様に、ORとNOTの組み合わせもできる。

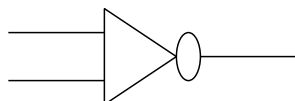


図 12: NOT 回路



図 13: AND 回路

2つのパーツに分けて書くだけなので、回路図は前記の回路と同じである。よって回路図は省略する。

4.5 半加算器および全加算器

半加算器とは、2進数での加算をする回路である。2入力、2出力である。2つの出力はそれぞれ繰り上がりと加算結果である。加算結果を S 、繰り上がりの結果を C とすると、以下のような真理値表となる。

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

$$C = A \text{ and } B$$

$$S = A \text{ xor } B$$

つまり、半加算器は以下のような構造であると推測される。そして、全加

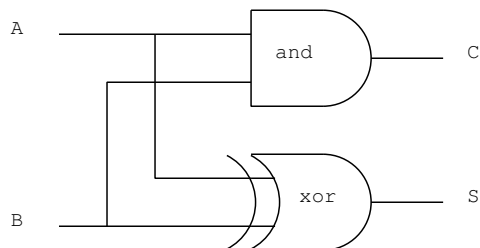


図 14: 半加算器

算器とは 3 入力 2 出力のゲート回路である。繰り上がり入力を入力 C とおくと、以下のような真理値表をとる。

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

全加算器は次のように表せる。

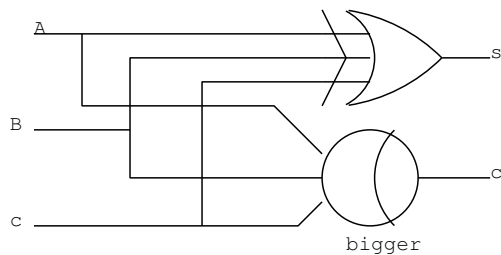


図 15: 全加算器

下の丸い論理素子は多数決で入力を選ぶしきい論理素子である。

5 考察

今回の実験では、論理回路の理解を深め、自ら設計できる知識と能力を身につけることができた。

論理回路の設計にあたっては、真理値表を使用することが大切だとわかった。1種類だけのゲートを用いて各ゲートを設計する場合のコツは、まずはNOTが容易に作れるか、が重要であると思われる。NOT回路が簡単に作れるのはNORやNANDである。そして、NOT回路が加われば自由度が爆発的に増え、ほとんどの回路を設計することができるとわかった。ICを用いた設計では、数が制限されるため、無駄の無い最小限の設計を余儀なくされる。また、コストダウンにつながるので難しいとって何も考えずに設計することは好ましくないと理解した。近代のチップは大規模集積化が進み、VLSIの設計は人間には不可能である。結局は設計用のコンピュータ言語で設計するため、今では人自らが設計することはないが、今回の実験は得られるものがあった。