

BDDに基づく光論理回路における双対端子を利用した面積と電力の削減手法

松尾 亮祐^{1,a)} 湊 真一¹

概要: 集積ナノフォトニクスに基づく光論理回路は非常に高速な動作を実現することができるために近年注目を集めている。光の高速性を活かした論理合成手法として BDD に基づく手法が注目されているが、回路中に多数存在するスプリッタが原因で消費電力が非常に大きくなる課題がある。BDD に基づく光論理回路では双対な入力端子が余っている。この端子を活用してスプリッタを削減する手法を提案する。提案手法により、回路の遅延を増加させることなく、面積を削減しながら消費電力を大幅に削減できることを示す。ISCAS'85 ベンチマーク回路に LUT ベースの FPGA テクノロジーマッパーを適用して得られる 10 入力関数を用いた実験を行い、提案手法によりベストケースでは消費電力が 2 桁程度削減されることを示す。

1. 序論

集積回路技術や光通信技術は、高度な情報化社会の実現に大いに貢献してきた。集積回路技術はその登場以来、目覚ましい発展を遂げてきたが、ゲートあたりの遅延は 10 ps 程度までしか削減できないことが示されている。[1] 近年、ナノフォトニクスの発展によって、光通信技術は短距離での通信やシリコンチップ上での通信に用いられるようになりつつある [2]。そして今日、ナノフォトニクスデバイスを用いた光集積回路により、超高速な動作を実現することが期待されている。ナノフォトニック DC に基づく光ゲートの遅延は数百 fs のオーダーである [3]。これは、CMOS 論理ゲートよりも 10 倍高速である。

既存の光集積回路の合成手法の多くは DC(DC) を基本ブロックとして用いている。図 1 (a) に示すように、DC は電圧入力端子を一つ、光の入力端子と出力端子を 2 つずつ持ち、入力光はその出力先を電圧入力によって制御される。パストランジスタ回路と同様に、DC を直列に接続することによりブール関数を回路実装することができる。光が DC を通過して出力に到達することにより演算が完了し、それぞれの DC に同時に電圧が印加されてから、入力光が出力に到達するまでの時間が回路の遅延に相当する。図 1 (b) に示す N 入力 AND 回路の遅延は N ps となる (本稿では、DC の遅延は 1 ps とする)。レーザーダイオード (LD) から出力された光が光検出器 (PD) で検出されるための必要十分条件は、全ての電圧入力が入力 ON であることであるため、図 1 (b)

の回路は N 入力 AND 回路として動作する。規模の大きな一般的な論理関数を光回路で実装するための自動設計手法が提案されている。さまざまなデータ構造に基づく手法が提案されているが、BDD に基づく光回路は超高速かつ面積効率がよいために、BDD に基づく設計手法がこれまでに主に研究されている。しかし、BDD に基づく光回路の消費電力は非常に大きくなるのがわかっており、消費電力を削減するための手法が提案されている [4-7]。

BDD に基づく回路の消費電力が大きくなる主な原因の一つは、回路中に導波路の枝分かれ (スプリッタと呼ばれる) が多数存在することである。スプリッタでの電力損失を削減する手法が提案されている [4-6]。本稿ではある条件を満たすノードに着目してスプリッタを削除する手法を提案する。このノードを双対ノードと呼び、提案手法を双対ノード共有と呼ぶ。双対ノード共有では双対ノードに対応する DC を共有することによりスプリッタを削除することで、回路の遅延を増加させることなく、回路面積を削減しながら消費電力を大幅に削減する。また、本稿では双対ノード共有を取り入れた BDD に基づく回路の論理合成手法を提案する。双対ノード共有を取り入れた合成手法の計算量は、双対ノード共有を取り入れない従来の BDD に基

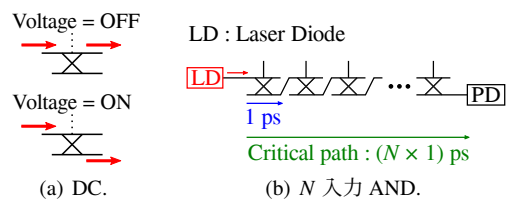


図 1 DC を用いた光論理回路。

¹ 京都大学大学院情報学研究所

^{a)} matsuo.ryosuke.25r@st.kyoto-u.ac.jp

づく合成手法と同程度となる。双対ノード共有の効果を評価するために、LUT ベースの FPGA テクノロジマッパーを ISCAS'85 の C7552 ベンチマーク回路に適用して得られる 10 入力関数を用いて実験を行った。双対ノード共有により指数的に消費電力が削減できるという結果が得られた。

本稿は以下のような章構成となる。2 章では、BDD に基づく光回路と関する先行研究をまとめる。3 章では、双対ノード共有を提案し、その効果を説明する。4 章では、双対ノード共有を取り入れた BDD に基づく回路の合成手法を提案する。5 章では、実験結果について考察を行う。6 章で、本稿のまとめを行い、今後の研究内容についても述べる。

2. 光論理回路と BDD

2.1 BDD に基づく光論理回路

図 2 (a) に示す BDD は $f = (\neg a \wedge b) \vee (a \wedge \neg b)$ に対応する。それぞれのノードは関数の入力変数でラベル付けされており、最も上位のノード(根ノード)から変数の論理値に応じて枝を選んでいき、最終的に終端ノードにたどり着く。変数の論理値が 0(1) のときに選ばれる枝を 0(1)-枝という。終端ノードには 0 または 1 の論理値が与えられており、その値が BDD が対応する論理関数の出力値となる。また BDD を構築する際には、ノード数を小さくするためにそれぞれのノードに固有の ID を用いてそれぞれのノードが削除可能かどうか調べている。図 2 (a) の BDD に基づく回路を図 2 (b) に示す。BDD に基づく回路設計では、根ノードを光の出力先として、1-終端を光を入射する部分として用いる。また、BDD のノードは DC で実装され、枝分かれはスプリッタで実装される。入力変数は DC に印加される入力電圧に対応しており、光の出力先を制御する。直列に接続された DC を光が通過することで演算が完了するため、BDD に基づく回路の動作は非常に高速である。 n 入力関数の場合には、回路の遅延は n ps となる。DC の数はノード数に相当し、スプリッタの面積は DC と比べて無視できる程に小さいため、本稿ではノード数が回路面積に相当するとみなす。BDD は少ないノード数で一般的な論理関数を表現可能であるため、BDD に基づく回路は高い面積効率を持つ。一方で、BDD に基づく回路は消費電力が非常に大きくなる課題を持つ。出力光の信号強度が十分でない場合、光の検出にかかる時間が増加することやノイズの影響により信号の検出を誤るといった不都合が生じる。しかし、入力光が DC やスプ

リッタを通過する度に信号強度が減衰するため、回路の出力で十分な信号強度を確保するために光源で必要な電力が大きくなる。DC での信号減衰は -1 dB と想定される [8]。スプリッタの分割比が 1:1 の場合、入力光の信号強度を二つの出力に等しく分配する。つまり、片方の出力から見ると信号強度の損失は $50\%(= 3$ dB) とみなせる。したがって、回路の出力で十分な信号強度を確保するためにレーザーダイオードで大量の電力が消費される。DC のスイッチングエネルギーは無視できるほどに小さいため、レーザーダイオードで消費される電力は回路の消費電力とみなすことができる。スプリッタの分割比を最適化することでスプリッタでの信号強度の損失を削減することができる [5]。しかし、この手法はスプリッタの出力光のうち一つは必ず捨てられるという根本的な課題を解決することはできない。この課題への対策としてスプリッタを DC に置き換える手法が提案されている [6]。しかし、この手法では DC の数と遅延が増加する。一方、双対ノード共有では遅延を増加させることなく DC の数を削減しながらスプリッタを削除することができる。

2.2 消費電力モデル

挿入損失 (Insertion loss) に着目して回路の消費電力とスプリッタの最適分割比を求める手法が提案されている [9]。ここでは、挿入損失を根からあるノード v までのパス上の DC やスプリッタでの信号損失の総量と定義する。挿入損失は次のように定式化される。

$$loss_v = A \times \sum loss_{parent} \quad (1)$$

ここでは、 A は DC での信号の減衰を補償するための値である。本稿では、DC での信号の減衰を -1 dB と想定するため、 $A = 1.25$ となる。 $loss_{parent}$ はノード v の親ノードの挿入損失と定義する。したがって、式 (1) は親ノードの挿入損失の和に A をかけることによりノード v の挿入損失を求めることができるということを意味している。しかし、[5] で提案されているスプリッタの分割比最適化を適用しない場合には、挿入損失は $loss_v$ よりも大きくなる。二つの親ノード x と y が与えられたとき、スプリッタの分割比 $SR_x : SR_y$ を $loss_x : loss_y$ に設定することでスプリッタでの信号損失を最小化することができる。ここでは、 $SR_x(SR_y)$ はノード $x(y)$ に分配される信号強度の割合を表す。同様に、三つの親ノード x と y, z が与えられたとき、スプリッタの分割比 $SR_x : SR_y : SR_z$ を $loss_x : loss_y : loss_z$ に設定することで信号損失を最小化することができる。これは親ノードの数が 4 以上の場合も同様である。式 (1) を根ノードから 1-終端まで全てのノードに対して計算することにより、回路の消費電力と最適なスプリッタの分割比を求めることができる。フォトディテクターで検出可能な最小の信号強度の値(本稿では $10 \mu\text{W}$ とする)と 1-終端の挿入損失の積が回路

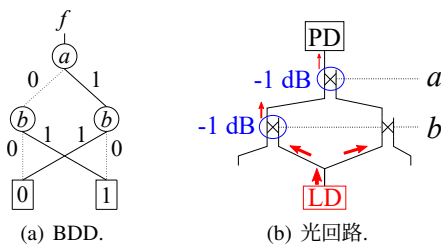


図 2 BDD に基づく光論理回路.

の消費電力となる。したがって、この手法の計算量はノード数に比例した値になる。

3. 双対ノード共有

3.1 概要と効果

本章では、双対ノード共有について説明し、さらに消費電力とノード数に対する双対ノードの効果について述べる。同じ変数でラベル付けされた二つのノード v と u が与えられたとき、二つのノードが次の条件 (1),(2) を満たす場合に、ノード v と u を双対ノードと定義する。(1) v の 0-子ノードと u の 1-子ノードが同じである。(2) v の 1-子ノードと u の 0-子ノードが同じである。図 3 (a) において、ノード v と u は双対ノードである。図 3 (b) は双対ノード a と b に対応する従来の BDD に基づく回路を示す。この回路中には DC とスプリッタが 2 つずつ存在する。式 (1) からノード c の挿入損失は $A \times (loss_a + loss_b)$ となる。ノード d についても同様である。従来の BDD に基づく回路では双対ノードに対応する部分にはスプリッタが必要であるが、使用されていない双対端子を用いることで二つのスプリッタを除去することができる。左の DC について考える。左上の端子と右上の端子は双対になっている。ここで双対とは、ノード $c(d)$ から来た光が左上の端子に到達するとき、ノード $d(c)$ から来た光は右上の端子に到達するということを意味する。図 3 (c) に示すように、右上の端子をノード b に接続することにより、一つの DC で双対ノードに対応する回路を実装することができる。 $x = 0$ のとき、ノード $c(d)$ からの光はノード $a(b)$ に到達する。 $x = 1$ のとき、ノード $c(d)$ からの光はノード $b(a)$ に到達する。図 3 (a) からこの動作が正しいことがわかる。全ての DC において、二つの出力端子は双対である。これは入力端子についても同様である。したがって、双対ノードの回路実装として図 3 (c) に示す回路を用いることができる。この手法を双対ノード共有と定義する。双対ノード共有によって、レーザーダイオードからフォトディテクターまでのパス上の DC の数が増えることはないため回路の遅延は増加しない。図 3 (c) のノード c の挿入損失について考える。ノード c からの光は DC を一つだけ通過してから入力変数 x に応じてノード a または b に到達する。どちらのノードに到達する場合でも、信号強度は光が到達するノードの挿入損失よりも大きくなっていなければならない。したがって、ノード c の挿入損失は $A \times (\max(loss_a, loss_b))$ となる。ノード d についても同様である。双対ノード共有を取り入れた回路の挿入損失は従来の BDD に基づく回路よりも小さくなるため、双対ノード共有により消費電力を削減することができる。 $loss_a = loss_b$ の場合を考える。従来の BDD に基づく回路では $loss_c = 2A \times loss_a$ となる。一方、双対ノード共有を適用した回路では $loss_c = A \times loss_a$ となる。したがって、ベストケースでは双対ノードのペア一つにつき消費電

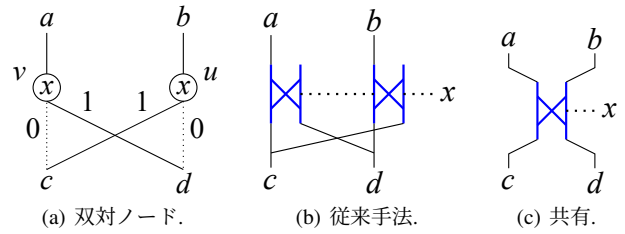


図 3 双対ノード共有の概要。

力が半減されるため、従来の BDD に基づく回路と比べて双対ノード共有を取り入れた回路の消費電力は指数的に小さくなる。また双対ノード共有により DC の数も削減することができる。図 3 (c) 中の DC の数は図 3 (b) と比べて一つ少なくなっている。つまり、双対ノードのペア一つにつき DC の数が一つ削減される。しかし、DC の数の削減量は消費電力の削減量と比べて小さいため、双対ノード共有は消費電力削減に適すると言える。まとめると、双対ノード共有は遅延を増加させることなく、DC の数と消費電力を削減することができる。したがって、全ての双対ノードに対して双対ノード共有を適用した回路が双対ノード共有を考慮した場合の最適な回路となる。

3.2 ベストケースの例

従来の BDD に基づく回路で論理関数を実装する場合には、ノード数の少ない BDD で表現可能な関数であっても、その回路の消費電力が非常に大きくなる場合がある。AND 関数と XOR 関数の例について考える。図 1 (b) と図 4 はそれぞれ AND と XOR の回路を示す。関数の入力数を n とすると AND と XOR のノード数はそれぞれ n と $2n-1$ である。したがって、AND と XOR の回路の DC の数はそれぞれ n と $2n-1$ となる。挿入損失に基づき回路の消費電力を考える。2 章で説明した計算法から AND の回路の消費電力は $A^n \times 10 \mu W$ (以下では、 $\times 10 \mu W$ は省略する) であるとわかる。図 4 中の x_{i+1} でラベル付けされた DC と x_i でラベル付けされた DC について考える。DC の間にはスプリッタが存在するため、 $loss_{x_{i+1}} = A \times loss_{x_i} + A \times loss_{x_i} = 2A \times loss_{x_i}$ ($i > 1$) となる。根ノードから 1-終端まで挿入損失を計算することで、回路の消費電力が $A^n \times 2^{n-1}$ であることがわかる。XOR の回路において、DC の数は入力数に対して線形オーダであるが、消費電力は指数オーダである。上記の議論から XOR の回路の消費電力は AND の回路と比べて入力数に対して指数的に大きくなることがわかる。しかし、双対ノード共有により XOR の回路の消費電力を指数的に削減する

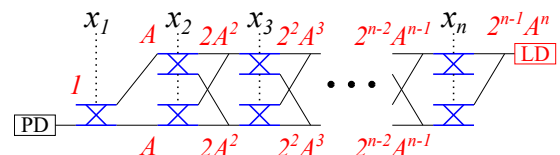


図 4 従来の BDD に基づく回路。

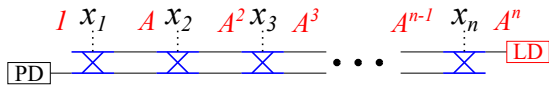


図5 双対ノード共有を取り入れた BDD に基づく回路.

ことができる。図5は双対ノード共有を取り入れた XOR の回路を示す。 x_{i+1} でラベル付けされた DC と x_i でラベル付けされた DC について考える。DC の間にはスプリッタが存在しないため、 $loss_{x_{i+1}} = A \times loss_{x_i}$ であり、消費電力は A^n まで削減される。 x_i でラベル付けされた DC の数は1であるため、回路全体の DC の数は n となる。つまり、双対ノード共有により遅延を増加させることなく、DC の数を削減しながら、消費電力を AND 回路と等しい値まで削減できる。まとめると、XOR などの消費電力の大きな関数は光論理回路の実用性における課題となる。しかし、スプリッタが原因で消費電力が大きくなっている場合には、双対ノード共有により消費電力を大幅に削減することができる可能性がある。したがって、双対ノード共有は BDD に基づく光論理回路の実用性における課題を解消することにつながる。

4. 論理合成

本章では、双対ノード共有を取り入れた BDD に基づく回路の論理合成手法を提案する。合成手法のフローを以下に示す。まず対象の論理関数を BDD に変換する。そしてこの BDD を基に回路を設計する。回路設計において提案の合成手法は DC の間の接続法とスプリッタの分割比の最適化を目指す。

4.1 BDD の最適化

Input inverters [10] を取り入れた BDD が提案合成手法に適する。Input inverter は属性エッジの一つである。その働きは属性エッジとしてよく知られている否定枝の働きと異なる。図6(b)は input inverter を取り入れた BDD であり、図6(a)に示す BDD に対応する。Input inverter は次のノードにおいて0-枝と1-枝を入れ替えるという操作を行うことを示すものである。つまり、この操作は入力変数の否定をとるとみなせる。Input inverter を適切に扱わないと、部分グラフがそれぞれに特有の関数を表すという BDD の性質が維持されなくなる。しかし、0-枝の ID は1-枝の ID よりも小さくなるという制約を設けることによりこの性質を保

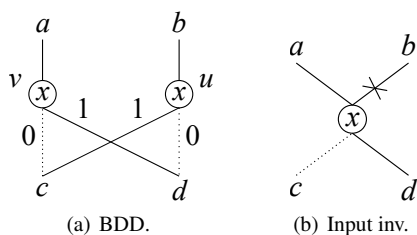


図6 Input inverter の概要.

つことができる。この制約のために、双対ノードを持たないノードであっても親ノードと input inverter で接続される場合がある。通常の BDD では、それぞれのノードに固有の ID を用いてあるノードが削除可能かどうか調べている。同様に、あるノードが親ノードと input inverter で接続できるかはノードの ID を調べることにより判定できる。したがって、input inverter を実装しても、通常の BDD と比べて計算コストが増加することはない。図6と3(a)からわかるように、双対ノードと input inverter は一対一対応している。ゆえに、input inverter を最大限に取り入れた BDD では全ての双対ノードのペアが一つのノードに統合されている。3章で説明したように、全ての双対ノードに双対ノード共有を適用することで、双対ノード共有を取り入れた回路を最適化することができる。したがって、提案合成手法では双対ノード共有を最大限に活用するために、対象の論理関数を表現する BDD に input inverter を適用する。Input inverter を取り入れた BDD のノード数は双対ノード共有が適用された回路の DC の数と等しくなるため、input inverter を適用することにより BDD から双対ノード共有を取り入れた回路の DC の数を求めることが容易になる。

4.2 回路の最適化

DC の間の接続法とスプリッタの分割比の最適化について説明する。その他の回路構成は2章で説明した従来の BDD に基づく回路の構成と同じである。まず、input inverter で接続されたノードに対応する DC をどのように接続するかについて説明する。図7(a)は input inverter を取り入れた BDD の一部分を表す。なお、前節で説明したように双対ノードを持たないノードであっても input inverter で接続される場合がある。input inverter の働きについて考える。通常のエッジで親ノードと接続されているノードにおいては、入力変数が 0(1) のとき、0(1)-子ノードが選ばれる。一方、input inverter で親ノードと接続されているノードにおいては、入力変数が 0(1) のとき、1(0)-子ノードが選ばれる。この動作を実装するための回路設計手法を以下で説明する。図7(b)に示す回路は図7(a)の BDD を実装するものである。ノード $u(w)$ において、左下の端子は 0-子ノードに対応するノード $a(c)$ に接続される。また、右下の端子は 1-子ノードに対応するノード $b(d)$ に接続される。ここで、通常

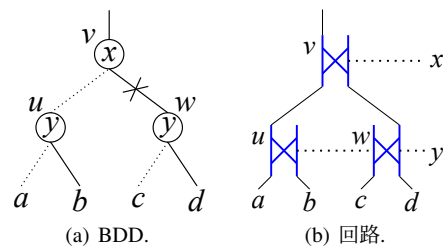


図7 Input inverter の実装.

エッジと input inverter のどちらで親ノードと接続されているかによって、上側の端子のうちどちらを親ノードに接続するかが決まる。ノード u は通常エッジで親ノードと接続されている。この場合には、 $y = 0(1)$ のときにノード $a(b)$ からの光がノード v に到達するように、左上の端子がノード u に接続される。一方、ノード w は input inverter で親ノードと接続されている。この場合には、 $y = 0(1)$ のときにノード $d(c)$ からの光がノード v に到達するように、右上の端子がノード u に接続される。入力変数 x と y に値を代入することで、この回路が論理的に正しいことがわかる。

次に、最適なスプリッタの分割比を求める手法を説明する。2章では、従来の BDD に基づく回路において最適なスプリッタの分割比を挿入損失に基づいて求める手法を説明した。しかし、この手法は双対ノード共有を考慮していない。そこで、双対ノード共有を扱うために、3章で説明した双対ノードにおける挿入損失を求める計算法を2章で説明した最適なスプリッタの分割比を求める手法に組み込む。ノード v の挿入損失は、ノード v が双対ノードを持つかどうかに関わらず、親ノードの挿入損失を調べることでより求めることができる。したがって、双対ノード共有は最適なスプリッタの分割比を求める際の計算量を増加させることはない。まとめると、従来の BDD に基づく合成手法と比べて、双対ノード共有を取り入れた提案合成手法は BDD の最適化でも回路最適化の観点でも計算コストの増加はない。したがって、提案合成手法の計算量は従来の BDD に基づく合成手法と同程度となる。双対ノード共有を取り入れた回路において最適なスプリッタの分割比を求める例を示す。図 8 (a) に input inverter を取り入れた BDD を示す。簡単のために、ここでは入力変数 a でラベル付けされたノードをノード a と呼ぶ。また、ノード b と c についても同様に扱う。初めに、 $loss_a$ は明らかに 1 である。次に、 $loss_b$ は $loss_a$ に A をかけることで得られる。つまり、 $loss_b = A$ とみなせる。次に、ノード c には双対ノード共有を適用することができるため、 $loss_c = A \times \max(loss_a + loss_b) = A^2$ となる。最後に、1-終端の挿入損失 $loss_t = A \times (loss_b + loss_c) = A^3 + A^2$ となる。また、最適なスプリッタの分割比 $SR_b : SR_c = A^3 : A^2$ が得

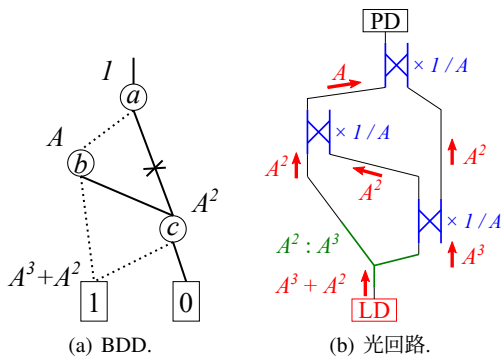


図 8 双対ノード共有を伴う回路のスプリッタの分割比最適化.

られる。図 8 (b) は上記の計算により得られた最適なスプリッタの分割比と消費電力に基づく回路を示す。赤色の矢印は光の伝搬を表す。青色の文字は DC での信号強度の減衰を表す。緑色の文字はスプリッタの分割比を表す。入力光がどのパスを通ったとしても、フォトディテクターで十分な信号強度が得られることがわかる。

5. 実験結果

本章では、実験結果から双対ノード共有による消費電力とノード数の削減の効果について考察する。実験結果から論理関数の入力数の違いによる影響を取り除くために、本稿の実験では入力数が同じ関数に双対ノード共有を適用した。そこで、ISCAS'85 C7552 ベンチマーク回路 [11] に ABC [12] で提供されている LUT ベースの FPGA テクノロジマッパーを適用して得られる回路を用いて実験を行った。この回路変形手法 [7] により、ISCAS'85 ベンチマーク回路などの大規模な回路の消費電力を数桁削減できる。LUT の入力数を 10 に設定したため 1 から 10 入力の関数が見つかる。それらのうちで 10 入力の関数の実験結果を示す。光論理回路の性能評価において、DC での光の減衰は -1 dB とし [13]、フォトディテクターで検出可能な最小の信号強度は $10 \mu\text{W}$ とする。従来の BDD に基づく回路と双対ノード共有を適用した回路を比較して双対ノード共有の効果を確認する。BDD に基づく光回路の消費電力は BDD の変数順序に大きく依存するため、双対ノード共有の効果を最大限に引き出すために全ての変数順序について調べた。

図 9 に消費電力を最小化する変数順序を選んだ場合の消費電力の結果を示す。"Conventional" は従来の BDD に基づく回路の結果を表し、"Dual Share" は双対ノード共有を取り入れた回路の結果を表す。それぞれの関数に "Conventional" の結果において消費電力が小さい順に 1 から番号を与える。"Function number" はそれぞれの関数に与えられた番号を表す。"Conventional" の結果から、従来の BDD に基づく

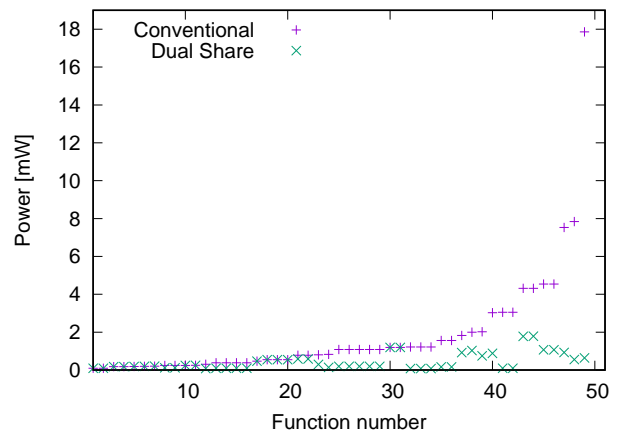


図 9 消費電力の結果.

回路では関数によって消費電力が大きく異なることがわかる。最小の消費電力は $93.1 \mu\text{W}$ であり、これは実現可能な全ての 10 入力関数の中で最小の値である。この関数の回路中にはスプリッタが存在しないため、双対ノード共有により消費電力を削減することはできない。同様に、回路中に存在するスプリッタの数が少ない関数の消費電力は、双対ノード共有によって大幅に削減することはできない。しかし、そのような関数においてはそもそも従来の BDD に基づく回路でも消費電力が大きくなることはない。1 から 22 番の関数の消費電力について考える。双対ノード共有により消費電力を大幅に削減することはできていない。しかし、従来の BDD に基づく回路の消費電力が小さいため、これらの関数は BDD に基づく光論理回路の実用性の課題となるものではない。なお、本研究の目的は従来の BDD に基づく回路で実装した場合に消費電力が膨大になる関数の消費電力を削減することである。40 から 49 番の関数の消費電力について考える。双対ノード共有により大幅に消費電力が削減されている。これはこれらの関数の消費電力が大きくなっている原因が双対ノードに対応するスプリッタにあることを意味している。最大の消費電力は 17.9 mW であり、これは双対ノード共有により $643 \mu\text{W}$ まで削減される。つまり、双対ノード共有はハードウェアとソフトウェア両方の観点で複雑な実装を必要としないが、ISCAS'85 C7552 ベンチマークのような一般的な回路の消費電力を大幅に削減可能であると言える。ノード数の結果について考える。なお、"Conventional"と"Dual Share"の結果の変数順序は異なる場合がある。したがって、"Dual Share"のノード数は"Conventional"と比べて大きくなる場合がある。しかし、このような結果は 5 つの関数のみでしか得られなかった。そのうちの 4 つの関数では、ノード数の増加は 1 または 2 であった。残りの 1 つの関数では、ノード数が 12 増加した。したがって、回路設計制約に回路面積が含まれる場合には、変数順序の最適化においてノード数も考慮する必要がある。

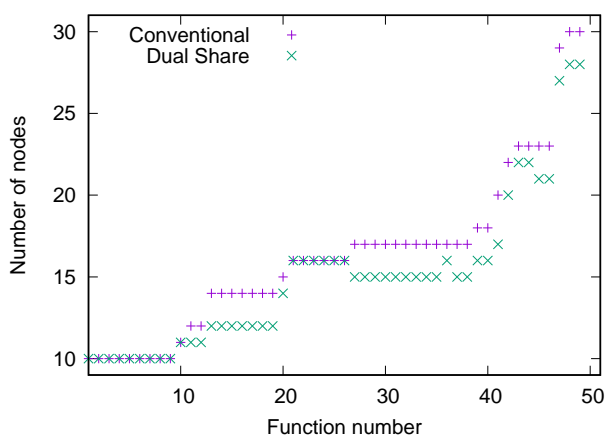


図 10 ノード数の結果。

次に、ノード数を最小化する変数順序を選んだ場合の結果について述べる。全ての関数において、"Dual Share"の消費電力は"Conventional"よりも大きくなることはなかった。図 10 にノード数の結果を示す。それぞれの関数に"Conventional"の結果においてノード数が小さい順に 1 から番号が与える。"Function number"はそれぞれの関数に与えられた番号を表す。ノード数の削減量は消費電力の削減量と比べて非常に小さいことがわかる。これは 3 章で述べたことと一致する。したがって、双対ノード共有は消費電力を削減することに適した手法であると言える。

6. 結論

BDD に基づく光論理回路において、XOR のように回路中にスプリッタが多数存在する場合には消費電力が非常に大きくなることを示した。そこで、本稿では回路中の DC において使用されていない双対端子を用いてスプリッタを削除する手法として双対ノード共有を提案した。さらに、BDD に基づく論理合成に双対ノード共有を取り入れた場合の合成手法を提案した。また、提案合成手法は従来の BDD に基づく合成手法と比べて計算コストが増加しないことを示した。ISCAS'85 ベンチマーク回路を用いた実験から、BDD に基づく光論理回路で実装した場合に消費電力が大きくなる関数の消費電力が、双対ノード共有により大幅に削減されることを示した。つまり、双対ノード共有は BDD に基づく光論理回路の実用性の課題を克服する鍵となると言える。今後は、光論理回路中のスプリッタの数をさらに削減する手法について研究する予定である。

謝辞

本研究の一部は、JST CREST JPMJCR18K3 と JSPS 科研費 JP21J12765 の助成により行われた。

参考文献

- [1] A. Ceyhan, M. Jung, S. Panth, S. K. Lim, and A. Naemi, "Impact of Size Effects in Local Interconnects for Future Technology Nodes: A Study Based on Full-Chip Layouts," in *Proc. IEEE Interconnect Technology Conference/Advanced Metallization Conference*, May 2014, pp. 345–348.
- [2] Y. Vlasov, "Silicon Photonics for Next Generation Computing Systems," in *Proc. European Conference on Optical Communications*, no. Tu.1.A.1, Sept. 2008.
- [3] N. Yamamoto, T. Ogawa, and K. Komori, "Photonic Crystal Directional Coupler Switch with Small Switching Length and Wide Bandwidth," *Optics Express*, vol. 14, no. 3, pp. 1223–1229, Feb. 2006.
- [4] R. Wille, O. Kesocze, C. Hopfmuller, and R. Drechsler, "Reverse BDD-based Synthesis for Splitter-Free Optical Circuits," in *Proc. Asia-South Pacific Design Automation Conference*, Jan. 2015, pp. 172–177.
- [5] Z. Zhao, Z. Wang, Z. Ying, S. Dhar, R. T. Chen, and D. Z. Pan, "Logic Synthesis for Energy-Efficient Photonic Integrated Circuits," in *Proc. Asia-South Pacific Design Automation Conference*, Jan. 2018.

- [6] R. Matsuo, J. Shiomi, T. Ishihara, H. Onodera, A. Shinya, and M. Notomi, “BDD-based Synthesis of Optical Logic Circuits Exploiting Wavelength Division Multiplexing,” in *Proc. Asia-South Pacific Design Automation Conference*, Jan. 2019, pp. 355–360.
- [7] R. Matsuo, J. Shiomi, T. Ishihara, H. Onodera, A. Shinya, and M. Notomi, “A synthesis method for power-efficient integrated optical logic circuits towards light speed processing,” in *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, July 2020, pp. 488–493.
- [8] H. C. Nguyen, N. Yazawa, S. Hashimoto, S. Otsuka, and T. Baba, “Sub-100 μm Photonic Crystal Si Optical Modulators: Spectral, Athermal, and High-Speed Performance,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 19, no. 6, pp. 127–137, June 2013.
- [9] R. Matsuo, S. Minato, “BDD Variable Ordering for Minimizing Power Consumption of Optical Logic Circuits,” in *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, July 2021, pp. 96–101.
- [10] S. Minato, N. Ishiura, and S. Yajima, “Shared Binary Decision Diagram with Attributed Edges for Efficient Boolean Function Manipulation,” in *Proceedings of the 27th ACM/IEEE Design Automation Conference*, ser. DAC '90, New York, NY, USA, 1991, pp. 52–57.
- [11] B. F. and F. H., “A neural netlist of ten combinational benchmark circuits and translator in Fortran,” in *Int. Symp. Circuits and Systems (ISCAS)*, June 1985, pp. 663–698.
- [12] Robert Brayton and Alan Mishchenko, “ABC: An Academic Industrial-Strength Verification Tool,” in *Proc. of International Conference on Computer Aided Verification*, 2010, pp. 24–40.
- [13] N. Yamamoto, T. Ogawa, and K. Komori, “Photonic crystal directional coupler switch with small switching length and wide bandwidth,” *Optics Express* 14, Feb 2006.