情報工学実験１

ーディジタル回路ー
汎用ロジックICによる組み合わせ回路の実現

075730G:澤岻千明

提出締切日:2008/07/04

共同研究者
075711A:山口藍
075726J:洌鎌温子
075759E:真地泰子

実験目的

本実験では、簡単な組み合わ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ている演算器の設計法について習得する。

実験概要

この実験では全加算器、3 ビットインクリメンタ、4 ビットインクリメンタを真理値表から論理関数を得、さらにこの論理関数を簡単化した。この簡単化はカルノー図を用いた。カルノー図から得た論理関数を分配則、XOR ゲートの使用、ド・モルガンの法則等を利用し、さらに簡単化を図る。

3 ビットインクリメンタの簡単化された論理関数から回路を設計した後、ブレッドボード、IC を用いて組み立てをし、真理値表通りに動くかどうか実際に確認する実験である。

実験

(1) 4 ビットインクリメンタの真理値表を書け。

(2) 全加算器 (表 2.2)、3 ビットインクリメンタ (表 2.3)、4 ビットインクリメンタ (実験 (1)) の各真理値表から、2.3.1 で述べた方法に従って、それぞれの論理関数を求めよ。

(3) 実験 (2) で用いた各真理値表をもとにカルノー図を描き、簡単化された論理関数を求めよ。
なお、このとき得られた論理関数に対して、さらに以下を適用しても構わない。

- **(a)** $A \cdot C + B \cdot C$ という式を、$(A + B) \cdot C$ に変換する (分配則の適用)
- **(b)** $A \cdot \overline{B} + \overline{A} \cdot B$ という式を、$A \oplus B$ に変換する (XOR ゲートの使用)

(4) 実験 (2) および実験 (3) で得られた論理関数を比較し、実験 (3) で得られた論理関数が簡単化されていることを確認せよ。

(5) 実験 (3) で得られた論理関数から、3 ビットインクリメンタの回路図を書け。

(6) 実験 (5) で描いた回路図をもとにして、ブレッドボード上に 3 ビットインクリメンタを実現し、その動作を確認せよ (3 つの回路全てをまとめて実現すること)。

実験結果

- 実験 (1),(2) 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。
    それぞれの真理値表を、全加算器 (表 1)、3 ビットインクリメンタ (表 2)、4 ビットインクリメンタ (表 3) として下に示す。

A	B	C_{in}	S	C_{ount}
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

表 1　全加算器の真理値表

x_2	x_1	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

表 2　3 ビットインクリメンタの真理値表

x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
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

表 3　4 ビットインクリメンタの真理値表

- 各真理値表から求めた論理関数を全て示せ (合計 9 つ)。
  なお、各論理関数の導出過程も示すこと。
  全加算器
  　真理値表が 1 の部分を抽出すると、
  　加算結果 S は

$$f(A,B,C_{in}) = f(0,0,1) + f(0,1,1) + f(1,0,0) + f(1,1,1)$$

　キャリー C_{out} は

$$f(A,B,C_{in}) = f(0,1,1) + f(1,0,1) + f(1,1,0) + f(1,1,1)$$

　これをまとめると次の様になる。

$$\begin{aligned} S &= \overline{A}\cdot\overline{B}\cdot C_{in} + \overline{A}\cdot B\cdot\overline{C_{in}} + A\cdot B\cdot C_{in} + A\cdot\overline{B}\cdot\overline{C_{in}} \\ C_{ont} &= \overline{A}\cdot B\cdot C_{in} + A\cdot\overline{B}\cdot C_{in} + A\cdot B\cdot\overline{C_{in}} + A\cdot B\cdot C_{in} \end{aligned}$$

3 ビットインクリメンタ

真理値表が 1 である出力を抽出すると

y_2 は

$$f(x_2, x_1, x_0) = f(0,1,1) + f(1,0,0) + f(1,0,1) + f(1,1,0)$$

y_1 は

$$f(x_2, x_1, x_0) = f(0,0,1) + f(0,1,0) + f(1,0,1) + f(1,1,0)$$

y_0 は

$$f(x_2, x_1, x_0) = f(0,0,0) + f(0,1,0) + f(1,0,0) + f(1,1,0)$$

これをまとめると次の様になる。

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

4 ビットインクリメンタ

真理値表で 1 の部分を抽出すると、

y_3 は

$$\begin{aligned}
f(x_3, x_2, x_1, x_0) &= f(0,1,1,1) + f(1,0,0,0) + f(1,0,0,1) + f(1,0,1,0) \\
&+ f(1,0,1,1) + f(1,1,0,0) + f(1,1,0,1) + f(1,1,1,0)
\end{aligned}$$

y_2 は

$$\begin{aligned}
f(x_3, x_2, x_1, x_0) &= f(0,0,1,1) + f(0,1,0,0) + f(0,1,0,1) + f(0,1,1,0) \\
&+ f(1,0,1,1) + f(1,1,0,0) + f(1,1,0,1) + f(1,1,1,0)
\end{aligned}$$

y_1 は

$$\begin{aligned}
f(x_3, x_2, x_1, x_0) &= f(0,0,0,1) + f(0,0,1,0) + f(0,1,0,1) + f(0,1,1,0) \\
&+ f(1,0,0,1) + f(1,0,1,0) + f(1,1,0,1) + f(1,1,1,0)
\end{aligned}$$

y_0 は

$$\begin{aligned}
f(x_3, x_2, x_1, x_0) &= f(0,0,0,0) + f(0,0,1,0) + f(0,1,0,0) + f(0,1,1,0) \\
&+ f(1,0,0,0) + f(1,0,1,0) + f(1,1,0,0) + f(1,1,1,0)
\end{aligned}$$

これらをまとめると次の様になる。

$$\begin{aligned}
y_3 &= \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 + x_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0} + x_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0 + x_3 \cdot \overline{x_2} \cdot x_1 \cdot \overline{x_0} \\
&+ x_3 \cdot \overline{x_2} \cdot x_1 \cdot x_0 + x_3 \cdot x_2 \cdot \overline{x_1} \cdot \overline{x_0} + x_3 \cdot x_2 \cdot \overline{x_1} \cdot x_0 + x_3 \cdot x_2 \cdot x_1 \cdot \overline{x_0} \\
y_2 &= \overline{x_3} \cdot \overline{x_2} \cdot x_1 \cdot x_0 + \overline{x_3} \cdot x_2 \cdot \overline{x_1} \cdot \overline{x_0} + \overline{x_3} \cdot x_2 \cdot \overline{x_1} \cdot x_0 + \overline{x_3} \cdot x_2 \cdot x_1 \cdot \overline{x_0} \\
&+ x_3 \cdot \overline{x_2} \cdot x_1 \cdot x_0 + x_3 \cdot x_2 \cdot \overline{x_1} \cdot \overline{x_0} + x_3 \cdot x_2 \cdot \overline{x_1} \cdot x_0 + x_3 \cdot x_2 \cdot x_1 \cdot \overline{x_0} \\
y_1 &= \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0 + \overline{x_3} \cdot \overline{x_2} \cdot x_1 \cdot \overline{x_0} + \overline{x_3} \cdot x_2 \cdot \overline{x_1} \cdot x_0 + \overline{x_3} \cdot x_2 \cdot x_1 \cdot \overline{x_0} \\
&+ x_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0 + x_3 \cdot \overline{x_2} \cdot x_1 \cdot \overline{x_0} + x_3 \cdot x_2 \cdot \overline{x_1} \cdot x_0 + x_3 \cdot x_2 \cdot x_1 \cdot \overline{x_0} \\
y_0 &= \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0} + \overline{x_3} \cdot \overline{x_2} \cdot x_1 \cdot \overline{x_0} + \overline{x_3} \cdot x_2 \cdot \overline{x_1} \cdot \overline{x_0} + \overline{x_3} \cdot x_2 \cdot x_1 \cdot \overline{x_0} \\
&+ x_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0} + x_3 \cdot \overline{x_2} \cdot x_1 \cdot \overline{x_0} + x_3 \cdot x_2 \cdot \overline{x_1} \cdot \overline{x_0} + x_3 \cdot x_2 \cdot x_1 \cdot \overline{x_0}
\end{aligned}$$

- 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数を全て示せ (合計 9 つ)。
    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も示すこと。
    全加算器

$$
\begin{aligned}
S &= \overline{A}\cdot\overline{B}\cdot C + \overline{A}\cdot B\cdot\overline{C} + A\cdot B\cdot C + A\cdot\overline{B}\cdot\overline{C} = \overline{A}(\overline{B}C + B\overline{C}) + A(BC + \overline{BC}) \\
&= \overline{A}(B \oplus C) + A(BC + \overline{BC}) \\
C_{out} &= A\cdot B + B\cdot C + C\cdot A = A(B + C) + BC
\end{aligned}
$$

3 ビットインクリメンタ

　3 ビットインクリメンタのカルノー図をそれぞれ、y_2(図 1)、y_1(図 2)、y_0(図 3) として次に示す。

x_0 \ $x_1 x_2$	00	01	11	10
0			1	1
1		1		1

図 1: y_2 のカルノー図

x_0 \ $x_1 x_2$	00	01	11	10
0		1	1	
1	1			1

図 2: y_1 のカルノー図

x_0 \ $x_1 x_2$	00	01	11	10
0	1	1	1	1
1				

図 3: y_0 のカルノー図

y_2 図１より、

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

y_1 図２より、

$$y_1 = x_1 \cdot \overline{x_0} + \overline{x_1} \cdot x_0 = x_1 \oplus x_0$$

y_0 図３より、

$$y_0 = \overline{x_0}$$

4 ビットインクリメンタ

$$\begin{aligned} y_3 &= x_3 \cdot \overline{x_1} + x_3 \cdot \overline{x_0} + x_3 \cdot \overline{x_2} + \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 \\ &= x_3(\overline{x_2} + \overline{x_1} + \overline{x_0}) + \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 \\ y_2 &= x_2 \cdot \overline{x_0} + x_2 \cdot \overline{x_1} + \overline{x_2} \cdot x_1 \cdot x_0 \\ y_1 &= \overline{x_1} \cdot x_0 + x_1 \cdot \overline{x_0} = x_1 \oplus x_0 \\ y_0 &= \overline{x_0} \end{aligned}$$

- 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。
    簡単化によって、それぞれの論理関数が纏まり見やすく、わかりやすくなった。これによって、回路の作成時、使用するゲート数が減り配線も少なくなるので、回路がより組み立て易くなる。特に、4 ビットインクリメンタはその違いが顕著である。
    3 ビットインクリメンタはカルノー図の簡単化後、分配則、XOR ゲートの使用によって更に簡単化された。
- 実験 (5) の結果について
  - 3 ビットインクリメンタの回路図を示せ。
    3 ビットインクリメンタの回路図を図４として、下に示す。

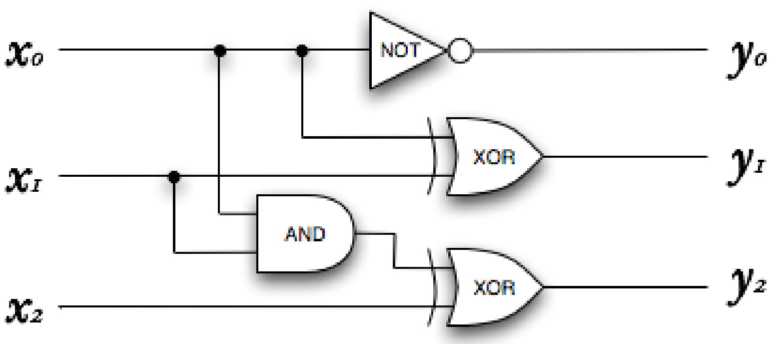

図 4: 3 ビットインクリメンタの回路図

- 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。
    この実験で作成した回路は、3 ビットインクリメンタの真理値表と同等の結果を示したので、これは 3 ビットインクリメンタと等価である。

考察

上記の実験結果に基づいて、以下の点について詳しく述べよ。

- 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。
    全加算器
    分配則、XOR ゲートを使用して簡単化を行ったが、これ以上は論理圧縮できない。
    3 ビットインクリメンタ
    これも全加算器と同様、これ以上の簡単化は行えない。
    4 ビットインクリメンタ
    この論理関数は、分配則、XOR ゲートも使用できないのでこのままでいい。
- 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、簡単化され易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。

    カルノー図を用いて論理圧縮する場合、'1' が隣合う、または 2^n で隣接するグループを作りそれぞれの論理和 (OR) を取り変数を消去して、簡単化された論理和を作成するのだが、このとき'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で簡単化しづらくなる。
    よって、全加算器の出力 C_{out}、3 ビットインクリメンタは簡単化され易く、4 ビットインクリメンタは単純に組み合わせが多いだけなので比較的簡単化され易い。全加算器の出力 S はすべて独立しているので、簡単化されにくい論理関数に分類される。
- その他の考察について
  - 本実験を通して得られた新たな知見について詳しく説明せよ。
    カルノー図は単純に論理関数を短くするものだと思っていたので、今回の授業は論理圧縮することによる利便さを知ることができた。また、簡単化された回路は使用するゲートも配線も少なくでき、大きな回路を設計するにはこのような論理圧縮はとても重要であると認識できた。

調査課題

下記の各項目について調査し、その結果を報告せよ。

(a) 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よく用いられるものに、桁上げ先見加算器 (carry look-ahead adder:CLA) と呼ばれる加算機がある。CLA とは、どのような加算器か調査し報告せよ。

通常の加算器では桁上げ入力、桁上げ出力がでてくるが、このままでは桁数が多くなると各素子の伝搬遅滞の合計より最終的な演算時間にも影響がでてくる。これを改善するためのものが桁上げ先読み (carry look ahead) である。これは桁上げ信号 (キャリー信号) 部を別に計算することで時間を短縮させる回路で、半加算器、全加算機と桁上げ先読み回路を含めて桁上げ先見加算器と呼ぶ。

(b) 乗算を行うディジタル回路を乗算器という。乗算器は、通常、多数の加算器から構成される。また、加算機の配置方法により、数種類の乗算器がある。乗算器の種類やそれぞれの特徴について調査し報告せよ。

乗算器にはディジタル乗算器とアナログ乗算器がある。その中のディジタル乗算器について調べた。

積和演算

演算の一つで、積の和を求める、つまり乗算の結果を順次加算する演算。累乗算とも言う。ディジタル信号処理において非常に使われる演算。また、デジタルシグナルプロセッサの性能指標としてこの演算が使われることもある。

リニアアレイ

1ビットずつ処理する単純な乗算器の複数回の繰り返しを1回で実行、乗数のビットの'0'、'1'に関わり無く複数ビットを同時に処理する方式である。複数の演算器を使って複数の部分積の加算を行っている。積レジスタの書き込みや読み出しの時間が減るので1ビットずつ処理する方式より多少は速くなるが、あまり高速化の効果は大きくない。

パラレルアダー

リニアトレイの部分積の加算という考え方より、乗数の各ビットに対応する全部の部分積を加算する回路を作ればループする必要は無く、1回で積が計算できる方式になる。乗数のビット数分の多数の部分積を、2分木構造に接続した Adder でトーナメント方式に加算を行えば $log2(N)$ 段の Adder の通過時間で計算できる。

(c) 本実験で学んだカルノー図による論理圧縮は、直観的にわかり易い方法であるが、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-マクラスキー法、あるいはクワイン-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-マクラスキー法とは、ど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-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化した場合の結果と比較せよ。

クワイン-マクラスキー法は、上記の問題を解決し、コンピュータ処理用に開発された圧縮手法の一つである。これはカルノー図と同様に、主加法標準形の論理圧縮に用いられる。ハミング距離が1のものは補元則を用いる。

クワイン・マクラスキー法は次の三段階からなる。

1. 関数の主項をすべて求める
2. 求めた主項を表に求め、必須項を求める
3. 最簡形を求める

例として、4ビットインクリメンタの y_3 の最簡形をクワイン・マクラスキー法で求める
4ビットインクリメンタの y_3 の論理式

$$
\begin{aligned}
y_3 &= \overline{x_3}\cdot x_2\cdot x_1\cdot x_0 + x_3\cdot\overline{x_2}\cdot\overline{x_1}\cdot\overline{x_0} + x_3\cdot\overline{x_2}\cdot\overline{x_1}\cdot x_0 + x_3\cdot\overline{x_2}\cdot x_1\cdot\overline{x_0} \\
&+ x_3\cdot\overline{x_2}\cdot x_1\cdot x_0 + x_3\cdot x_2\cdot\overline{x_1}\cdot\overline{x_0} + x_3\cdot x_2\cdot\overline{x_1}\cdot x_0 + x_3\cdot x_2\cdot x_1\cdot\overline{x_0}
\end{aligned}
$$

この論理関数から、ハミング距離1となる最小項の対を見つける必要がある。
その為に、論理関数の最小項をハミング重み(1の数)によって分類する。ここでは、ハミング重みをグループ G_n の n によって分ける。この表を表4に示す。

グループ	最小項
G_1	$x_3\cdot\overline{x_2}\cdot\overline{x_1}\cdot\overline{x_0}$
G_2	$x_3\cdot\overline{x_2}\cdot\overline{x_1}\cdot x_0$
	$x_3\cdot\overline{x_2}\cdot x_1\cdot\overline{x_0}$
	$x_3\cdot x_2\cdot\overline{x_1}\cdot\overline{x_0}$
G_3	$\overline{x_3}\cdot x_2\cdot x_1\cdot x_0$
	$x_3\cdot\overline{x_2}\cdot x_1\cdot x_0$
	$x_3\cdot x_2\cdot\overline{x_1}\cdot x_0$
	$x_3\cdot x_2\cdot x_1\cdot\overline{x_0}$

表4　y_3 の最小項の分類

この表の各最小項について、ハミング距離が1である組み同士で圧縮を行う。
両最小項の論理和に補元息を適用し、1変数少なくなった最小項を求める。(1次圧縮)
この圧縮された後の様子を表5に示す。

グループ	最小項
G_1,G_2	$x_3\cdot\overline{x_2}\cdot\overline{x_1}$
	$x_3\cdot\overline{x_2}\cdot\overline{x_0}$
	$x_3\cdot\overline{x_1}\cdot\overline{x_0}$
G_2,G_3	$x_3\cdot\overline{x_2}\cdot x_0$
	$x_3\cdot\overline{x_1}\cdot x_0$
	$x_3\cdot\overline{x_2}\cdot x_1$
	$x_3\cdot x_1\cdot\overline{x_0}$
	$x_3\cdot x_2\cdot\overline{x_1}$
	$x_3\cdot x_2\cdot\overline{x_0}$

表5　y_3 の1次圧縮

これを、圧縮できなくなるまで繰り返す。2次圧縮を表6に示す。

グループ	最小項
G_1,G_2,G_3	$x_3\cdot\overline{x_2}$
	$x_3\cdot\overline{x_1}$
	$x_3\cdot\overline{x_2}$
	$x_3\cdot\overline{x_0}$
	$x_3\cdot\overline{x_1}$
	$x_3\cdot\overline{x_0}$

表6　y_3 の2次圧縮

これ以上は圧縮できないので、圧縮はこれで終わるがまだ終了してはいない。
表 6 より、$x_3 \cdot \overline{x_2}$、$x_3 \cdot \overline{x_1}$、$x_3 \cdot \overline{x_0}$
が求められる。また、表 4 より圧縮されなかった $\overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0$ より、この論理関数は

$$\begin{aligned} y_3 &= x_3 \cdot \overline{x_2} + x_3 \cdot \overline{x_1} + x_3 \cdot \overline{x_0} + \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 \\ &= x_3(\overline{x_2} + \overline{x_1} + \overline{x_0}) + \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 \end{aligned}$$

と書き出せる。これは、実験 (3) の結果より、4 ビットインクリメンタの y_3 と等しいことがわかる。
この処理の流れを図 5 に示す。

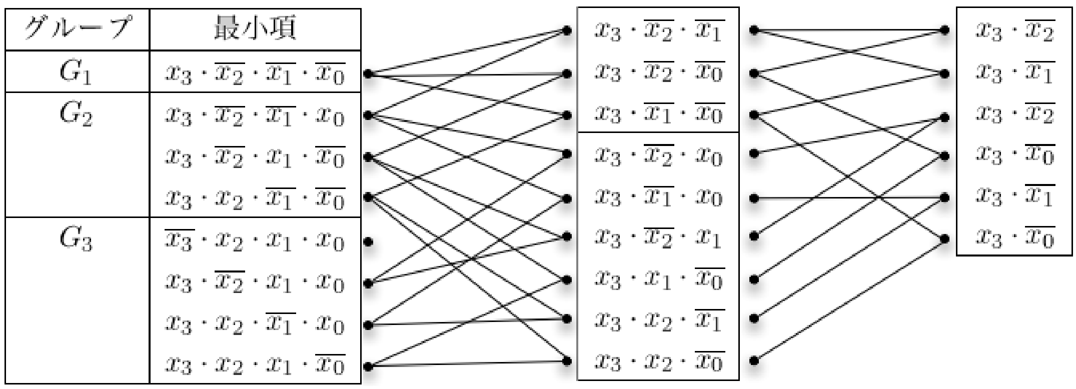

図 5: y_3 のクワイン・マクラスキー法による圧縮

感想

調査課題が一番手こずった。時間が無くやっつけ半分で仕上げたのであまり納得いくものではない。もっと図表を貼付れば、わかり易く充実したレポートを作成できたかもしれないと思うと残念。図表の作成自体は楽しかったが、手間がかかり過ぎたのであまり作れなかった。

前回と今回の実験で、ディジタル回路についてそこそこ理解できたのでそれを忘れないようにしたい。

参考文献・URL

- LaTex コマンドシート一覧　http://www002.upp.so-net.ne.jp/latex/
- LaTex-コマンド一覧　http://www1.kiy.jp/ yoka/LaTeX/latex.html
- 吉田たけお　尾知博　共著　「VHDL で学ぶディジタル回路設計」 2002 年出版、CQ 出版社
- 【コラム】コンピュータアーキテクスチャの話 (79)
  http://journal.mycom.co.jp/column/architecture/079/index.html
- Wikipedia
  加算器　http://ja.wikipedia.org/wiki/加算器
  クワイン・マクラスキー法　http://ja.wikipedia.org/wiki/クワイン・マクラスキー法
  積和演算　http://ja.wikipedia.org/wiki/積和演算