

3 入力多数決ゲートを用いた 5 変数論理関数の最小段数回路

All Five-Variable Logic Functions Can Be Computed by
Three-Input Majority Gates with Depth Four

守家大雄*

高木一義†

高木直史‡

1 序論

半導体集積回路の微細化や集積度向上の技術が発展するにつれて消費電力の増大が問題となっており、半導体に代わる低消費電力なデバイスの研究が進められている。断熱型磁束量子パラメトロン (Adiabatic Quantum Flux Parametron: AQFP) は消費電力が非常に小さいデバイスであり、データセンターといった消費電力の大きさが制約となる大規模な施設における運用が期待される。AQFP は AND ゲートや OR ゲートと同じ面積及び段数で多数決ゲートを構成することが可能であり、且つ論理否定を配線で実現可能であるため、AQFP 回路の高速化や面積削減をおこなうためには従来の論理合成手法とは異なる手法の開発が必須となる。

回路の段数は論理回路の速度及び面積に影響するため、段数削減は回路の最適化に非常に有効である。3 入力多数決ゲートを用いたパラメトロンによる 4 変数以下の各論理関数を実現する回路の最小段数は既に示されている [1]。本研究ではこれよりも変数が多い 5 変数論理関数を実現する回路の最小段数を求めることで、[2] などの既存のヒューリスティックな論理合成手法より最適化能力の高い論理合成手法の開発を目指す。

本稿では各 5 変数論理関数を実現する回路の最小段数の探索手法を示し、必要な段数の最大値を確認す

る。また、探索によって得られた事実を考慮して、より一般的な論理関数の最小段数についても議論する。

2 準備

断熱型磁束量子パラメトロン (AQFP) は奇数入力多数決ゲートを基本論理とする。入力数が増加するにつれゲートの動作は不安定になるため、一般に用いられるゲートは 3 入力もしくは 5 入力の多数決ゲートである。5 入力のゲートは一部の有効な回路においてのみ用いられることが多く、3 入力多数決ゲートを用いて回路を構成することが多いため、本稿では 3 入力多数決ゲートのみを扱う。3 入力多数決ゲートは入力に 3 個の論理関数 F, G, H が与えられると出力に $FG + GH + HF$ を返すゲートである。ここで、 $+$ は論理和を示す。

論理関数 F に変数の否定や変数の置換、関数値の否定の操作をおこなうことで論理関数 G が得られる場合に関数 F と G は NPN 同値であり、同じ NPN 同値類に属する。AQFP において値の反転は配線によって実現可能であるので、入力や出力の否定及び入力の置換をおこなっても回路の段数は変化しない。すなわち、関数 F と G が同じ NPN 同値類に属し、且つ G が段数 k の回路で実現可能な場合、 F もまた段数 k の回路で実現可能である。

*京都大学大学院情報科学研究科

†第 1 著者に同じ

‡第 1 著者に同じ

3 k 段以下の回路で実現可能な関数

3.1 k 段以下の回路で実現可能な関数の探索をおこなう 3 つの手法

すべての 5 変数論理関数が 4 段以下の回路で実現可能であることを以下の 3 段階に分割して確認した。

- ゲートへの入力 of 全組み合わせ
- シヤノン展開による部分的な確認
- 特定の関数の展開による確認

k 段目の多数決ゲートへの入力 3 個を $k-1$ 段以下で実現可能な関数から選ぶ。全ての組み合わせについて計算することで、 k 段以下で実現可能な関数を列挙できる。列挙手法について 3.2 節で述べる。この手法では 4 段以上のゲートへの入力に取り得る関数の個数が大きく、計算時間が非常に長くなる。

次に、3 段以下の回路では実現できない関数についてシヤノン展開をおこない、分割された 2 つの項が 3 段以下の回路で実現可能であるか否かを確認する。可能であれば、OR ゲートを用いることで 2 つの項の論理和を実現できるため、4 段の回路で実現可能な関数である。この部分的な探索について、3.3 節で述べる。

この探索では 4 段の回路で実現可能か判明しなかった関数について、全ての関数が特定の論理式の形をとったため、その形を分解することで探索空間の小さい探索をおこない、分解された 2 つの項がともに 3 段以下の回路で実現可能な関数であることを確認した。この探索について、3.4 節で述べる。

3.2 k 段以下の回路で実現可能な論理関数の列挙

最小段数回路を探索するために、 k 段目の多数決ゲートの入力に対してすべての $k-1$ 段以下で実現可能な論理関数の組み合わせを入力とした出力の計算をおこなう。まず、 $G_{\leq k}$ を k 段以下で実現可能な論理関数の集合とする。段数 1 のゲートへの入力にとり

得る論理関数は、基本入力変数と定数及びこれらの否定のみである。これは 0 段で実現できる論理関数であると見做すことが可能であるので、 $G_{\leq 0}$ は集合 $\{x, y, z, u, v, 0, x', y', z', u', v', 1\}$ となる。ここで、 x, y, z, u, v は基本入力変数であり、 $'$ は否定を表す。 $GNPN_{\leq k}$ を k 段以下で実現可能な各論理関数が属する NPN 同値類の関数元の集合とする。 $GNPN_{\leq 0}$ は集合 $\{x, 0\}$ となる。 $GNPN_{all}$ を各論理関数が属する NPN 同値類の代表関数すべての集合とする。 $GNPN_{all}$ の個数 $|GNPN_{all}|$ は 616126 である [3]。

Algorithm 1 $GNPN_{\leq k}$ を求める手法

Require: $G_{\leq k-1}, GNPN_{\leq k-1}$

Ensure: $GNPN_{\leq k}$

- 1: $GNPN_{\leq k} = GNPN_{\leq k-1}$
 - 2: **foreach** $F_1 \in GNPN_{\leq k-1}$
 - 3: **foreach** $F_2 \in G_{\leq k-1}$
 - 4: **foreach** $F_3 \in G_{\leq k-1}$
 - 5: $GNPN_{\leq k} = GNPN_{\leq k} \cup \{F_1 F_2 + F_2 F_3 + F_3 F_1$
が属する NPN 同値類の代表関数}
 - 6: **end foreach**
 - 7: **end foreach**
 - 8: **end foreach**
-

Algorithm 1 に、 $GNPN_{\leq k-1}$ から $GNPN_{\leq k}$ を計算するアルゴリズムを示す。 $G_{\leq k-1}$ は $GNPN_{\leq k-1}$ に NPN 操作を適用して展開することで得られる。 k 段目の多数決ゲートへの入力の 1 つに $GNPN_{\leq k-1}$ の要素を選択し、残りの 2 つの入力には $G_{\leq k-1}$ の要素を選択する。 F_{dif} を $G_{\leq k-1} \setminus GNPN_{\leq k-1}$ の要素であるとする。ここで、 $\setminus$ は差集合のことである。多数決ゲートへの 1 つ目の入力が F_{dif} である場合、NPN 操作（入力の否定及び置換、出力の否定）をおこない $GNPN_{\leq k-1}$ に属する NPN 同値類の代表関数へと変換する。同一の操作を $G_{\leq k-1}$ に属する他の 2 個の入力に適用してもその関数は $G_{\leq k-1}$ に属し、Algorithm 1 の計算と重複することになるため 1 つ目の入力 F_{dif} が $G_{\leq k-1} \setminus GNPN_{\leq k-1}$ の要素である必要はない。

Algorithm 1 による 3 段以下の AQFP 回路で実現可能な論理関数の列挙は現実的な計算時間で可能であ

る。計算時間は $|GNPN_{\leq k}| |G_{\leq k}|^2$ に比例するため、4 段もしくはそれ以上の段数については全組み合わせの探索は現実的ではない。そこで、探索空間を小さくすることで計算時間を削減し、効率的に探索をおこなうことを検討する。

3.3 シャノン展開による部分的な空間探索

$GNPN_{\geq k}$ を、実現に k 段以上の回路が必要な各論理関数が属する NPN 同値類の代表関数の集合とする。 $GNPN_{\geq 4} = GNPN_{all} \setminus GNPN_{\leq 3}$ である。3 段までの探索結果より、 $|GNPN_{\geq 4}| = 12236$ である。

5 変数論理関数を変数 x についてシャノン展開すると、式 (1) を得る。この展開は多数決ゲート 1 個で実現でき、式 (2) を得る。

$$F(x, y, z, u, v) = xF(1, y, z, u, v) + x'F(0, y, z, u, v) \quad (1)$$

$$F(x, y, z, u, v) = MAJ(1, xF(1, y, z, u, v), x'F(0, y, z, u, v)) \quad (2)$$

故に、 $xF(1, y, z, u, v)$ と $x'F(0, y, z, u, v)$ がそれぞれ属する NPN 同値類の代表関数が $GNPN_{\leq k-1}$ に存在すれば、 $F \in GNPN_{\geq 4}$ となる関数 F について、 $F \in GNPN_{\leq k}$ となる。3 段以下で実現可能な関数は既に列挙済みのため、 $GNPN_{\geq 4}$ の各関数 F についてシャノン展開をおこない、 $xF(1, y, z, u, v) \in GNPN_{\leq 3}$ 及び $x'F(0, y, z, u, v) \in GNPN_{\leq 3}$ を満たす F を $GNPN_{\leq 4}$ に挿入する。

探索の結果、12236 個の関数のうち 12101 個の関数はこの条件を満たすことを確認した。なお、各関数 F を展開して NPN 同値類の代表関数に変換し、集合 $GNPN_{\leq 3}$ 内に存在するか否かを確認する探索のため、 $|GNPN_{\leq k}| |G_{\leq k}|^2$ と比べて計算量が非常に小さく、容易にこの探索をおこなうことが可能である。

3.4 関数の分解による部分的な空間探索

残りの 135 個の関数によって構成される集合を $GNPN_{parity}$ とする。すべての関数 $F \in GNPN_{parity}$ が

以下の式 (3), (4) で表現できることが確認できた。

$$x'F(0, y, z, u, v) + x(y \oplus z \oplus u \oplus v) \quad (3)$$

$$x'F(0, y, z, u, v) + x(y \oplus z \oplus u \oplus v)' \quad (4)$$

論理関数は必ず 2 個の論理関数の論理積の形で表すことが可能であるため、 $F(0, y, z, u, v)$ を式 (5) のように分解する。式 (3), (4) の第 2 項について、式 (6), (7) のように分解可能であるため、式 (3), (4) は式 (8), (9) のように分解可能である。

$$F(0, y, z, u, v) =$$

$$F_{1st}(0, y, z, u, v) F_{2nd}(0, y, z, u, v) \quad (5)$$

$$y \oplus z \oplus u \oplus v =$$

$$(yz' + y'z + uv' + u'v)(yz + y'z' + uv + u'v') \quad (6)$$

$$(y \oplus z \oplus u \oplus v)' =$$

$$(yz + y'z' + uv' + u'v)(yz' + y'z + uv + u'v') \quad (7)$$

$$x'F(0, y, z, u, v) + x(y \oplus z \oplus u \oplus v) =$$

$$(x'F_{1st} + x(yz' + y'z + uv' + u'v)) \wedge$$

$$(x'F_{2nd} + x(yz + y'z' + uv + u'v')) \quad (8)$$

$$x'F(0, y, z, u, v) + x(y \oplus z \oplus u \oplus v)' =$$

$$(x'F_{1st} + x(yz + y'z' + uv' + u'v)) \wedge$$

$$(x'F_{2nd} + x(yz' + y'z + uv + u'v')) \quad (9)$$

故に、分解後の 2 個の論理関数の NPN 同値類の代表関数がそれぞれ $GNPN_{\leq 3}$ に属するのであれば $F \in GNPN_{parity}$ は $GNPN_{\leq 4}$ に属する。分解後の 2 個の論理関数は高々 2^{16} 程度の数であるため、現実的な計算時間で探索可能である。探索の結果、 $GNPN_{parity}$ に属するすべての関数が $GNPN_{\leq 4}$ に属することを確認した。

3.5 最小段数の探索結果

計算機を用いて探索をおこない、すべての 5 変数論理関数は 4 段以下の回路で実現可能であることが確認できた。表 1 は k 段以下で実現可能な論理関数の数を示したものである。

表 1: k 段以下で実現可能な関数の数

段数	5変数論理関数の数	同値類の代表関数の数
1	172	4
2	253732	132
3	4225442636	603890
4	4294967296	616126

4 3入力多数決ゲートを用いた回路の段数の上界及び下界

k_n を, 全ての n 変数論理関数を実現可能な回路の最小段数とする.

本章では, k_n の上界及び下界について議論する. 本章における段数は, k_n のことを示す. まず k_n の上界について議論する.

n 変数論理関数の各変数についてシャノン展開をおこなうと, 最大で 2^n 個の最小項の論理和に展開される. 例えば 2 変数であれば, $F(x,y) = xF(1,y) + x'F(0,y) = xyF(1,1) + xy'F(1,0) + x'yF(0,1) + x'y'F(0,0)$ となり, $F(1,1), F(1,0), F(0,1)$ 及び $F(0,0)$ は定数となるため最大で $xy, xy', x'y, x'y'$ の 4 個の最小項の論理和となる. 最小項は各変数の論理積をとったものであるため, 各最小項は $\lceil \log_2 n \rceil$ 段の AND ゲートを用いて実現可能である. 最小項の個数を p 個とすると, 関数値を否定して得た関数の最小項の個数は $2^n - p$ 個となり, 最大で 2^{n-1} 個の最小項で十分となる. 故に, 2^n 個の最小項の論理和をとるためには $\log_2 2^{n-1} = (n-1)$ 段の OR ゲートを用いれば実現可能である. また, AND ゲート及び OR ゲートは, 多数決ゲートの入力の 1 つをそれぞれ 0,1 に固定すれば実現可能である. 故に, 最大で $\lceil \log_2 n \rceil + (n-1)$ 段の多数決ゲートが存在すれば, n 変数論理関数を実現する回路を構成可能となり, これが段数の上界となる.

また, n 変数論理関数の 1 個の変数 (x とする) についてシャノン展開をおこなうと, これは $xF(1,y,z,\dots) + x'F(0,y,z,\dots)$ となり, ある $(n-1)$ 変

数論理関数及び x の論理積と, ある $(n-1)$ 変数論理関数及び x' の論理積との論理和をとった関数である. AND ゲートと OR ゲートを用いて実現すれば, n 変数論理関数は $(k_{n-1} + 2)$ 段以下で実現可能となる. $\lceil \log_2 n \rceil + (n-1) \leq k_{n-1} + 2$ であるが, k_{n-1} の値が判明している 5 段までは上界として用いることが可能である.

以上より, n 変数論理関数の上界として, $\lceil \log_2 n \rceil + n - 1$ の値をとることが可能である. k の値が確認されているのは 5 段までであり, $k_2 = 2, k_3 = 2, k_4 = 4$ 及び $k_5 = 4$ である.

特殊な例として, EXOR 演算には結合律が成り立ち, パリティ関数は 3 変数では 2 段で実現可能であるので, この 3 個の変数にそれぞれ 3 変数までのパリティ関数を代入すれば, 4 段で 9 変数までのパリティ関数を実現できる. 同様の操作をおこなうことで, n 変数パリティ関数は $2\lceil \log_3 n \rceil$ 段で実現可能である.

3 入力のゲートを用いるため, 真に n 変数の論理関数を実現するためには $\lceil \log_3 n \rceil$ 段は必要である. 故に下界は $\lceil \log_3 n \rceil$ 段である.

表 2 に, 3 入力多数決ゲートによって n 変数論理関数を実現する場合の回路の段数の上界と下界の値を示す.

表 2: n 変数論理関数を実現する回路の段数の上界及び下界

n	上界	k_{n-1} を用いる上界	下界	k_n
2	2	2	1	2
3	4	4	1	2
4	5	4	2	4
5	7	6	2	4
6	8	6	3	
$\vdots$		$\vdots$	$\vdots$	
n	$\lceil \log_2 n \rceil + n - 1$	$k_n + 2$	$\lceil \log_2 n \rceil$	

5 論理合成への応用

各5変数論理関数を実現する回路のライブラリを探索結果を元に作成しておき、必要な関数に適する回路を求める際には元の関数を5変数論理関数に分解して各5変数関数ごとにライブラリから回路データを出力して合成することで、一定の段数削減された回路を得る応用は可能である。本研究では4変数と5変数の論理関数はともに最大4段の回路で実現可能であることを確認しており、4変数論理関数に分解して回路を構成した場合には5段以上になる5変数論理関数についても4段までの回路で抑えることができるということを示している。

しかし、より段数削減能力を高めるために、既存のヒューリスティックな手法と本研究の結果との比較をおこない既存手法の改善をおこなうことが今後の課題となる。

参考文献

- [1] T. Tsuboi, “A logical Design of Circuit Representing 4-Variable Boolean Functions by means of 3 Fan-ins Parametrons”, Information Processing Society of Japan, 1963.
- [2] L. Amaru, P.-E. Gaillardon, G. De Micheli, *Majority-Inverter Graph: A Novel Data-Structure and Algorithms for Efficient logic Optimization*, DAC, 2014 51st ACM/EDAC/IEEE.
- [3] S. Muroga, “*logic Design and Switching Theory*”, Kyoritsu Shuppan, 1981.