情報工学実験1

「基本ゲート回路」

実施日：2010 年 7 月 6 日

学籍番号：	０９５７０７Ｂ
氏　　名：	大 城 佳 明
締 切 日：	2010 年 7 月 13 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

本実験では簡単な組合せ回路の設計および実現を行うことによって、カルノー図などを用いた論理関数の簡単化に慣れるとともに、実際のコンピュータに使用される演算器の設定法について習得する

2　実験概要

まず、４ビットインクリメンタを真理値表で書く。そこから論理関数を求める。真理値表から論理関数の求め方がわかるようになり、簡単化するための準備となる。次に、カルノー図を使い、簡単化をする。３ビットインクリメンタをカルノー図で簡単化し、実装する。そうすることで、実際にカルノー図で求めた論理関数が正しいかを学ぶことができる。カルノー図だけではなく、論理関数からさらにド・モルガンなどを使い、さらに簡単化を行う。さらに簡単化されることで、ブレットボードへの実装が楽になる。そうすることで、簡単化の必要性を学ぶことができる。加算器、３ビットインクリメンタ、４ビットインクリメンタを作ることにより、コンピュータに使用されている演算器の設定法について習得することができる。

3　実験結果

3.1　実験 (1),(2) にの結果について

全加算器、３ビットインクリメンタ、４ビットインクリメンタの各真理値表を示せ。
各真理値表から求めた論理関数を全て示せ。

3.1.1　全加算器

1. 全加算器の真理値表である

表 1: 全加算器の真理値表

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

2. 表??からの論理関数

$$S = \overline{A}\cdot\overline{B}\cdot C_{in} + \overline{A}\cdot B\cdot\overline{C_{in}} + A\cdot\overline{B}\cdot\overline{C_{in}} + A\cdot B\cdot C_{in} \quad (1)$$

$$C_{out} = \overline{A}\cdot B\cdot C_{in} + A\cdot\overline{B}\cdot C_{in} + A\cdot B\cdot\overline{C_{in}} + A\cdot B\cdot C_{in} \quad (2)$$

3.1.2　３ビットインクリメンタ

1. ３ビットインクリメンタの真理値表である

表 2: ３ビットインクリメンタの真理値表

x_1	x_2	x_0	y_2	y_1	y_0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

2. 表**??**からの論理関数

$$y_0 = \overline{x_2}\cdot\overline{x_1}\cdot\overline{x_0}+\overline{x_2}\cdot x_1\cdot\overline{x_0}+x_2\cdot\overline{x_1}\cdot\overline{x_0}+x_2\cdot x_1\cdot\overline{x_0} \tag{3}$$

$$y_1 = \overline{x_2}\cdot\overline{x_1}\cdot x_0+\overline{x_2}\cdot x_1\cdot\overline{x_0}+x_2\cdot\overline{x_1}\cdot x_0+x_2\cdot x_1\cdot\overline{x_0} \tag{4}$$

$$y_2 = \overline{x_2}\cdot x_1\cdot x_0+x_2\cdot\overline{x_1}\cdot\overline{x_0}+x_2\cdot\overline{x_1}\cdot x_0+x_2\cdot x_1\cdot\overline{x_0} \tag{5}$$

3.2　４ビットインクリメンタ

1. 表**??**は４ビットインクリメンタの真理値表である

2. 表**??**の論理関数

$$\begin{aligned} f_1 &= \overline{A}\cdot B\cdot C\cdot D+A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+A\cdot\overline{B}\cdot\overline{C}\cdot D+A\cdot\overline{B}\cdot C\cdot\overline{D} \\ &+ A\cdot\overline{B}\cdot C\cdot D+A\cdot B\cdot\overline{C}\cdot\overline{D}+A\cdot B\cdot\overline{C}\cdot D+A\cdot B\cdot C\cdot\overline{D} \end{aligned} \tag{6}$$

$$\begin{aligned} f_2 &= \overline{A}\cdot\overline{B}\cdot C\cdot D+\overline{A}\cdot B\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot B\cdot\overline{C}\cdot D+\overline{A}\cdot B\cdot C\cdot\overline{D} \\ &+ A\cdot\overline{B}\cdot C\cdot D+A\cdot B\cdot\overline{C}\cdot\overline{D}+A\cdot B\cdot\overline{C}\cdot D+A\cdot B\cdot C\cdot\overline{D} \end{aligned} \tag{7}$$

$$\begin{aligned} f_3 &= \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot D+\overline{A}\cdot\overline{B}\cdot C\cdot\overline{D}+\overline{A}\cdot B\cdot\overline{C}\cdot D+\overline{A}\cdot B\cdot C\cdot\overline{D} \\ &+ A\cdot\overline{B}\cdot\overline{C}\cdot D+A\cdot\overline{B}\cdot C\cdot\overline{D}+A\cdot B\cdot\overline{C}\cdot D+A\cdot B\cdot C\cdot\overline{D} \end{aligned} \tag{8}$$

$$\begin{aligned} f_4 &= \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot\overline{B}\cdot C\cdot\overline{D}+\overline{A}\cdot B\cdot\overline{C}\cdot\overline{D}+\overline{A}\cdot B\cdot C\cdot\overline{D} \\ &+ A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+A\cdot\overline{B}\cdot C\cdot\overline{D}+A\cdot B\cdot\overline{C}\cdot\overline{D}+A\cdot B\cdot C\cdot\overline{D} \end{aligned} \tag{9}$$

表 3: 4ビットインクリメンタの真理値表

A	B	C	D	f_1	f_2	f_3	f_4
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

3.3 実験 (3) の結果について

実験 (3) で得られた簡単化後の論理関すを全て示せ。なお、3ビットインクリメンタに関してはカルノー図と簡単化の過程（規則性適用の様子）も示すこと。

3.3.1 全加算器

$$
\begin{aligned}
C_{out} &= \overline{A}BC_{in} + A\overline{B}C_{in} + AB\overline{C}_{in} + ABC_{in} \\
&= AB + BC_{in} + AC_{in} \qquad (10) \\
S &= \overline{AB}C_{in} + \overline{A}B\overline{C_{in}} + A\overline{BC_{in}} + ABC_{in} \qquad (11)
\end{aligned}
$$

3.3.2 3ビットインクリメンタ

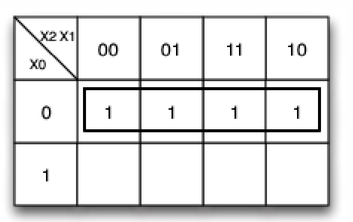

図 1: カルノー図 (3ビットインクリメンタ y_0)

$$
\begin{aligned}
y_0 &= (\overline{x_2}\cdot\overline{x_1}\cdot\overline{x_0}+\overline{x_2}\cdot x_1\cdot\overline{x_0}+x_2\cdot x_1\cdot\overline{x_0}+x_2\cdot\overline{x_1}\cdot\overline{x_0}) \\
&= \overline{x_2}\cdot\overline{x_0}+x_2\cdot\overline{x_0} \\
&= \overline{x_0} \qquad (12)
\end{aligned}
$$

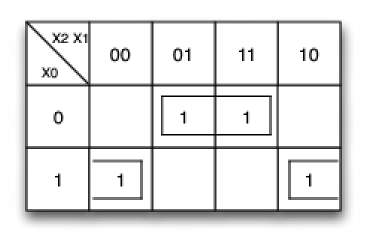

図 2: カルノー図 (3 ビットインクリメンタ y_1)

$$
\begin{aligned}
y_1 &= (\overline{x_2}\cdot x_1\cdot\overline{x_0}+x_2\cdot x_1\cdot\overline{x_0})+(\overline{x_2}\cdot\overline{x_1}\cdot x_0+x_2\cdot\overline{x_1}\cdot x_0) \\
&= x_1\cdot\overline{x_0}+\overline{x_1}\cdot x_0 = x_1\oplus x_2 \qquad (13)
\end{aligned}
$$

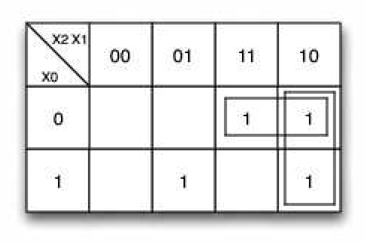

図 3: カルノー図 (3 ビットインクリメンタ y_2)

$$
\begin{aligned}
y_2 &= (x_2\cdot x_1\cdot\overline{x_0}+x_2\cdot\overline{x_1}\cdot\overline{x_0})+(x_2\cdot\overline{x_1}\cdot\overline{x_0}+x_2\cdot\overline{x_1}\cdot x_0)+\overline{x_2}\cdot x_1\cdot x_0 \\
&= x_2\cdot\overline{x_0}+x_2\cdot\overline{x_1}+\overline{x_2}\cdot x_1\cdot x_0 \\
&= x_2\cdot(\overline{x_0}+\overline{x_1})+\overline{x_2}\cdot x_1\cdot x_0 \\
&= x_2\cdot(\overline{\overline{\overline{x_0}+\overline{x_1}}})+\overline{x_2}\cdot x_1\cdot x_0 \\
&= x_2\cdot(\overline{x_1\cdot x_0})+\overline{x_2}\cdot(x_1\cdot x_0) = x_2\oplus(x_1\cdot x_2) \qquad (14)
\end{aligned}
$$

3.3.3 4 ビットインクリメンタ

$$
\begin{aligned}
f_1 &= (A\cdot B\cdot\overline{C}\cdot\overline{D}+A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+A\cdot B\cdot C\cdot\overline{D}+A\cdot\overline{B}\cdot C\cdot\overline{D}) \\
&+ (A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D}+A\cdot\overline{B}\cdot\overline{C}\cdot D+A\cdot\overline{B}\cdot C\cdot D+A\cdot\overline{B}\cdot C\cdot\overline{D}) \\
&+ A\cdot\overline{C}+\overline{A}\cdot B\cdot C\cdot D
\end{aligned}
$$

$$
\begin{aligned}
&= A \cdot \overline{B} + A \cdot \overline{C} + A \cdot \overline{D} + \overline{A} \cdot B \cdot C \cdot D \\
&= A(\overline{B} + \overline{C} + \overline{D}) + \overline{A} \cdot B \cdot C \cdot D \\
&= A(\overline{\overline{\overline{B} + \overline{C} + \overline{D}}}) + \overline{A} \cdot B \cdot C \cdot D \\
&= A(\overline{B \cdot C \cdot D}) + \overline{A} \cdot (B \cdot C \cdot D) \\
&= A \oplus (B \cdot C \cdot D)
\end{aligned}
\tag{15}
$$

$$
\begin{aligned}
f_2 &= (\overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} \cdot D + A \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot D) \\
&\quad + (\overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot C \cdot \overline{D} + A \cdot B \cdot C \cdot \overline{D}) \\
&\quad + (\overline{A} \cdot \overline{B} \cdot C \cdot D + A \cdot \overline{B} \cdot C \cdot D) \\
&= B \cdot \overline{C} + B \cdot \overline{D} + \overline{B} \cdot C \cdot D \\
&= B(\overline{C} + \overline{D}) + \overline{B} \cdot C \cdot D \\
&= B(\overline{\overline{\overline{C} + \overline{D}}}) + \overline{B} \cdot C \cdot D \\
&= B(\overline{C \cdot D}) + \overline{B} \cdot (C \cdot D) \\
&= B \oplus (C \cdot D)
\end{aligned}
\tag{16}
$$

$$
\begin{aligned}
f_3 &= (\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot B \cdot \overline{C} \cdot D + A \cdot \overline{B} \cdot \overline{C} \cdot D + A \cdot B \cdot \overline{C} \cdot D) \\
&\quad + (\overline{A} \cdot B \cdot C \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + A \cdot \overline{B} \cdot C \cdot \overline{D} + A \cdot B \cdot C \cdot \overline{D}) \\
&= \overline{C} \cdot D + C \cdot \overline{D} \\
&= C \oplus D
\end{aligned}
\tag{17}
$$

$$
\begin{aligned}
f_4 &= (\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}) \\
&\quad + (\overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + \overline{A} \cdot B \cdot C \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D}) \\
&= (\overline{A} \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{C} \cdot \overline{D}) + (\overline{A} \cdot C \cdot \overline{D} + A \cdot \overline{C} \cdot \overline{D}) \\
&= \overline{C} \cdot \overline{D} + C \cdot \overline{D} \\
&= \overline{D}
\end{aligned}
\tag{18}
$$

3.4　実験 (4) の結果について

簡単化後に、各論理関数にどのような変化があったか説明せよ

1. 全加算器の C_{out} はカルノー図で簡単化できた
2. 全加算器の S はカルノー図では簡単化できない
3. ３ビットインクリメンタ、４ビットインクリメンタはカルノー図で簡単化できた
4. さらにそこからさらに簡単化できた
5. 簡単化することにより、論理演算子の数を減らした
6. 簡単化することにより、項の数が減らした
7. 論理演算子と項の数を減らすと回路図が簡単にできる

3.5　実験 (5) の結果について

３ビットインクリメンタの回路設計を示せ

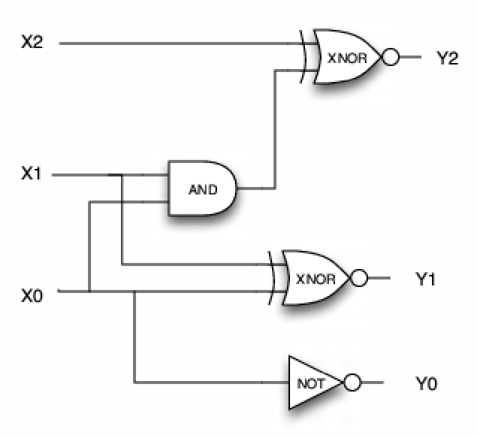

図 4: ３ビットインクリメンタ（回路図）

3.6 実験 (6) の結果について

ブレッドボード上に実現した回路が、３ビットインクリメンタと等価な動作をしたか否かを報告せよ

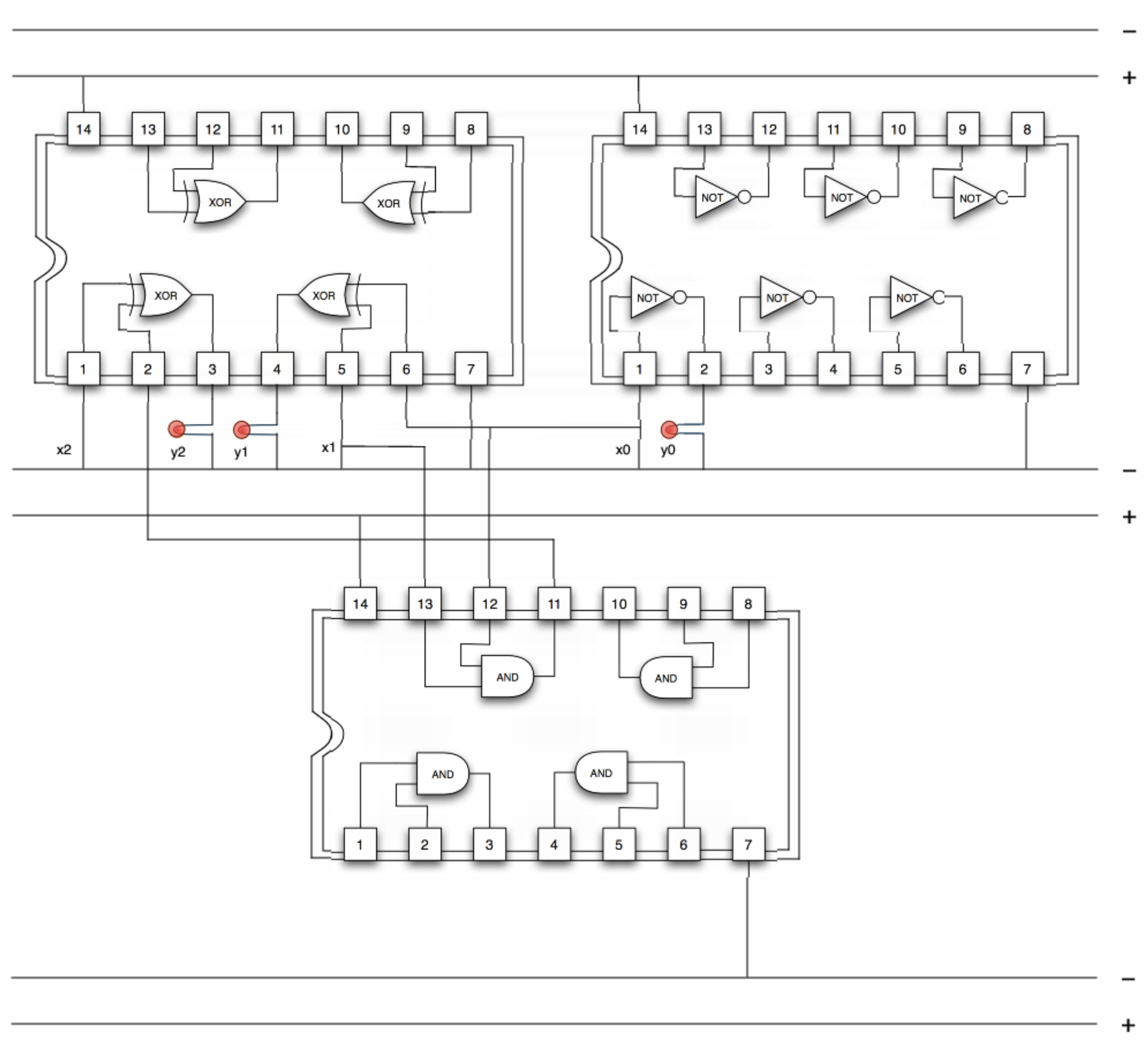

図 5: ３ビットインクリメンタ（ブレッドボード上）

図??のように設置した結果、実現した。

4　考察

4.1　実験 (3) の考察について

上記の実験結果で報告した論理関数よりも、使用する論理演算子の数を減らすことが可能かどうかを、全加算器、３ビットインクリメンタ、４ビットインクリメンタのそれぞれについて考察せよ。

4.1.1　全加算器

$$\begin{aligned} C_{out} &= AB + BC_{in} + AC_{in} && (19) \\ S &= \overline{A}\cdot\overline{B}\cdot C_{in} + \overline{A}\cdot B\cdot\overline{C_{in}} + A\cdot\overline{B}\cdot\overline{C_{in}} + A\cdot B\cdot C_{in} \\ &= A\cdot(\overline{B}\cdot\overline{C_{in}} + B\cdot C_{in}) + \overline{A}\cdot(\overline{B}\cdot C_{in} + B\cdot\overline{C_{in}}) \\ &= A\cdot(\overline{B\oplus C_{in}}) + \overline{A}\cdot(B\oplus C_{in}) \\ &= A\oplus(B\oplus C_{in}) && (20) \end{aligned}$$

1. C_{out} は簡単にすることはできない
2. S は排他的論理和を使用することができた
3. これ以上簡単にすることが出来ない

4.1.2　３ビットインクリメンタ

$$\begin{aligned} y_0 &= \overline{x_0} && (21) \\ y_1 &= x_1 \oplus x_2 && (22) \\ y_2 &= x_2 \oplus (x_1 \cdot x_2) && (23) \end{aligned}$$

1. y_0 は１つしかないので必要ない
2. y_1 は排他的論理和を使用している
3. y_2 も排他的論理和を使用している
4. これ以上簡単にすることは出来ない

4.1.3　４ビットインクリメンタ

$$\begin{aligned} f_1 &= A \oplus (B\cdot C\cdot D) && (24) \\ f_2 &= B \oplus (C\cdot D) && (25) \\ f_3 &= C \oplus D && (26) \\ f_4 &= \overline{D} && (27) \end{aligned}$$

1. f_4 は１つしかないため必要がない
2. f_1, f_2, f_3 とも排他的論理和を利用している
3. これ以上簡単化できない

4.2 実験 (4) の考察について

カルノー図を使って論理圧縮する場合、簡単化されやすい論理関数と簡単化されにくい論理関数がある。それらの違いにについて考察せよ。また、各自の考え（考察結果）に基づいて、全加算器、３ビットインクリメンタ、４ビットインクリメンタのそれぞれの論理関数が、簡単化され易い論理関数と簡単化されにくい論理関数のどちらに分類されるか示せ。

1. 簡単化しにくいのはどんなものがあるか
   (a) 1 が隣り合わず、独立している場合は上記の隣接するグループが作れないの
   (b) 隣り合わないといけない
   (c) ハミング距離が 1 となるのがない
2. 加算器、３ビットインクリメンタ、４ビットインクリメンタで簡単化されにくい関数
   (a) 全加算器 S の出力はハミング距離が１となるのがない
   (b) ３ビットインクリメンタの y_2 はハミング距離が１でないものが１つだけある
   (c) ４ビットインクリメンタの f_1 もハミング距離が１ではないのが１つだけある
3. 以上が簡単化されにくい関数である

4.3 その他の考察について

５ビットインクリメンタ、６ビットインクリメンタの論理関数を求める

1. ３ビットインクリメンタ、４ビットインクリメンタから予測する
2. ３ビットインクリメンタの論理関数

$$\begin{aligned} y_0 &= \overline{x_0} \\ y_1 &= x_1 \oplus x_2 \\ y_2 &= x_2 \oplus (x_1 \cdot x_2) \end{aligned}$$

3. ４ビットインクリメンタの論理関数

$$\begin{aligned} f_4 &= \overline{D} \\ f_3 &= C \oplus D \\ f_2 &= B \oplus (C \cdot D) \\ f_1 &= A \oplus (B \cdot C \cdot D) \end{aligned}$$

4. パターンがあることがわかる

表 4: ４ビットインクリメンタの真理値表

A	B	C	D	E	f_1	f_2	f_3	f_4	f_5
0	0	0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0	1	0

5. 表??のような５ビットインクリメンタの真理値表があるとする

6. これより、以下の式が出来る

$$\begin{aligned} f_5 &= \overline{E} \\ f_4 &= D \oplus E \\ f_3 &= C \oplus (E \cdot E) \\ f_2 &= B \oplus (C \cdot D \cdot E) \\ f_1 &= A \oplus (B \cdot C \cdot D \cdot E) \end{aligned}$$

7. 同様にさらに６ビットインクリメンタを求めると、

$$\begin{aligned} f_6 &= \overline{F} \\ f_5 &= E \oplus F \\ f_4 &= D \oplus (E \cdot F) \\ f_3 &= C \oplus (D \cdot E \cdot F) \\ f_2 &= B \oplus (C \cdot D \cdot E \cdot F) \\ f_1 &= A \oplus (B \cdot C \cdot D \cdot E \cdot F) \end{aligned}$$

8. 以上のようになると思われる

5 調査課題

5.1 CLA

加算器には、今回の実験で学んだもの以外にも様々な加算器がある。その中で、高速な加算器として用いられるものに、桁上げ先見加算器（carry look-ahead adder : CLA）と呼ばれる加算器がある。CLA とは堂のような加算器か調査し、報告せよ。

1. CLA 加算器について
   (a) 通常の加算器の出力には、キャリー (桁上げ) を次の桁に伝える伝播部分がある。
   (b) 伝播部分があるため、下の桁の処理が終了しなければ次の桁の処理は開始する事が出来ない
   (c) CLA を論理関数で表す

$$Gn = An \cdot Bn (\text{生成項}) \tag{28}$$
$$Qn = An \oplus Bn (\text{伝播項}) \tag{29}$$
$$Cn = Gn + Qn \cdot C_{n-1} \tag{30}$$

   (d) CLA は下の桁の処理結果を元に処理するのではない
   (e) 全ての桁の計算を同時に行う事ができる
   (f) 入力から桁上がりを出し計算する
   (g) よって、計算時間の短縮化をはかる事ができる。
2. メリット
   (a) 通常の加算器よりも高速に 処理を行う事のできる
   (b) 桁数に比例した演算時間がかかることはない
3. デメリット
   (a) CLA は桁数が大きくなるにつれてキャリーを求める論理関数が複雑になる
   (b) 回路が大規模になってしまう

5.2 クワインーマクラスキー法

本実験で学んだカルノー図による論理圧縮は、直感的にわかり易い方法である。５変数程度までの論理関数にしか適用できず、あまり実用的ではない。このため実際には、クワインーマクラスキー法、あるいはクワインーマクラスキー法に基づいた論理圧縮手法が用いられている。クワインーマクラスキー法とは、どうのような方法か調査し報告せよ。また実際に４ビットインクリメンタの論理関数をクワインーマクラスキー法を用いて簡単化し、カルノー図を用いて簡単書いた場合の結果と比較せよ。

1. クワインーマクラスキー法について
   (a) ブール関数を簡単んかするための手法
   (b) コンピュータによる自動化に適している
   (c) ブール関数が最簡形かどうか決定的に求めることができる
   (d) ハミング距離が１のもの同士を集め、式を圧縮していき、これ以上圧縮出来ないものの和を最終的な論理関数とする
   (e) 方法は以下の通りである
      i. 関数の主項を全て求める
      ii. 求めた主項を表にまとめ、必須項を求める
      iii. 最簡形を求める
2. ４ビットインクリメンタの論理回路をクワインーマクラスキー法で簡単化する
   (a) 表**??**の f_1 を求める
   (b) 表**??**より、以下の表ができる。このときビット列表現をそれぞれ 1〜8 に置き換える
   (c) ハミング距離が 1 である組み同士で圧縮を行う
   (d) それ以上まとめることができない項には「*」をつける。
   (e) 「*」を付けた項が主項となる。
   (f) 表**??**、表**??**のようになる

表 5: f_1 の最小項まとめ

１の数	ビット列表現	置き換えた数
1	1000	1
2	1001	2
	1010	3
	1100	4
3	0111*	5
	1011	6
	1110	7
	1101	8

表 6: f_1 の１次圧縮

１の数	置き換えた数	ビット列表現
1	1,2	100-
	1,3	10-0
	1,4	1-00
2	2,6	10-1
	2,8	1-01
	3,6	101-
	3,7	1-01
	4,7	11-0
	4,8	110-

(g) 圧縮できなくなるまで何度も繰り返す

表 7: f_1 の２次圧縮

置き換えた数	ビット列表現
1,2,3,6*	10–
1,2,4,8*	1-0-
1,3,4,7*	1–0

(h) 求めた主項を表にまとめ、必須項を求める

(i) $e_0 = 1, 2$、$e_1 = 1, 2, 3, 6$、$e_2 = 1, 2, 4, 8$、$e_3 = 1, 3, 4, 7$ と置く

(j) 必須項には「*」をつける

表 8: f_1 の必須項

		1	2	3	4	5	6	7	8
e_0	5*					X			
e_1	1,2,3,6*	X	X	X			X		
e_2	1,2,4,8*	X	X		X				X
e_3	1,3,4,7*	X		X	X			X	

(k) 必須項は簡単化した関数に必ず含まれていなければならない項である

(l) 最簡形をもとめる

(m) 図??より求める

$$(e_1 + e_2 + e_3)(e_1 + e_2)(e_1 + e_3)(e_2 + e_3)e_0e_1e_2e_3$$

(n) これをまとめる

$$e_0e_1e_2e_3$$

(o) 最も簡単な形は $e_0e_1e_2e_3$

$$
\begin{aligned}
f_1 &= e_0 + e_1 + e_2 + e_3 \\
&= \overline{A}BCD + A\overline{B} + A\overline{C} + A\overline{D}
\end{aligned}
$$

(p) 最簡形をもとめることができた

(q) 同様に f_2, f_3, f_4 も求めた

$$
\begin{aligned}
f_2 &= B \cdot \overline{C} + B \cdot \overline{D} + \overline{B} \cdot C \cdot D \\
f_3 &= \overline{C} \cdot D + C \cdot \overline{D} \\
f_4 &= \overline{D}
\end{aligned}
$$

(r) カルノー図を用いて簡単化した場合と比べると、明らかに簡単になっている

(s) これはカルノー図を用いたあと、さらにド・モルガンなどを利用して、簡単化した形と同じである（加標準形）

(t) 以上のことより、クワインーマクラスキー法を使用した方が簡単化できることがわかる

5.3 組合せ回路、順序回路

ディジタル回路は、組合せ回路と順序回路に大別される。両者の違いを調査し説明せよ。

1. 組合せ回路
   - (a) 入力のみで出力が決まる回路
   - (b) AND ゲート、OR ゲート、NOT ゲート、XOR ゲートなどの基本となる論理演算を使用する
   - (c) 例：加算器、エンコーダ、デコーダなど
2. 順序回路
   - (a) 過去の内部状態と取得時の入力信号とで出力が決まる回路
   - (b) つまり現在と過去の入力 2 つの値によって決まる
   - (c) 内部に回路の状態を保持する素子がある
   - (d) 入力の組み合わせだけで出力が一意に決まる
   - (e) 複雑な制御などを行うことができる
   - (f) 例：フリップフロップ、カウンタなど

5.4 同期式、非同期式

順路回路は、同期式と非同期式に大別される。これらの違いについて調査し説明せよ。また、実際のディジタル回路のほとんどは同期式順序回路である。この理由について考察せよ。

1. 同期式の回路
   (a) クロック信号の立ち上がり、または立ち下がりに合わせて回路全体が動作するものである
   (b) その時点 (クロックが立ち上がるか立ち下がる) での入力に対応する信号を出力する。
   (c) 出力は次にクロックが立ち上がる（又は立ち下る）まで更新されない。
   (d) 同期式の回路の方が設計が簡単で扱いやすい
2. 非同期式の回路
   (a) クロック信号 を用いない回路である。
   (b) その時点での入力の値に応じて出力が変化する
   (c) 新たな入力信号が来るまで同じ状態を保持する
3. 現在のデジタル回路では同期式の回路が主に使われている理由
   (a) 非同期回路は組みにくい
   (b) 段階が多くなるので、遅れが大きくなり、高速には向かない
   (c) 設計ミスが多くなる
   (d) 予期しない状態に陥ったとき、正常状態に復帰できないことがある
   (e) 個人差ができ難い傾向がある
   (f) 以上のことより、非同期回路には問題点が多い
   (g) より簡単で扱いやすい同期回路を使用した方がいい

6 感想

感想

今回のレポートもとても時間がかかりました。一つ一つの課題はそれほど難しくありませんが、量が多かったです。量が多いのは特に問題ありませんが、問題文の意味が読み取れないのがありました。そこのところは前回と同じく、とてもふわふわしています。ここを書きなさい！などと言う感ではなく、〜について説明しなさい。が多くて大変でした。説明の場合、説明しようと思えば、沢山できます。逆に簡単に書こうと思えば簡単に書けます。どの程度書けばいいのか全くわかりません。その点では前回と同じく困りました。

でも今回のレポートはやってて、楽しかったです。もっと時間があれば、証明の仕方や計算方法をいろいろ調べ、いろんな解き方で解いてみたかったです。今回は時間がなくて、内容が薄くなっていますが、一生懸命頑張りました。とても大変でしたが面白かったです。でも、前回と同じく、考察やる気が起きませんでした。

参考文献

wikipedia
http://ja.wikipedia.org/wiki/

桁上がり先見加算器
http://ambition.ec.t.kanazawa-u.ac.jp/class/06/lsi/4.html

同期回路と非同期回路の違い
http://www.miyazaki-gijutsu.com/series4/densi0527.html