C-007

反転条件緩和による低消費電力指向バス反転回路の設計
A Design of Low Power Oriented Bus Inverting Circuits with Loose Inversion

稲岡　智哉[1]，　市原　英行[1]，　岩垣　剛[1]，　井上　智生[1]
Tomoya INAOKA, Hideyuki ICHIHARA, Tsuyoshi IWAGAKI, Tomoo INOUE

1　はじめに

近年，携帯電話やノート PC などの LSI を利用したポータブルデバイスの普及に伴い，低消費電力の LSI が求められており，LSI の低消費電力化の研究が行われている．その中で複数のデバイスを接続するバス (データバス，アドレスバス，コントローラバスなど) も消費電力が比較的大きいことから低消費電力化の研究の対象とされており，バスの低消費電力化手法として，バスの信号値を反転することによるトグル削減手法 [1, 2] やバスの信号値を符号化することによるトグル削減手法 [3] などが提案されている．例えば文献 [1] のバス反転法では，バスに流れる信号値を適切に反転するバス反転回路をバスの送信側に，そして反転された信号値を元に戻すための回路をバスの受信側に挿入し，バスに流すベクトル系列において連続する 2 ベクトルのハミング距離がバス幅 W の半分以下になるように, ベクトルを適切に反転を行う．これによりバスのトグル数を削減することができ，バスの低消費電力化ができる．

このようなバスの低消費電力化は，エラートレラントアプリケーションの 1 つであると考えられる．エラートレラントアプリケーションとはそのアプリケーションを実装する回路の出力誤りに対する許容性が高いアプリケーションのことである [4, 5, 6, 7, 8]．主なアプリケーションとしては，動画像や音声を処理して出力するアプリケーション [9] や，プロセッサの分岐予測や今回対象としている低消費電力化のように出力がベストエフォート型のアプリケーションが挙げられる [4]．バスの低消費電力化 [1] はパフォーマンスを向上する (電力量を削減する) ことを目的とした処理であり，この処理が正しく行われなかったとしても，データを正しく転送するというバスの本来の機能は損なわれない．例えば，バス反転回路において，連続する 2 ベクトルのハミング距離の計算結果に誤りが生じ，バスの反転の判定が間違った場合でも，バスのトグル数 (つまり，消費電力) の削減効果が低下するだけで，バスの転送機能には影響がない．つまり，この回路は出力誤りに対する許容性が高く，バスの低消費電力化はエラートレラントアプリケーションである．エラートレラントアプリケーションを実現する論理回路においては，その誤りに対する許容性を利用して，許容範囲内で回路が簡単化できるように設計する方法が研究されている [6, 7]．

本研究では，バス反転による低消費電力化が持つ誤りに対する許容性を利用して，バスの反転条件を緩和することでバス反転回路を簡単化することを提案する．提案するバス反転回路の簡単化では，一般的な冗長な論理簡単化とは異なり，簡単化後のバス反転回路が簡単化前にそれとは異なる値を出力するように設計する．ただしこのとき，バス反転による低消費電力化がエラートレラントアプリケーションであることを踏まえて，この 2 つの回路の出力の差分をエラー (誤り) と考え，その誤りが許容できる範囲で設計を変更する．この結果，バスの低消費電力化の効果を大きく損なうことなく，回路の論理素子数を大幅に減らす事ができる．さらに，簡単化後のバス反転回路は，使用する論理素子数が減少するために，そのバス反転回路が消費する電力も削減することができる．結果として，バス反転回路の簡単化は，バスの低消費電力化の効果を多少損なうもののバス反転回路自身の低消費電力化ができることにより，バス反転回路の簡単化前に比べて，バスとバス反転回路の両方を含んだ総消費電力を削減できることが期待できる．

以下では，バス反転法について簡単に述べた後，バス反転回路の設計例を示し，反転条件を緩和したときのバス反転回路の回路簡単化法を提案する．そして，計算機実験によりバスの簡単化の効果と，消費電力に与える影響について実験的に考察する．

2　バス反転によるトグル数削減手法

2.1　バス反転とバスのトグル数削減

バスの消費電力は，バスの信号値のトグル（0 から 1，または，1 から 0 に変化すること）の回数に比例する．表 1 の 1 列目に示すように，バス幅 8 ビット ($W = 8$) のデータバスに流す信号値（ベクトル）系列 $A : a_0, a_1, \ldots, a_9$ を考える．表 1 の 2 列目にはベクトル系列 A の現時刻のベクトル a_t と前時刻のベクトル a_{t-1} のハミング距離 $dis(a_t, a_{t-1})$ を示している．このとき，例えば，ベクトル a_0 と a_1 のハミング距離 $dis(a_0, a_1)$ は 7 であるため，ベクトル a_0 後に a_1 をバスに流した場合，7 本の信号線の値がトグルすることになる．バスが消費する電力はトグル数に比例するため，このベクトル系列をそのままバスに流した場合，バストグル数 T_B は 49 となり，これに比例した電力が消費されることになる．

バス反転によるトグル数削減手法（バス反転法）[1] は，バスに流すベクトル系列における時刻 $t-1$ と t のベクトル間のハミング距離を計算し，ハミング距離がバス幅 W の半分よりも大きければ，時刻 t の信号値を反転する手法である．表 1 の 3 列目にハミング距離が $W/2 = 4$ 以下（$dis(b_t, b_{t-1}) \leq 4$）になるように反転したベクトル系列 B を示す．ベクトル系列 B では，ベクトル系列 A の a_1, a_3, a_4, a_7, a_9 を反転しており，表の 4 列目に示すようにベクトル間のハミング距離は必ず 4 以下になっている（括弧の中の数字は削減数）．例えば，時刻 1 では前のベクトル $b_0 = (01101111)$ とベクトル $a_1 = (11010000)$ のハミング距離が 7 であるため，ベクトル a_1 を反転したベクトル $b_1 = (00101111)$ をバスに流すことで，ハミング距離は 1 となり 6 回のトグルを削減している．また，

[1]広島市立大学大学院情報科学研究科

表 1: ベクトル系列とバストグル数.

t	ベクトル系列 A	ハミング距離 $dis(a_t, a_{t-1})$	ベクトル系列 B $dis(b_t, b_{t-1}) \leq 4$	ハミング距離 $dis(b_t, b_{t-1})$	ベクトル系列 C $dis(c_t, c_{t-1}) \leq 6$	ハミング距離 $dis(c_t, c_{t-1})$
0	a_0: 01101111		b_0 : 01101111(a_0)		c_0 : 01101111(a_0)	
1	a_1: 11010000	7	b_1 : 00101111($\overline{a_1}$)	**1 (-6)**	c_1 : 00101111($\overline{a_1}$)	**1 (-6)**
2	a_2: 01101011	6	b_2 : 01101011(a_2)	**2 (-4)**	c_2 : 01101011(a_2)	**2 (-4)**
3	a_3: 00001100	5	b_3 : 11110011($\overline{a_3}$)	**3 (-2)**	c_3 : 00001100(a_3)	5
4	a_4: 01000000	3	b_4 : 10111111($\overline{a_4}$)	3	c_4 : 01000000(a_4)	3
5	a_5: 10111111	8	b_5 : 10111111(a_5)	**0 (-8)**	c_5 : 01000000($\overline{a_5}$)	**0 (-8)**
6	a_6: 00111000	4	b_6 : 00111000(a_6)	4	c_6 : 00111000(a_6)	4
7	a_7: 11100111	7	b_7 : 00011000($\overline{a_7}$)	**1 (-6)**	c_7 : 00011000($\overline{a_7}$)	**1 (-6)**
8	a_8: 00001111	4	b_8 : 00001111(a_8)	4	c_8 : 00001111(a_8)	4
9	a_9: 11011100	5	b_9 : 00100011($\overline{a_9}$)	**3 (-2)**	c_9 : 11011100(a_9)	5
総バストグル数 T_B		49		21		25

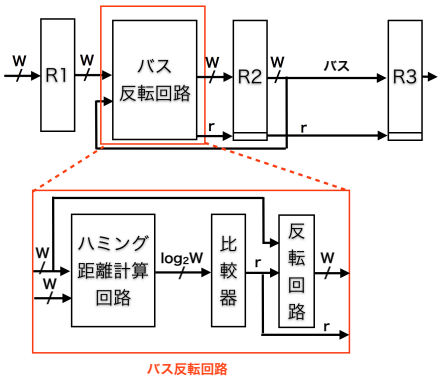


図 1: バス反転回路

時刻 2 では，ベクトル b_1 がベクトル a_1 の反転ベクトルであることから，ベクトル a_2 とのハミング距離が 2 となっているため，ベクトル a_2 をそのままベクトル b_2 としている．このように連続する 2 ベクトルのハミング距離に応じて反転を行うことで，ベクトル系列 B によるバスの総トグル数 T_B は 21 となり，バスの消費電力削減率は $21/49 \simeq 43\%$ となる．

バス反転法では，図 1 に示すように，バスに流れる信号値を適切に反転するバス反転回路をバスの送信側に，そして反転した信号値をもとに戻すための反転回路をバスの受信側に挿入する．また，信号値が反転しているのかどうかを示す反転信号 r をバスの信号値とは別に用意する（つまり，$W+1$ ビットのベクトルを実際には送信することになる）．このため，反転信号 r のトグル率も消費電力に影響することになる．例えば，表 1 のベクトル系列では反転信号は 7 回トグルするため，最終的な反転信号まで含めたバスの総トグル数は $21+7=28$ となる．以下の議論では反転信号 r のトグル数については考慮しないが，最後に示す実験結果ではこの反転信号のトグル数も含めてバストグル数を計算している．

3 反転条件の緩和によるバス反転回路の簡単化

バス反転法によるバスの低消費電力化はエラートレラントプリケーションであるため，それを実現するバス反転回路はその誤りに対する許容性を利用して回路簡単化することができる．バス反転法では，誤りに対する許容性を反転条件の緩和として表現する．まず 3.1 で一般的なバス反転回路を紹介する．次に 3.2 でそのバス反転回路を用いてバス反転法におけるバス反転回路の簡単化について考察する．バス反転回路の簡単化はバスを反転する条件を緩和し，バスの多少の消費電力の増加を許容することで行う．

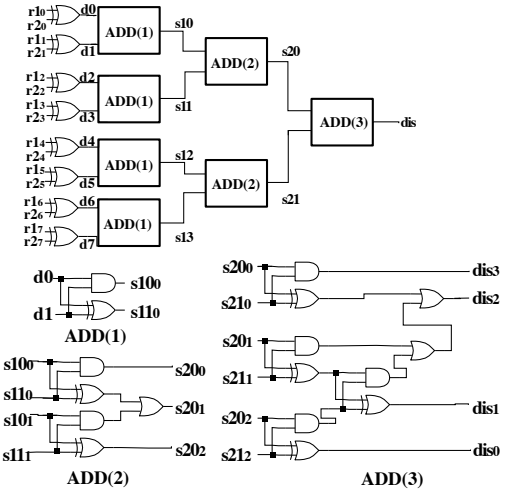


図 2: 8 ビットハミング距離計算回路

3.1 バス反転回路

バス反転の条件は次のように表現できる．

[バス反転の条件] バス幅 W と，ベクトル $r1$ と $r2$ のハミング距離 $dis(r1, r2)$ に対して，

- $dis(r1, r2) \leq W/2$ ならば，反転しない（$r=0$).
- $dis(r1, r2) > W/2$ ならば，反転する（$r=1$）.

バス反転回路の実現方法は様々な方法があるが [1]，ここではハミング距離計算回路，比較器，反転回路の 3 つの組合せ回路からなる設計を考える．なお，簡単化の対象とするのはハミング距離計算回路と比較器であるため，ここではこの 2 つの回路について説明する．

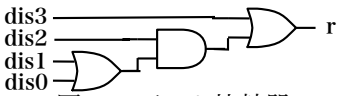


図 3: 8 ビット比較器

ハミング距離計算回路

ハミング距離計算回路は，入力された 2 つのベクトル $r1$ と $r2$ のハミング距離 $dis(r1,r2)$ を計算する回路である．バス幅 W（以下では，W は 2 のべき乗と仮定）のハミング距離計算回路は，2 つのベクトルの差分ベクトルを計算する W 個の排他的論理和（EXOR）ゲート列と，その差分ベクトル内の論理値 1 の数を数える加算器からなる．加算器は部分加算器 ADD を頂点とする高さ $\log_2 W$ の 2 分木構造をしており，k 段目の $ADD(k)$（入力側を 1 段目とする．$1 \leq k \leq \log_2 W$）は，k ビット加算器である．

図 2 は，部分加算器 ADD を桁上げ加算器 (RCA) で構成した $W = 8$ のハミング距離計算回路を示している．入力ベクトル $r1 = (r1_0, r1_1, \ldots, r1_7)$ と $r2 = (r2_0, r2_1, \ldots, r2_7)$ とすると，EXOR ゲート列で $R1$ と $R2$ の差分ベクトル $d = (d_0, d_1, \ldots, d_7)$ を求める．ここで $d_i = r1_i \oplus r2_i$ である．

後半のツリー型加算器は，まず差分ベクトル d の隣り合う 2 ビット d_i と d_{i+1} $(i = 0,2,4,6)$ を 1 段目の 1 ビット加算器 $ADD(1)$ で足しあわせ，出力 $s1i = (s1i_0, s1i_1)$ として出力する．次に，$s1j$ と $s1(j+1)$ $(j = 0,2)$ を $ADD(2)$ で加算し，3 ビット出力 $s2j = (s2j_2, s2j_1, s2j_0)$ を得る．最後に，$s20$ と $s21$ を 3 ビット加算器 $ADD(3)$ で足しあわせて，ハミング距離 $dis = (dis_3, dis_2, dis_1, dis_0)$ を得る．

表 2(a) に図 2 の 3 段目 $ADD(3)$ の真理値表を示す．加算器の入力 $s20$ と $s21$ は入れ換え可能であるため，入れ換えた場合の出力値は省略している．また，信号値はすべて 10 進数で表記している．この真理値表が示すように入力 $s20$ と $s21$ の変域は 0 から 4 までの範囲であるため，図 2 に示したように，$ADD(3)$ は一般的な 3 ビット加算器よりも簡単な構造となっている．

比較器

比較器は計算されたハミング距離 $dis(r1,r2)$ がバス幅 $W/2$ より大きいとき，反転符号 $r = 1$ を，バス幅 $W/2$ 以下であれば $r = 0$ を出力する定数比較器である．表 4(a) はバス幅 W が 8 のときの比較器の真理値表を示す．この真理値表に基づく論理回路を図 3 に示す．

3.2 簡単化の方針

バス反転法では，連続するベクトル間のハミング距離が大きいと反転の効果（トグル数の削減効果）は大きく，ハミング距離が $W/2$ に近づくにつれて反転の効果は小さくなる．例えば，ハミング距離が W の場合は現時刻のベクトルを反転することでバスのトグル回数を 0 にできるが，ハミング距離が $W/2+1$ のときは，現時刻のベクトルを反転したとしてもバスのトグル数は $W/2-1$ となり，その差は 2 である．

そこで，ハミング距離が $W/2$ よりも大きくても $W/2$ に近いとき，つまり反転することの効果が小さいときには，反転を必ずしもしなくてもよいように反転の条件を緩和することにする．表 1 の 5, 6 列目に，ハミング距離

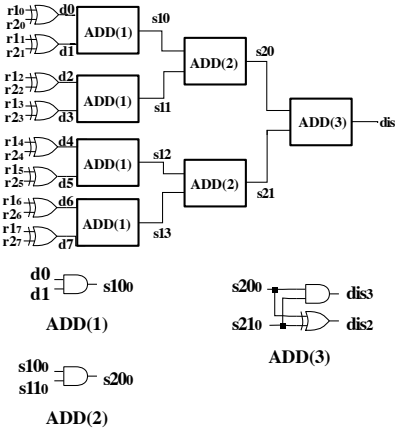


図 4: 反転の条件を緩和して簡単化したハミング距離計算回路

$dis(c_t, c_{t-1})$ が $W/2+2$ 以下（$dis(c_t, c_{t-1}) \leq 6$）になるように反転したベクトル系列 C と，そのベクトル間ハミング距離 $dis(c_t, c_{t-1})$ を示している．ベクトル系列 C では，時刻 3 と時刻 9 においてそれぞれの前時刻とのベクトル間ハミング距離が 5 であるため反転していない．この結果，バスのトグル数 T_B は 25 に増え，トグル数の削減率は $25/49 \simeq 51\%$ となり，簡単化前に比べるとわずかな増加（簡単化前は 43%）となっている．

一方で，このバス反転条件の緩和は，バス反転回路のハミング距離計算回路や比較器の簡単化につながる．図 4 は，同様な反転の条件緩和により簡単化した 8 ビットハミング距離計算回路である．この簡単化で，論理ゲート数を 28 から 16 に削減できる．このように，バスのトグル数のわずかな増加で，バス反転回路の論理ゲート数を半分以下にすることができる．

反転条件の緩和を考えるために，次のように反転の条件を再定義する．

[緩和を考慮したバス反転の条件] バス幅 W と，ベクトル $r1$ と $r2$ のハミング距離 $dis(r1,r2)$ に対して，

- $dis(r1,r2) < W/2$ ならば，反転しない（$r = 0$).
- $W/2 \leq dis(r1,r2) \leq W/2+d$ ならば，反転してもしなくてもよい（$r = X$(ドントケア)）.
- $dis(r1,r2) > W/2+d$ ならば，反転する（$r = 1$）.

変数 d は条件の緩和の程度を表現するパラメータであり，出力ドントケアの範囲 $W/2 \leq dis(r1,r2) \leq W/2+d$（以下では，変数 d を出力ドントケアの範囲と呼ぶ）を表している．$d = 0$ の場合は，ハミング距離が $W/2$ のときのみ反転の有無をドントケアにすること $(r = X)$ を意味しており，d を大きくするほど出力ドントケアの範囲は広がり（条件は緩和され），その結果バス反転回路は簡単化できることになる．例えば表 1 に示した条件緩和の例は出力ドントケアの範囲が $d = 2$ に相当し，ベクトル系列 B では反転を行っていた時刻 3 と 9 では，ベクトル系列 C では反転を行っていない．なお，時刻 2 では $dis(a_2, a_1) = 6$ であるが，ベクトル系列 B や C では時刻 1 のベクトルが反転しているため前時刻とのハミング距

離は 2 となっているため，出力ドントケアの範囲には含まれないと考えている．

バス反転の条件緩和によるバスのトグル数の増加は，連続する 2 ベクトルのハミング距離が出力ドントケアの範囲 d に含まれる場合が多いほど大きくなるため，与えられるベクトル系列がこのような場合をできるだけ含まないものであれば，バス反転回路の簡単化の効果は大きくなる．例えば，表 1 に示したベクトル系列 A において連続する 2 ベクトル間のハミング距離が 5 および 6 のものが 1 つも存在しなければ，ドントケア出力の範囲 $d=2$ で条件を緩和したとしても，バスのトグル数の増加はないことになる．

また，バスの判定条件の緩和により簡単化したバス反転回路は消費する電力も減るため，バスのトグル数が増えてバスの消費電力が増加したとしても，バス反転回路とバスの両方をあわせた消費電力は削減する可能性がある．4.2 の計算機実験では，そのようなケースが存在することを示す．

3.3 バス反転回路の簡単化

出力ドントケアの範囲 d が与えられたとき，このバス反転回路を簡単化する方法を提案する．提案するバス反転回路の簡単化は，比較器とハミング距離計算回路を対象とする．

3.3.1 比較器の簡単化

比較器の論理関数は上で示した [緩和を考慮したバス反転の条件] そのものであるため，この条件を満たす比較器の論理のうち，できるだけ論理素子数が少なくなるようにな設計法を考える．一般的には論理関数の入力変数数（リテラル数）が少ないほど論理素子数が少ない設計になると考えられるため，出力ドントケアを入力数が減るように 0 または 1 に設定する．これは言い換えれば，r の論理関数を最小積和標準形で表現したときに，もっとも相乗項数とリテラル数が少なくなるようにドントケアを決めることになる．具体的には，$W/2+d+1$ を超えない最大の 2 のべき乗を 2^M とすると[1]，比較器の論理を

- $dis(r1,r2) < 2^M$ ならば，$r=0$.
- $dis(r1,r2) \geq 2^M$ ならば，$r=1$.

とすれば，入力変数数が最も小さくなるようにドントケアを 0 または 1 に決定したことになる．

例えば，$W=8, d=2$ の場合を考えると比較器の論理関数は表 4(b) に示したように出力値にドントケアを含んだものとなる．このとき，$W/2+2=6$ を超えない最大の 2 のべき乗は $4=2^2$ であるため，ハミング距離 $dis(r1,r2)$ が 4 以上のときは 1，それ以外のときは 0 となるように（つまり出力のドントケアをすべて 1 になるように）設計すると，真理値表は表 4(c) のようになる．この結果，上位 2 ビットの dis_3 と dis_2 だけで出力が決まることになる．よって，簡単化した比較器は図 5 のような回路となり，簡単化前の図 3 の比較器よりも論理素子数が少ないことがわかる．

以上をまとめると，比較器の簡単化は次のようになる．

[1] $M = \lfloor \log_2(W/2+d+1) \rfloor$ となる．

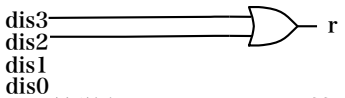


図 5: 簡単化した 8 ビット比較器

表 2: $ADD(3)$ の真理値表

(a) 簡単化前			(b) 入力 $s2i_0$ 削除時			(c) 入力 $s2i_0, s2i_1$ 削除時		
$s20$	$s21$	dis	$s20'$	$s21'$	dis	$s20''$	$s21''$	dis
0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	2	2	0	2	2	0	0	0
1	1	2	0	0	0	0	0	0
0	3	3	0	2	2	0	0	0
1	2	3	0	2	2	0	0	0
0	4	4	0	4	4	0	4	4
1	3	4	0	2	2	0	0	0
2	2	4	0	2	2	0	0	0
1	4	5	0	4	4	0	4	4
2	3	5	2	2	4	0	0	0
2	4	6	2	4	6	0	4	4
3	3	6	2	2	4	0	0	0
3	4	7	2	4	6	0	4	4
4	4	8	4	4	8	4	4	8

[比較器の簡単化] バス幅 W と出力ドントケアの範囲 d が与えられたとき，$W/2+d$ を超えない最大の 2 のべき乗 2^M を求め，比較器の入力下位 M ビットを削除する．

3.3.2 ハミング距離計算回路の簡単化

ハミング距離計算回路の簡単化も，構成要素である各部分加算器 $ADD(k)$ の入力信号線数（入力変数数）を削減することで，論理を簡単化することを考える．

ハミング距離計算回路では，簡単化によって出力結果が変化しても，後に続く比較器の出力 r が反転の条件にあっていればその簡単化は許容できる．比較器は 3.2.1 で述べたように 2^M が出力 r の閾値となるように簡単化を行っているため，ハミング距離計算回路の論理関数は次のように考えることができる．

- $dis(r1,r2) < W/2$ となるような入力に対しては，出力は 2^M より小さい任意の値.
- $W/2 \leq dis(r1,r2) \leq W/2+d$ となるような入力に対しては，出力はドントケア.
- $dis(r1,r2) > W/2+d$ となるような入力に対しては，2^M 以上の任意の値.

この条件の満たす範囲内で，入力変数をできるだけ削減することを考える．表 2(b), (c) には，$W=8$ のときの $ADD(3)$ の 2 つの入力 $s20$ と $s21$ の最下位ビット（$s20_0$ と $s21_0$）を削除した場合（削除後の入力は $s20'$ と $s21'$）と，下位 2 ビット（さらに $s20_1$ と $s21_1$）を削除した場合（削除後の入力は $s20''$ と $s21''$）の入力値の変化と，それに伴う出力値の変化（dis' と dis''）を示している．それぞれの表には，上で示したハミング距離計算回路の論理関数に関する 3 つの範囲を，$d=2$ の場合について示している．つまり，表 2(a) に示す dis が 3 以下の範囲，4 以上 6 以下の範囲，7 以上の範囲の 3 つである．これを見

てわかるように下位 2 ビットの削除を行っても，$W = 8$, $d = 2$ の場合は上記のハミング距離計算回路の論理関数を満たすことがわかる．

以上のことからハミング距離計算回路の各加算器 $ADD(k)$ の入力のうち，削減できる最大の下位入力数は W と d により一意に求まることがわかり，削減できる最大の下位入力数は $\lfloor \log_2(d+2) \rfloor - (\log_2 W - k)$ と表現できる．よって，ハミング距離計算回路の各部分加算器 $ADD(k)$ の入力信号線を削除することによる簡単化は次のようになる．

[ハミング距離計算回路の論理簡単化] バス幅 W と条件緩和の範囲 d が与えられたとき，k 段目の $ADD(k)$ の入力の下位 $\lfloor \log_2(d+2) \rfloor - (\log_2 W - k)$ ビットを削除する．

なお，表 2(a) において，$a20$ と $a21$ の和（ハミング距離）が同じであっても，表 2(b) や (c) に示すように簡単化後ハミング距離計算回路では出力結果が異なる場合がある．例えば，$(a20, a21) = (2,4)$ のときと $(a20, a21) = (3,3)$ のときは簡単化前のハミング距離計算回路の出力はどちらも 6 であるが，簡単化後は前者は 4 で後者は 0 となっている．結果，同じハミング距離でも反転するかどうかは入力に依存し，この例では前者は反転信号 r が 1 となり後者は 0 となる．

3.4 バス反転回路の簡単化アルゴリズム

これまでに述べた比較器とハミング距離計算回路の簡単化方法をまとめると，バス反転回路のアルゴリズムは次のように表すことができる．

[バス反転回路の簡単化アルゴリズム]
入力：バス幅 W，出力ドントケアの範囲 d，バス反転回路 C

(1) 3.2.1 で述べた C の比較器の簡単化を行う．

(2) $D(k) \equiv \lfloor \log_2(d+2) \rfloor - (\log_2 W - k)$ とする．

(3) $k = \log_2 W$ とし，$D(k) > 0$ の間以下の処理を繰り返す．ただし，繰り返す毎に k は 1 ずつ減少させる．

(3-1) $ADD(k)$ で出力が削除されているゲートと関連する信号線を削除．

(3-2) 3.2.2 で述べた $ADD(k)$ の下位 $D(k)$ ビットの入力を削除し，関係するゲートと信号線を削除．

このアルゴリズムはまず，3.2.1 で説明した比較器の簡単化を行い，それに伴ってハミング距離計算回路の最上段の部分加算器 $ADD(\log_2 W)$ の不要な論理を削減する．その後，3.2.2 で述べた方法で改めて $ADD(\log_2 W)$ の簡単化を行い，それに伴って不要となる次段の部分加算器 $ADD(\log_2 W - 1)$ の論理を削減する．以下，同様な手順で部分加算器の簡単化を繰り返し，部分加算器の簡単化ができなくなれば終了する．

4 実験

4.1 バス反転回路の面積とバストグル数に関する実験と考察

提案した簡単化手法によるバス反転回路の面積削減の効果を調べるために，バス幅 W を 32，64 としたときをバス反転回路を設計した．そして，それぞれの回路を出力ドントケアの範囲 d を 0，2，6 として，提案したバス反転回路簡単化アルゴリズムで簡単化した．また，反転の条件を緩和したことによりバストグル数が増加することが予想されるため，計算機シミュレーションによりバストグル数を調べた．シミュレーションには 4 種類の JPEG, BMP 画像ファイル（lena.jpg, pica.jpg, pict.jpg, miku.bmp）と 4 種類の Mach-O 64 ビット実行ファイル（fs, mv_cover, sample, short）を用いた．

表 4: 8 ビット比較器の真理値表

(a) 簡単化前		(b)$d = 2$ のとき	(c) 簡単化後
$(dis_3, dis_2, dis_1, dis_0)$	r	r	r
0000	0	0	0
0001	0	0	0
0010	0	0	0
0011	0	0	0
0100	0	X	1
0101	1	X	1
0110	1	X	1
0111	1	1	1
1000	1	1	1

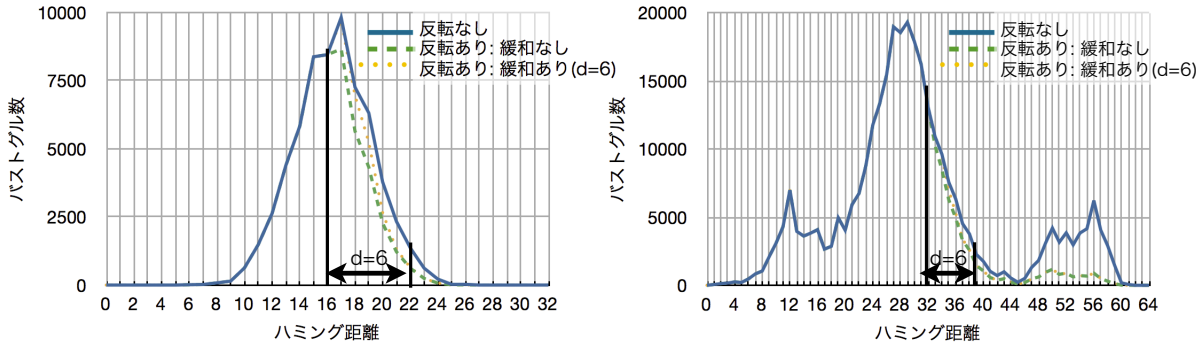
表 3 に，バス幅 $W = 32, 64$ に対する，簡単化前（「緩和無し」）と簡単化後（「緩和 (d=0)」「緩和 (d=2)」「緩和 (d=6)」）のバス反転回路の面積とバストグル数の削減率を示す．回路面積は NOT ゲートのサイズを 1 として計算しており，3 列目にその値を示している．括弧の中には「緩和無し」のバス反転回路の面積を 1 としたときのそれぞれの面積の比率を示している．また，バストグル数の削減率は，バス反転法を用いない場合のバストグル数に対する用いた場合のトグル数の割合として示している．

表 3 からわかるように，バス反転回路面積は d が増加するほど小さくなっており，$W = 32, d = 6$ では 40%，$W = 64, d = 6$ では，20%の面積削減が可能となっている．一方でバスのトグル数の増加は $W = 32$ の場合では高々5 ポイント，$W = 64$ の場合では 3 ポイント程度の増加にとどまっている．このことからバスのトグル数をほとんど増加させることなく，バス反転回路の面積を大幅に削減することができることがわかる．

表 3 を見ると，ベクトル系列 lena.jpg よりも fs のほうが（特に $W = 64$ のとき）バス反転法の効果が高く，さらに条件を緩和したとしてもバストグル数の増加が小さいことがわかる．これは 3.1 で述べたように，fs のほうが，$W/2$ よりも大きなハミング距離をもつ連続 2 ベクトルを多く持っており，かつ，出力ドントケアの範囲に属するハミング距離を持つものは少ないためと考えられる．そこで，lena.jpg と fs において，それぞれのベクトル系列における連続 2 ベクトルハミング距離に対するバストグル数の分布を調べた．図 6 に lena.jpg ($W = 16$) と fs ($W = 32$) の系列のバストグル数の分布（「反転無し」）と，2 つのバス反転法「緩和無し」と「緩和 (d=6)」を行った後の分布の変化も併せて示す．例えば図 6(a) は lena.jpg の分布では，前時刻とのハミング距離が 19 のものは 332 個あるため，$19 \times 332 = 6308$ 回のバストグルを行っていることがわかる．ここで，バス反転法「緩和無し」を行うことによりこの 332 個のベクトルは適切に反転され，前時刻とのハミング距離は 13 となる．よって，バスト

表 3: バス反転回路の面積とバストグル数の削減率

W	バス反転法	バス反転回路面積	バストグル数の削減率 lena.jpg	pica.jpg	pict.jpg	miku.bmp	fs	mv_cover	sample	short
32	緩和無し	1,397(1.00)	89.5%	90.1%	89.2%	89.3%	89.9%	90.1%	89.9%	89.9%
	緩和 (d=0)	1,299(0.93)	89.5%	90.2%	89.2%	89.3%	89.9%	90.2%	89.9%	90.0%
	緩和 (d=2)	1,173(0.84)	90.6%	91.0%	90.2%	90.3%	90.4%	90.7%	91.6%	90.6%
	緩和 (d=6)	879(0.63)	94.0%	94.4%	93.7%	93.7%	92.3%	93.2%	92.7%	93.1%
64	緩和無し	2,937(1.00)	92.2%	90.7%	91.6%	91.8%	86.2%	90.7%	90.4%	90.9%
	緩和 (d=0)	2,825(0.96)	92.2%	90.8%	91.6%	91.8%	86.2%	90.7%	90.4%	90.9%
	緩和 (d=2)	2,699(0.92)	92.7%	91.1%	92.0%	92.2%	86.4%	90.9%	90.6%	91.1%
	緩和 (d=6)	2,405(0.82)	95.7%	92.8%	93.7%	93.8%	88.2%	91.7%	91.4%	91.8%



(a) lena.jpgのバストグル数の分布(W=32)　　(b) fsのバストグル数の分布 (W=64)

図 6: ハミング距離に対するバストグル数の分布

グル数は $13 \times 332 = 4316$ に削減できる．さらにバス反転法「緩和 ($d = 6$)」を行うと，332 個のベクトルのうち 180 個は前時刻とのハミング距離が 13 になるように反転し，残りは 19 のままとするため，バストグル数は 5228 となる．それぞれのグラフにおいて「反転無し」の曲線と各バス反転法の曲線にはさまれている領域の面積がバス反転法の効果を現している．

図 6(a) より，lena.jpg のハミング距離に対するバストグル数の分布はバス反転の効果が少ないハミング距離 16($= W/2$) のあたりにあつまっており，バス反転法の効果が大きく現れる 32($= W$) 付近には存在しない．よって，そもそもバス反転法の効果が小さい（バストグル数の 10%程度の削減）例であることがわかる．また，これはドントケアの範囲 $d = 7$ に分布が集中していることも意味しているため，回路簡単化によってバストグル数の増加がおきやすいベクトル系列であることがわかる．一方で図 6(b) の fs の場合は，ハミング距離が 44〜60 の部分にバストグル数が存在しており，これがバス反転法の効果が高めている（バストグル数を 15%程度削減できる）ことがわかる．また，これらのバストグル数の削減効果は条件緩和に影響を受けない範囲に存在しているため，バス反転回路を簡単化してもバストグル数はあまり増加しないことがわかる．

4.2 総消費電力に関する実験と考察

3.2 で触れたように，バス反転回路の面積削減はバス反転回路自身の消費電力の削減に貢献すると考えられるため，バス反転回路の論理簡単化がバス部分の消費電力を多少増加させたとしても，バスとバス反転回路の消費電力を足しあわせた総消費電力はバス反転回路の簡単化により削減できる可能性がある．バスのトグル数を T_B，バス反転回路の論理ゲートのトグル数を T_L，さらにバス 1 トグルあたりの消費電力を P_B，バス反転回路の論理ゲートにおける 1 トグルの消費電力を P_L とすると，バス反転法による総消費電力 TP は

$$TP = P_B \cdot T_B + P_L \cdot T_L$$

と表現できる．なお一般的には，バスの 1 つの信号線がトグルしたときに消費する電力は，バス反転回路の 1 つのゲートの出力値がトグルしたときに消費する電力より大きいことが想定されるため，$P_B/P_L > 1$ と考えられる．

バス反転回路の簡単化が総消費電力 TP に与える影響を調べるために，バスの信号値のトグル数に加えて，バス反転回路自身の論理ゲートのトグル数を調べるためのシミュレーションを行った．なお，バス反転回路のトグル数として数えているのは，簡単化の対象であるハミング距離計算回路と比較器の論理ゲートのトグル数のみで

表 5: 総消費電力 TP

ベクトル系列	バス反転法	バス反転回路トグル数 T_L	バストグル数 T_B	総消費電力 TP とバス反転回路消費電力比 ρ_L			
				$(P_L,P_B)=(1,50)$		$(P_L,P_B)=(1,100)$	
lena.jpg (W=32)	緩和無し	326,650	57,052	3,179,250	10.3%	6,031,850	5.4%
	緩和 (d=0)	301,060	57,048	**3,153,460**	9.5%	**6,005,860**	5.0%
	緩和 (d=2)	267,864	57,760	3,255,864	8.2%	6,043,864	4.4%
	緩和 (d=6)	192,840	59,966	3,191,140	6.0%	6,189,440	3.1%
fs (W=64)	緩和無し	1,635,165	266,305	14,950,415	10.9%	27,069,111	6.0%
	緩和 (d=0)	1,637,920	266,337	14,954,770	11.0%	**27,036,963**	6.1%
	緩和 (d=2)	1,465,973	266,855	**14,808,723**	9.9%	27,044,220	5.4%
	緩和 (d=6)	1,239,290	272,428	14,860,690	8.3%	27,500,015	4.5%

あり，バスにつながるレジスタとその前後に位置する反転回路はバスの消費電力に含まれると考えている．

表 5 に系列 lena.jpg, 表に系列 fs の結果を示す．それぞれの系列に対して，バス反転回路のトグル数 T_L，バスのトグル数 T_B，そして 2 つの電力比 $P_L:P_B=1:50$ の場合と $P_L:P_B=1:100$ の場合の総消費電力 TP とそのうちバス反転回路の消費電力の割合 $\rho_L=P_L\cdot T_L/TP(\%)$ を示している．条件を緩和しないものと複数の緩和によるバス反転法の結果のうち，最も総消費電力 TP が小さいものを太字で示している．

この結果から lena.jpg, fs ともに，反転条件を緩和することでわずかではあるが総消費電力が下げられる場合があることがわかる．バス反転回路の消費電力が総消費電力に占める割合は，条件を緩和するほど小さくなっていることがわかる．総消費電力が最も小さくなるところが「緩和 ($d=0$)」や「緩和 ($d=2$)」であることを併せて考えると，全く緩和しないときはバス反転回路の消費電力が高いことが原因で，緩和しすぎた場合（$d=6$）はバスのトグル数が増加しすぎることが原因で，総消費電力が大きくなっていると考えられる．

5 まとめ

本研究では，バスの低消費電力化を目的としたバス反転手法に用いられるバス反転回路の論理簡単化について議論を行った．バス反転回路の論理簡単化は，十分に許容することができる反転条件の緩和に基づいている．具体的なバス反転回路の設計とその簡単化アルゴリズムを提案し，計算機シミュレーションを行うことで，反転の効果がもともと高い場合にはバス反転の効果をほとんど落とすことなく大幅な回路簡単化ができることがわかった．また，バス反転回路の論理簡単化がバス反転回路自身の低消費化につながり，結果としてバスとバス反転回路の両方が消費する電力を削減できるケースがあることを示した．今後の課題としては，論理合成ツールを使った回路設計との比較などが挙げられる．

謝辞

広島市立大学大学院情報科学研究科情報工学専攻コンピュータデザイン研究室の亀井惇平氏には，プログラミングなどで有益なアドバイスを頂きました．ここに感謝を表します．

参考文献

[1] M. R. Stan and W. P. Burleson, “Bus-invert coding for low-power I/O, ” IEEE Trans. VLSI, Vol. 3, No. 1, pp. 49–58, 1995.

[2] S. Hong, U. Narayanan, K.-S. Chung, and T. Kim, “Bus-invert coding for low-power I/O - a decomposition approach, ” IEEE Proc. MWSCAS, pp. 750–753, 2000.

[3] M. Madhu, V. Srinivasa Murty and V. Kamakoti, “Dynamic Coding Technique For Low-Power Data Bus,” IEEE Proc. ISVLSI, pp. 252–253, 2003.

[4] T.-Y. Hsieh, M. Breuer, M. Annavaram, S. Gupta, and K.-J. Lee, “Tolerance of performance degrading faults for effective yield improvement, ”Proc. ITC, pp. 1–10, 2009.

[5] M. Breuer, “Hardware that produces bounded rather than exact results, ” Proc. DAC, pp. 871–876, 2010.

[6] D. Shin and S. Gupta, “Approximate logic synthesis for error tolerant applications, ”Proc. DATE, pp. 957–960, 2010.

[7] D. Shin and S. K. Gupta, “A new circuit simplification method for error tolerant applications, ” Proc. DATE, pp. 1–6, 2011.

[8] 亀井惇平, 松木伸伍, 岩垣剛, 市原英行, 井上智生, “エラートレラントアプリケーションのための多重縮退故障を用いた論理簡単化アルゴリズム,” 信学技報 (VLD2012-136), Vol. 112, No. 451, pp. 1-6, 2013 年 3 月.

[9] D.Shin and S.Gupta, “A Re-design Technique for Datapath Modules in Error Tolerant Applications, ” 17th Asian Test Symposium 2008 (ATS ’08), pp. 431-437, 2008.