EXOR ゲートを用いた多段論理回路の検討

神田 徳夫[*]　　新田 貴之[*]

Multi-level Logic Circuits using EXOR Gates

Norio KODA[*] and Takayuki NITTA[*]

Abstract

This paper shows the properties of three-level logic circuits using EXOR gates. It is known that two-level AND-OR circuits(SOPs) require at most 8 product terms for 4-variable functions, while two-level AND-EXOR circuits(ESOPs) require at most 6 products. On the average, ESOPs require fewer products than SOPs. Therefore, using EXOR gates effectively, we can obtain smaller circuits than those of using conventional gates such as ANDs, ORs and NOTs. By the way, increasing the level of the circuit, we can synthesize circuits with fewer gates. We consider two types of three-level circuits: 1) SOP-EXOR and 2) ESOP-OR. As a results, it is shown that the proposed three-level circuits require, on the average, fewer AND gates than those of two-level SOPs and ESOPs.

Key Words: Combinational circuit, Exclusive-OR, SOP, ESOP, Multi-level circuit

1. まえがき

一般の論理回路は，通常 AND，OR，NOT を基本論理素子として設計する．これらを基本ゲートとする回路の設計は比較的容易である．このうち，AND - OR 二段論理回路は設計法が確立されており，PLA 構造で LSI 上に効率よく実現できるため広く利用されている．AND - OR 二段論理回路の性質は AND - OR 二段論理式（SOP）を用いて検討される．しかし，算術演算回路や通信用回路などでは，AND，OR，NOT のみで構成するよりも，EXOR ゲートを併用する方がゲート数を大幅に削減できる．一般の論理関数においても，平均すると EXOR ゲートを用いた方が OR ゲートを用いた場合よりも簡単な回路で実現できることが５変数以下の関数において証明されており[2]，このことは 6 変数以上の関数においても成立すると推測されている．そのため，EXOR ゲートを基本論理素子とした論理設計も考える必要がある．EXOR ゲートを用いた論理回路の基本は AND - EXOR 二段論理回路であり，その性質は AND - EXOR 二段論理式（ESOP）を用いて検討できる．与えられた関数を ESOP で表現した場合の性質や，出来るだけ簡単な ESOP で表現するための方法などについて近年盛んに研究されている[1−7]．

さて，与えられた論理関数を論理回路で実現するとき，基本となる回路構成は通常二段回路で検討されるが，この回路構成を三段に拡張すると，構成上の自由度が増すため，二段回路よりも簡単になることが多い[9]．本論文では，EXOR ゲートを用いた三段論理回路の性質について検討している．EXOR ゲートを用いた三段回路構成は種々考えられるが，本論文では，AND - OR 二段回路および AND - EXOR 二段回路との比較検討を目的とするため，SOP - EXOR 三段回路および ESOP - OR 三段回路構成を取り上げる．これらの回路構成について，4 変数以下の関数の最小回路を網羅的方法によって求め，この結果を二段最小回路と比較検討する．その結果， EXOR ゲートを用いることにより，いずれの回路構成においても二段回路よりも簡単な回路に合成できることを示す．また，従来の論理回路関係のカリキュラムにおいては，EXOR ゲートを用いた設計例としては算術演算回路や通信用回路などの特定の回路のみに適用して説明されてきたが，近年

[*] 情報電子工学科

の研究成果等から EXOR ゲートを一般的な論理関数の合成にも適用した説明を付加する必要性についても簡単に触れる.

2. 用語の定義

本章では, 本論文で用いる用語の定義について述べる.

[定義1] x と $\bar{x}$ を変数 x のリテラルという. また, 論理式に含まれるリテラルの総数をリテラル数という.

[定義2] 各変数のリテラルを高々１つしか含まない論理積を積項という. 論理式に含まれる積項の総数を積項数という.

[定義3] 積項を OR で結合したものを AND‐OR 論理和形（SOP : Sum Of Products expression）という. また, 積項を EXOR で結合したものを AND‐EXOR 論理和形(ESOP : Exclusive-OR Sum of Products expression）という.

[注意1] 図１は SOP および ESOP の回路構成を示す. SOP と ESOP は二段回路で実現できる. 論理式中の積項数とリテラル数はそれぞれ回路中の AND ゲート数と入力配線数に対応する.

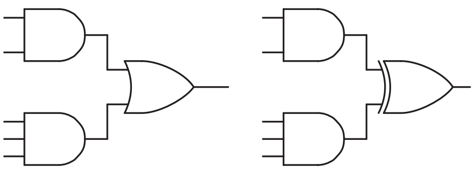

図１　SOP と ESOP の回路構成

[定義4] 論理関数 f の最小項展開

$$f = p_{n-1} \cdot x_1 x_2 \cdots x_n \vee p_{n-1} \cdot x_1 x_2 \cdots \bar{x}_n \vee \cdots \vee p_0 \cdot \bar{x}_1 \bar{x}_2 \cdots \bar{x}_n$$

において, その係数の列 $p_{n-1} p_{n-2} \cdots p_0$ を 2^n ビットの 2 進数とみたとき, これを論理関数 f の2進表現という. また, 2進表現を 16 進表現に変換したものを f の 16 進表現といい $hex(f)$ で表す.

3. 三段論理回路

4 変数関数を二段回路で合成した場合, SOP の場合は最大 8 積項, ESOP の場合は最大 6 積項で実現できる. また, 平均すると ESOP の方が SOP よりも少ない積項数で実現できるが, SOP の方が少ない積項数で実現できる関数も存在する[1]. さて, SOP や ESOP は二段回路であるが, この回路構成を三段以上の多段にすると構成上の自由度が増加するので, 積項数や配線数が少ない回路が合成できる可能性がある. これらのことから, 本論文では, SOP および ESOP の特徴を活用した三段回路について検討し, 二段回路との比較検討を行う.

3.1 三段論理回路の構成

三段回路の構成は種々考えられるが, 本論文では, SOP および ESOP による二段回路と比較すること, および, EXOR ゲートを用いることの効果を検討することを目的とするために,

１）2 つの SOP を EXOR する SOP‐EXOR 三段回路

２）2 つの ESOP を OR する ESOP‐OR 三段回路

の２つの回路構成について検討した. これらの回路構成を図２に示す.

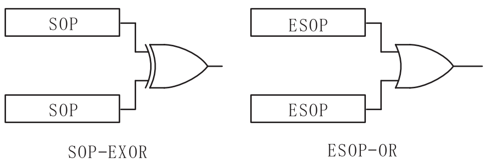

図２　三段回路構成

[定義5] 論理関数 f の真理値表中の１の個数を真理値表濃度といい, $d(f)$ で表す.

さて, 与えられた論理関数を論理式で表現する場合, 関数の真理値表濃度と必要な積項数との間には統計的に図３のような関係がある[10].

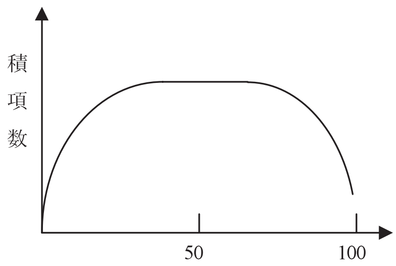

図３　真理値表濃度と積項数

すなわち, 平均すると, 与えられた関数の真理値表濃度が小さいかまたは大きいとき, 簡単な論理式で

表現できると言える．このことから本稿で検討する三段回路は次のような特徴を持つと考えられる．

1）SOP‐EXOR 三段回路の特徴

この回路構成の場合，2 つの SOP の表現する関数をそれぞれ，f_1，f_2 とするとき，与えられた関数 f は,

$$f = f_1 \oplus f_2$$

と表される．f を排他的に２つの部分関数に分解した場合は,

$$d(f_1) + d(f_2) = d(f)$$

となり，真理値表濃度の小さい２つの部分関数が得られる．また，$1 \oplus 1 = 0$ であるので,

$$d(f_1) + d(f_2) > d(f)$$

とすることもできる．この場合，真理値表濃度の大きい２つの部分関数が得られる．これらの性質により，部分関数 f_1，f_2 を SOP で合成し，最終段でこれらを EXOR することにより，二段回路よりも簡単な回路に合成できる可能性がある．

［例題１］表１は 4 変数関数 $f(x_1, x_2, x_3, x_4)$ を $f = f_1 \oplus f_2 = g_1 \oplus g_2$ と分解した例である．$d(f) = 10$ である．$f = f_1 \oplus f_2$ と分解したとき，$d(f_1) = 8$，$d(f_2) = 8$ となる．この場合，EXOR の性質により，与えられた関数 f の真理値が 0 の場合も部分関数 f_1，f_2 において１を立てることができる．また，$f = g_1 \oplus g_2$ と排他的に分解したときは，$d(g_1) = 6$，$d(g_2) = 4$ となる． ［例終］

表１　4 変数関数の分解例１

x_1	x_2	x_3	x_4	f	f_1	f_2	g_1	g_2
0	0	0	0	0	1	1	0	0
0	0	0	1	1	0	1	0	1
0	0	1	0	1	1	0	1	0
0	0	1	1	1	0	1	1	0
0	1	0	0	1	1	0	1	0
0	1	0	1	1	0	1	1	0
0	1	1	0	1	1	0	1	0
0	1	1	1	1	0	1	1	0
1	0	0	0	0	1	1	0	0
1	0	0	1	1	0	1	0	1
1	0	1	0	1	1	0	0	1
1	0	1	1	0	0	0	0	0
1	1	0	0	1	1	0	0	1
1	1	0	1	0	0	0	0	0
1	1	1	0	0	1	1	0	0
1	1	1	1	0	0	0	0	0

2）ESOP‐OR 三段回路の特徴

この回路構成の場合，２つの ESOP の表現する関数を f_1，f_2 とするとき，与えられた関数 f は,

$$f = f_1 \vee f_2$$

と表される．この場合も f を排他的に２つの部分関数に分解すると,

$$d(f_1) + d(f_2) = d(f)$$

となり，真理値表濃度の小さい２つの部分関数が得られる．また，$1 \vee 1 = 1$ であるので,

$$d(f_1) + d(f_2) > d(f)$$

となるように分解することが可能であり，真理値表濃度の大きい２つの関数が得られる．これらの性質により，部分関数 f_1，f_2 を ESOP で合成したものを OR することにより，二段回路よりも簡単な回路に合成できる可能性がある．

[例題２] 表２は表１と同じ関数を $f = f_1 \vee f_2 = g_1 \vee g_2$ となるように分解した例である．$f = f_1 \vee f_2$ と分解したとき，$d(f_1) = 8$，$d(f_2) = 8$ となる．この場合，OR の性質により，与えられた関数 f の真理値が 1 のとき，部分関数 f_1，f_2 の両方に１を立てることができる．また，$f = g_1 \vee g_2$ と排他的に分解したときは，$d(g_1) = 6$，$d(g_2) = 4$ となる． ［例終］

表２　4 変数関数の分解例２

x_1	x_2	x_3	x_4	f	f_1	f_2	g_1	g_2
0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	1
0	0	1	0	1	1	1	1	0
0	0	1	1	1	1	1	1	0
0	1	0	0	1	1	1	1	0
0	1	0	1	1	1	1	1	0
0	1	1	0	1	0	1	1	0
0	1	1	1	1	0	1	1	0
1	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	0	1
1	0	1	0	1	1	1	0	1
1	0	1	1	0	0	0	0	0
1	1	0	0	1	1	0	0	1
1	1	0	1	0	0	0	0	0
1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0

3.2 三段最小回路の合成法

本論文では，取り扱いの容易な 4 変数以下の論理関数を実現する回路を合成する．前述のように，4 変数関数を実現するために必要な最大積項数は，SOP の場合は 8 個，ESOP の場合は 6 個である．そこで，4 変数関数を三段回路で合成した場合，最小回路は 6 積項以下で実現できると推測される．例えば，SOP‐EXOR 回路の場合，2 つの SOP の積項数の和は 6 以下となると仮定する．そこで，入力側の２つの二段回路の論理式を f_1，f_2 としたとき，

$$（f_1 \text{ の積項数}）＋（f_2 \text{ の積項数}）\leq 6$$

の条件を満足する範囲で，任意個の積項を網羅的に組み合わせて２つの二段回路を構成し，それらのORまたはEXORしたものが与えられた関数を実現するように合成する．その際，２つの部分関数の間で共通の積項を用いると，回路上ではゲートを共有できることになり，全体としてより簡単な回路に合成できる．

[例題] 図４の回路において＊で示されたゲートは，SOP1およびSOP2の両方で共通に使われているので，必要なゲートが少なくてよい． [例終]

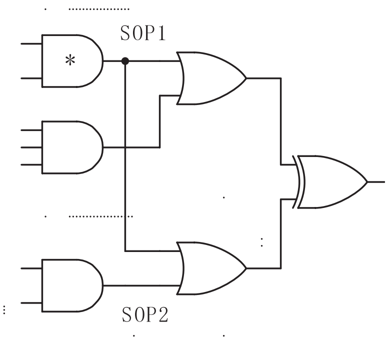

図４　ゲートを共有する回路例

さて，三段回路構成にすると，ORやEXORゲートが必ず3個必要になり，変数の個数の少ない関数の場合，総ゲート数は二段回路よりも多くなる．しかし，三段回路にすることによるゲート数の増加は3個と固定であり，変数の個数が多い関数になるとこの個数は無視できる．従って，本論文では，積項を構成するANDゲートの総数を回路の複雑度の尺度とする．

4. 結果及び検討

4変数論理関数の個数は65536個存在するが，これを性質の同じ関数群に分類することにより，対象とする関数の個数を少なくできる．本論文では，関数を変数の否定と置換に基づくNP同値類[8]に分類して検討した．１つのNP同値類に属する論理関数の回路構成は同一となり，回路の接続線を変更するだけでよい．すなわち，１つのNP同値類に属する関数群は同じ積項数の論理式で表現できることになる．4変数論理関数は402個のNP同値類に分類できるので，402個の代表関数の最小論理式を求めればよい．

3.2で述べたように最大積項数に制限を設けて合成した結果，全ての代表関数の三段回路を合成できた．このことは，必要な最大積項数の推測が正しかったといえる．表３は，4変数関数を各種回路構成によって合成したときの必要な積項数とそれによって実現できるNP同値類代表関数の個数を示す．

これより，二段回路の場合はSOPで最大8積項，ESOPで最大6積項が必要であるが，三段回路構成にすると，SOP - EXOR，ESOP - ORのいずれの場合においても最大５積項で全ての関数が合成できることが分かった．また，三段回路の場合，与えられた積項数で実現できる関数の個数はほぼ同じであり，２つの三段回路構成によってほぼ同等の品質の回路を合成できるといえる．

表３ 各種回路構成で必要な積項数と実現できるＮＰ同値類代表関数の個数の比較

積項数	ＮＰ同値類代表関数の個数			
	二段回路		三段回路	
	SOP	ESOP	SOP - EXOR	ESOP - OR
0	1	1	1	1
1	5	5	5	5
2	21	27	26	28
3	75	121	126	130
4	156	200	209	206
5	98	46	35	32
6	33	2	0	0
7	10	0	0	0
8	3	0	0	0

[合成例１] SOPとSOP - EXOR合成の比較例

$hex(f)=6996$ をSOPで合成すると，

$$f=\bar{x}_1\bar{x}_2\bar{x}_3x_4\vee\bar{x}_1\bar{x}_2x_3\bar{x}_4\vee\bar{x}_1x_2\bar{x}_3\bar{x}_4\vee\bar{x}_1x_2x_3x_4$$
$$\vee x_1\bar{x}_2\bar{x}_3\bar{x}_4\vee x_1\bar{x}_2x_3x_4\vee x_1x_2\bar{x}_3x_4\vee x_1x_2x_3\bar{x}_4$$

となり，SOP - EXORで合成すると，

$$f=(\bar{x}_3\bar{x}_4\vee x_3x_4)\oplus(\bar{x}_1\bar{x}_2\vee x_1x_2)$$

となる．SOPの場合は8積項が必要であるが，SOP - EXORの場合は４積項でよい．表４は，与えられた関数 f およびSOP - ESOPで合成したときの２つのSOPの表現する部分関数 f_1， f_2 を示す． $d(f)=8$， $d(f_1)=8$， $d(f_2)=8$ であり，EXORの性質により，部分関数の１の個数の和は，与えられた関数の１の個数よりも多くすることが可能となる． [例終]

[合成例２] ESOPとESOP - ORの合成比較例

$hex(f)=16fe$ をESOPで合成すると，

$$f=\bar{x}_1\bar{x}_2\bar{x}_3x_4\oplus x_1x_2x_3\bar{x}_4\oplus\bar{x}_2\bar{x}_3\oplus x_1x_4\oplus 1$$

となる．ESOP - ORで合成すると，

$$f=(\bar{x}_3\bar{x}_4\oplus\bar{x}_2\bar{x}_4)\vee(x_1\bar{x}_2\bar{x}_3\oplus\bar{x}_3\bar{x}_4\oplus\bar{x}_1)$$

となる．ESOPの場合は５積項必要であるが，

ESOP‐OR の場合，積項$\bar{x}_3\bar{x}_4$は両方の ESOP で共有できるので４積項でよい．表５は，与えられた関数および ESOP‐OR で合成したときの２つの ESOP の表現する部分関数を示す．$d(f)=10$，$d(f_1)=4$，$d(f_2)=8$ である．この場合は，部分関数の１つを１の少ない簡単な関数にすることにより，全体を簡単に合成できたと考えられる．

［例終］

表４　関数 6996 の合成

x_1	x_2	x_3	x_4	f	f_1	f_2
0	0	0	0	0	1	1
0	0	0	1	1	0	1
0	0	1	0	1	0	1
0	0	1	1	0	1	1
0	1	0	0	1	1	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	1	1	0
1	0	0	0	1	1	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	1	1	0
1	1	0	0	0	1	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	0	1	1

表５　関数 16fe の合成

x_1	x_2	x_3	x_4	f	f_1	f_2
0	0	0	0	0	0	0
0	0	0	1	1	0	1
0	0	1	0	1	1	1
0	0	1	1	1	0	1
0	1	0	0	1	1	0
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	0	0	0
1	0	0	1	1	0	1
1	0	1	0	1	1	0
1	0	1	1	0	0	0
1	1	0	0	1	1	1
1	1	0	1	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0

従来，EXOR を用いた二段回路は OR を用いた二段回路よりも簡単な回路で実現できると考えられてきたが，EXOR を用いた三段回路にすることにより，更に簡単な回路になる可能性が示されたといえる．

なお，AND，OR を用いた三段最小回路は筆者の知る限りにおいては公表されていないので厳密な比較はできないが，二段回路において EXOR を用いた方がコンパクトな回路に合成できるという有効性は，三段以上の多段回路においても保持されると推測される．

さて，従来の論理回路系の科目においては，EXOR を用いる回路例としてパリティ回路などの通信用回路や，加算器などの算術演算回路に限定されていた．しかし，近年の EXOR を用いた回路合成に関する研究の成果などから，EXOR を一般的な回路に適用することの有用性が明らかとなっている．EXOR を用いた論理回路を一般的に論じた日本語の教科書は一例しか見られないが[11]，今後，論理回路系の科目で EXOR を用いた回路合成に関する内容を付加していく必要があると考えられる．

5. まとめ

本論文では，EXOR を用いた三段回路合成について述べた．提案した三段回路構成は，従来提案されている二段回路の SOP および ESOP と比較検討するために，ESOP‐OR 構成および SOP‐EXOR 構成とした．4 変数関数の最小回路を網羅的方法によって合成した結果，以下の点が明らかとなった．

１）提案した三段回路は SOP や ESOP よりも簡単な回路に合成できる．

２）ESOP‐OR 回路と SOP‐EXOR 回路はほぼ同等の品質の回路となる．

これらのことから EXOR を用いて三段回路にするとより簡単な回路に合成できる可能性があるといえる．

また，従来の研究や本論文の成果などから EXOR ゲートを用いることの有用性の評価は定着してきたと言え，今後，論理回路系の科目の内容に取り入れていくことを検討する必要があると考えられる．

本論文では，限定した三段回路構成としたが，今後，ESOP と SOP を混合した，より一般的な三段回路構成についても検討する必要がある．また，三段回路に合成したときに最も簡単な回路になるように，与えられた論理関数を２つの部分関数に分解するヒューリスティックなアルゴリズムの開発が必要である．

文献

1）神田徳夫，笹尾勤：4 変数 AND‐EXOR 最小論理式とその性質，電子情報通信学会論文誌，Vol.J74-D-I，No.11, pp.765-773(1991)

2）神田徳夫，笹尾勤：AND‐EXOR 論理式の積項数の上界について，電子情報通信学会論文誌，Vol.J75-D-I，No.3, pp.135-142(1992)

3）神田徳夫，笹尾勤：下界界定理を用いた AND ‐ EXOR 論理式の簡単化法，電子情報通信学会論文誌，Vol.J76-D-I，No.1, pp.1-10 (1993)
4）神田徳夫，笹尾勤：多出力 AND ‐ EXOR 論理式簡単化の一手法，電子情報通信学会論文誌, Vol.J79-D-I，No.2, pp.43-52 (1996)
5）立野竜也，神田徳夫：積項の変形則を用いた AND ‐ EXOR 論理式の簡単化法の改善：電気・情報関連学会中国支部連合大会，pp.454(1997)
6）清水将史，神田徳夫：積項選択法による ESOP 論理式の簡単化，電気・情報関連学会中国支部連合大会，pp.32(1998)
7）Sasao, T.：EXMIN2: A simplification algorithm for Exclusive-Or-Sum of Products expressions for multiple-valued input two-valued output functions, IEEE Trans. on CAD, Vol.12, No.5, pp.621-632(1993)
8）室賀三郎著，笹尾勤訳：論理設計とスイッチング理論，pp.183，共立出版(1981)
9）笹尾勤：論理設計，pp.197，近代科学社(1995)
10）文献９），pp.229
11）文献９），第１２章

（2005.9.1 受理）