A $4n$ LOWER BOUND ON THE COMBINATIONAL COMPLEXITY OVER $U_2 = B_2 - \{\oplus, \equiv\}$ OF CERTAIN SYMMETRIC BOOLEAN FUNCTIONS.

Uri Zwick
Department of Computer Science
Tel-Aviv University
Tel-Aviv, Israel 69978

Abstract

A simple and easy to check property of a symmetric boolean function is shown to imply a $4n-9$ lower bound on the circuit complexity of the function over $U_2 = B_2 - \{\oplus, \equiv\}$. Among the functions to which this lower bound applies are the modular functions $\mathrm{MOD}_k(n)$ for any fixed $k \geq 3$ ($\mathrm{MOD}_k(n)$ is the function which returns 1 iff $(\sum x_i) \bmod k = 0$). Finally, we obtain a $5n$ upper bound on the circuit complexity over U_2 of the function $\mathrm{MOD}_4(n)$.

1. Introduction

In 1949, Shannon [Sh] showed that the circuit complexity of almost all boolean functions is exponential. However attempts to obtain concrete lower bounds for functions in NP (see [KM],[HHS],[Sc1],[Sc2],[P],[St],[B]) yielded only linear results. The best lower bound of this kind known today over B_2, the full binary basis, is a $3n$ lower bound obtained by Blum [B].

In this note we show that a $4n-9$ lower bound may be proved if the linear functions XOR and its complement are removed from the basis. The best previous lower bound over this basis, denoted by U_2, was a $3n$ lower bound on the circuit complexity of the function $\mathrm{MOD}_2(n)$ obtained by Schnorr [Sc1]. The result presented here is in a sense a generalization of Schnorr's result. Another result which is close in spirit to the result presented here is the $2.5n$ lower bound (over B_2) obtained by Stockmeyer [St]. Some of the ideas in this work were inspired by the work of Lai and Muroga [LM].

For additional lower bounds over the bases $\{|\}$, $\{\vee,\neg\}$, $\{\vee,\wedge,\neg\}$ see [So] and [Re].

In proving the $4n-9$ lower bound we use a simple variation of the elimination method. It is very unlikely that this method will enable us to produce significantly better (for example, non linear) lower bounds. For such an improvement a majorbreak through, like the one recently obtained in the theory of monotone circuits (see [Ra],[A],[AB]), is probably needed.

2. Preliminaries

In order to keep this note concise, we do not repeat the formal definitions of boolean functions, circuits and circuit complexity, which we assume are known to the reader.

The circuits and bounds in this note are, unless otherwise stated, over U_2. In the circuits we consider we allow constant input nodes but assume that all the internal nodes are labeled by the eight non-degenerate functions $(x^a \wedge y^b)^c$ where $x^a = x \oplus a$. If A is a node in a circuit β we denote by $d_\beta(A)$ the outdegree of A, and by $\mathrm{res}_\beta(A)$ the function computed at A (the subscript β is omitted when no confusion arises). We denote by $\mathrm{res}(\beta)$ the function computed by β (i.e., the function computed at the output node of β; We consider only circuits with one output node) and by $\mathrm{C}(\beta)$ the size of β (i.e., the number of internal nodes in β; Constant input nodes are not counted), and by $d_1(\beta)$ the number of variables with outdegree one (we assume that every variable appears in only one input node). The circuit complexity of a function f (over U_2) (i.e., the size of the minimal circuit computing f) is denoted by $\mathrm{C}_{U_2}(f) = \mathrm{C}(f)$.

If a gate A in a circuit β is fed by the node B and $\mathrm{res}(B)$ is constant then we can obtain a smaller circuit which computes f in the following way: If $\mathrm{res}(A)$ is also constant, then we simply remove the incoming edges of A from the circuit and turn A into a constant input node. Otherwise $\mathrm{res}(A) = \mathrm{res}(C)^a$ where C is the second node in the circuit feeding A and $a \in \{0,1\}$. In this case we remove the node A and all its incoming and outgoing edges from the circuit. The gates which were fed by A will be now fed directly by C. If $a = 1$ a complement must be incorporated into these gates.

In the next section we encounter many situations in which we are given a circuit β that some of its gates are fed by constants. In each such situation we explicitly identify a subset of these gates and remove them one by one, as explained in the previous paragraph.

A circuit β is said to be *simplified* if the output of every node in it is not constant (otherwise the circuit can be simplified).

If $g(x,y) \in U_2$ then there exists a constant $c \in \{0,1\}$ such that $g(c,y)$ is a constant. We say that the constant c *blocks* the function g. Notice that this property does not hold for B_2 and this is why proving lower bounds over this base is a harder problem.

3. The lower bound

We begin by defining the set of functions for which our lower bound applies:

Definition 3.1. *The sets* $\mathrm{S}(n), \mathrm{M}_k(n), \mathrm{N}_\ell(n), \mathrm{MN}_{k,\ell}(n)$.

The sets $\mathrm{S}(n), \mathrm{M}_k(n), \mathrm{N}_\ell(n), \mathrm{MN}_{k,\ell}(n)$ are defined in the following way:

> $f \in \mathrm{S}(n)$ iff $f(x_1,\ldots,x_k)$ depends only on $\sum_{i=1}^{n} x_i$. Functions belonging to $\mathrm{S}(n)$ are called *symmetric functions*.

$f \in \mathrm{M}_k(n)$ iff $f \in \mathrm{S}(n)$ and every restriction of f to a subset of k variables is not constant.

$f \in \mathrm{N}_\ell(n)$ iff $f \in \mathrm{S}(n)$ and every restriction of f to a subset $\{x_1, \ldots, x_\ell\}$ of f's variables is not $x_1 \oplus \ldots \oplus x_\ell$ or its complement.

And finally we define: $\quad \mathrm{MN}_{k,\ell}(n) = \mathrm{M}_k(n) \cap \mathrm{N}_\ell(n) \quad .$

The structure of the sets $\quad \mathrm{M}_k(n), \mathrm{N}_\ell(n), \mathrm{MN}_{k,\ell}(n) \quad$ for small k and ℓ is very simple:

$$\mathrm{M}_1(n) = \{\, x_1 \oplus \ldots \oplus x_n \oplus c \,:\, c = 0, 1 \,\} \qquad , \qquad \mathrm{N}_1(n) = \{0, 1\}$$

$$\mathrm{MN}_{k,1}(n) = \emptyset \quad \text{for} \quad n \geq k \qquad , \qquad \mathrm{MN}_{1,\ell}(n) = \emptyset \quad \text{for} \quad n \geq \ell$$

$$\mathrm{MN}_{2,2} = \left\{ \left\lfloor \frac{(\sum x_i + c) \bmod 4}{2} \right\rfloor \,:\, c = 0, 1, 2, 3 \right\}$$

As a further example we note that $\mathrm{MOD}_k(n) \in \mathrm{MN}_{k-1,3}(n)$ for $k > 2$.

It is easy to check that if $f \in \mathrm{MN}_{k,\ell}(n)$ for some $n \geq k, \ell$ and x, y are two variables of f, then f has a restriction to $\{x, y\}$ of the form $(x^a \wedge y^b)^c$. It is also obvious that if $f \in \mathrm{MN}_{k,\ell}(n)$ for some $n > k, \ell$ then every restriction of f obtained by fixing the value of one variable belongs to $\mathrm{MN}_{k,\ell}(n-1)$.

We can now prove that if $f \in \mathrm{MN}_{k,\ell}(n)$ and $n \geq k+1, l$, then $\mathrm{C}_{U_2}(f) \geq 4(n-m) - 1$ where $m = \max\{k+1, \ell\}$.

Lemma 3.2. *Let β be a circuit which computes a function $f \in \mathrm{MN}_{k,\ell}(n)$ for $n > k+1, \ell$. There there exists a circuit δ which computes a function $f' \in \mathrm{MN}_{k,\ell}(n-1)$ and which satisfies $[\mathrm{C}(\delta) - d_1(\delta)] \leq [\mathrm{C}(\beta) - d_1(\beta)] - 4$.*

Proof: Let β be a circuit which computes a function $f \in \mathrm{MN}_{k,\ell}(n)$ where $n > k+1, \ell$. If β is not simplified then simplify it and denote the simplified circuit obtained by γ. It is easy to check that $[\mathrm{C}(\gamma) - d_1(\gamma)] \leq [\mathrm{C}(\beta) - d_1(\beta)]$. In γ there exists a gate B which is fed by two input variables. Let x, y be the variables feeding B. The outdegrees of x and y must

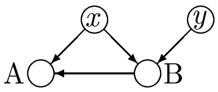

Figure 3.1:

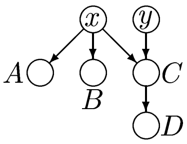

Figure 3.2:

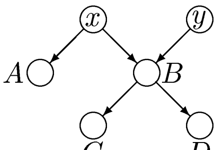

Figure 3.3:

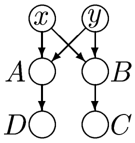

Figure 3.4:

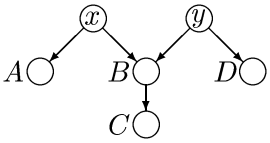

Figure 3.5:

be at least 2 (since otherwise we an assign a value to one of them and make the output independent of the other).

We say that the circuit γ is *degenerate* if it contains the situation shown in Fig. 3.1 (or the symmetrical situation with the rolls of x and y switched). It is easy to transform γ into a non degenerate circuit γ' which also computes f and which satisfies $\mathrm{C}(\gamma') \leq \mathrm{C}(\gamma), d_1(\gamma') = d_1(\gamma)$. This can be accomplished by either deleting A from the circuit, if its output does not depend on both x and y, or by replacing the edge $B \to A$ by an edge $y \to A$, and by adjusting the gate A if necessary, otherwise. Notice that in the latter case $\mathrm{res}(A) = (x^a \wedge y^b)^c$ since in order to compute $(x \oplus y)^d$ at least 3 U_2-gates are needed.

We may therefore assume, without loss of generality, that the circuit γ is non-degenerate. We consider the following three cases:

Case 1. *$d_\gamma(x) \geq 3$ or $d_\gamma(y) \geq 3$.*

Assume without loss of generality that $d_\gamma(x) \geq 3$. Let A, B, C be three distinct gates fed by x, and suppose that C is fed also by y, as shown in Fig. 3.2. Let D be a gate fed by C. Since γ is non-degenerate $D \neq A, B$. We assign to x the constant c which blocks C and delete the gates A, B, C, D from β as explained in the previous section. Denote by δ the new circuit obtained. The circuit δ computes the function $f_{x:=c} \in \mathrm{MN}_{k,\ell}(n-1)$. Notice that if $d_\gamma(z) = 1$ then z does not feed A, B or D in γ (since otherwise choosing the right values for x and y would make the output of γ independent of z and this is a contradiction since $n > k+1$). Therefore $d_\delta(z)$ remains 1 and thus we have $\mathrm{C}(\delta) = \mathrm{C}(\gamma) - 4$ and $d_1(\delta) \geq d_1(\gamma)$ as required.

If case 1 does not hold then $d_\gamma(x) = d_\gamma(y) = 2$. Denote by A, B the gates fed by x and suppose that B is fed also by y.

Case 2. $d_\gamma(B) \geq 2$

Denote by C, D the two distinct gates fed by B (see Fig. 3.3). Since γ is non-degenerate $C, D \neq A$. We assign to x the constant c which blocks B. As in the previous case, we delete the gates A, B, C, D and we are left with a circuit δ which satisfies $\mathrm{C}(\delta) = \mathrm{C}(\gamma) - 4$ and $d_1(\delta) \geq d_1(\gamma)$, as required.

The last case we have to consider is:

Case 3. $d_\gamma(B) = 1$

We break this case into two sub-cases:

Case 3.1. *There exists an edge $y \to A$ in γ.*

If $d_\gamma(A) \geq 2$ then after switching the roles of x and y we are back in case 2. We therefore assume that $d_\gamma(A) = d_\gamma(B) = 1$. Denote by C the only gate fed by B, and by D the only gate fed by A. We claim that $C \neq D$, for otherwise the output of γ depends on x and y only through the gate $C = D$. If $\mathrm{res}(C)$ is constant or of one of the forms x^a,y^b, $(x^a \wedge y^b)^c$ then one of x or y may block the other, and this is clearly a contradiction. The only possibility left is that $\mathrm{res}(C) = (x \oplus y)^d$ but this also leads to a contradiction for in this case f cannot have a restriction to $\{x, y\}$ of the form $(x^a \wedge y^b)^c$.

The situation is therefore as shown in Fig. 3.4. We assign to x the constant which blocks B, and delete the gates A, B, C from the circuit. Denote the resulting circuit by δ. Notice that $d_\delta(y) = 1$ since y feeds in δ

only the gate D. Once again if $d_\gamma(z) = 1$ then also $d_\delta(z) = 1$ and therefore $\mathrm{C}(\delta) \leq \mathrm{C}(\gamma) - 3$ and $d_1(\delta) \geq d_1(\gamma) + 1$ as required.

Case 3.2. *There is no edge $y \to A$ in γ.*

Denote by C the unique gate fed by B. Denote by D the second gate fed by y. By the non-degeneracy of γ we obtain that $C \neq A, D$. We assume in this case that $D \neq A$, thus the situation is as shown in Fig. 3.5. As usual, we assign to x the constant which blocks B, and delete the gates A, B, C from the circuit. We can now repeat the arguments of the previous sub-case.

In each one of the above cases we obtained a circuit which satisfies the conditions required and thus the proof is complete.

□

Theorem 3.3. *If $f \in \mathrm{MN}_{k,\ell}(n)$ then $\mathrm{C}_{U_2}(f) \geq 4(n-m)+3$ where $m = \max\{k+1, \ell\}$.*

Proof: We prove by induction on n that if β is a circuit which computes $f \in \mathrm{MN}_{k,\ell}(n)$ then $[\mathrm{C}(\beta) - d_1(\beta)] \geq 4(n-m)-1$. The basis of the induction for $n = m$ follows from the fact that $\mathrm{C}(\beta) \geq m-1$, $d_1(\beta) \leq m$ and therefore $[\mathrm{C}(\beta) - d_1(\beta)] \geq -1$. The induction step follows immediately from lemma 3.2.

□

4. An upper bound.

In this section we present U_2-circuits of size $5n-7$ which compute the functions $\mathrm{MOD}_4(n)$ (for $n \geq 3$). This shows that $4n - O(1) \leq \mathrm{C}_{U_2}(\mathrm{MOD}_4) \leq 5n - O(1)$.

The basic building block in our circuits is the binary full adder (FA) shown in Fig. 4.1(a). In Fig. 4.1(b) and 4.1(c) it is shown how a FA can be implemented using 5 B_2-gates or 7 U_2-gates. It is easy to check that both implementations are optimal.

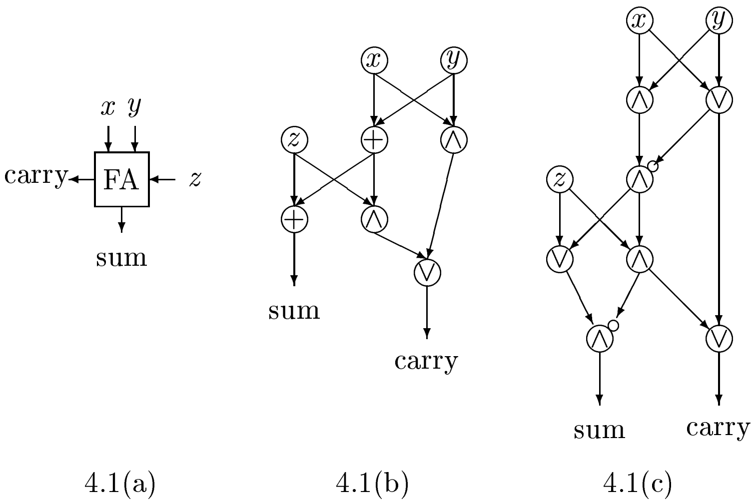

Figure 4.1: Implementations of Full Adders.

We now present the circuits for $\mathrm{MOD}_4(n)$. Assume for simplicity that $n \geq 3$ is odd. Construct a trenary tree of $FA's$ in which every FA is fed by 3 inputs which are either input variables or sum outputs of previous FA's (This is possible since n is odd). The exact structure of the tree is immaterial. The number of FA's in the tree is $\frac{n-1}{2}$. Now compute the XOR of all the carries produced by the $\frac{n-1}{2}$ FA's. This XOR can be computed using $3 \cdot \frac{n-1}{2} - 3$ U_2-gates. The value of the function $\mathrm{MOD}_4(n)$ is obtained by performing a NOR operation between this XOR and the sum output of the root FA. The total size of the circuit is $7 \cdot \frac{n-1}{2} + \left(3 \cdot \frac{n-1}{2} - 3\right) + 1 = 5n - 7$.

The same construction yields B_2-circuits of size $3n$ for the functions $\mathrm{MOD}_4(n)$. Stockmeyer [St] constructed more efficient circuits for $\mathrm{MOD}_4(n)$ over B_2 and showed that $\mathrm{C}_{B_2}(\mathrm{MOD}_4(n)) = 2.5n - O(1)$.

Over U_2 there is still an unresolved gap between the $4n$ lower bound and the $5n$ upper bound presented for MOD_4. We believe that the upper bound is closer to the truth and that the function $\mathrm{MOD}_4(n)$ is the easiest function among the functions to which the lower bound presented in this note applies.

References

[A] Andreev A.E., On a method for obtaining lower bounds for the complexity of individual monotone functions. *Dokl. Ak. Nauk. SSSR 282 (1985) p. 1033-1037* (in Russian). English translation in *Sov. Math. Dokl. 31 (1985) pp. 530-534.*

[AB] Alon N., Boppana R.B., The monotone circuit complexity of boolean functions. *Combinatorica vol. 7 (1987) no. 1 pp. 1-22.*

[B] Blum N., A Boolean function requiring $3n$ network size. *Theoretical Computer Science 28 (1984) pp. 337-345.*

[HHS] Harper L.H., Hsieh W.N., Savage J.E., A class of boolean functions with linear combinational complexity. *Theoretical Computer Science 1 (1975) pp. 161-183.*

[KM] Kloss B.M., Malyshev V.A., Bounds on complexity of some classes of functions. *Vestnik MGU, Ser. Mat. Mech. 4 (1965) pp. 44-51* (in Russian).

[LM] Lai H.C., Muroga S., Logic networks with a minimum number of NOR (NAND) gates for parity functions of n variables. *IEEE Transactions of Computers vol. C-36 (1987) no. 2 pp. 156-167.*

[P] Paul W.J., A $2.5n$-lower bound on the combinatorial complexity of boolean functions. *SIAM J. Comp. Vol. 6 (1977), No. 3, pp. 427-443.*

[Ra] Razborov A.A., Lower bounds for the monotone complexity of some boolean functions. *Dokl. Ak. Nauk. SSSR 281 (1985) pp. 791-801.* (in Russian). English translation in *Sov. Math. Dokl. 31 (1985) p. 354-357.*

[Re] Red'kin N.P., Proof of minimality of circuits consisting of functional elements. *Problemy Kibernetiki 23 (1970) pp. 83-101* (in Russian).

[Sc1] Schnorr C.P., Zwei lineare untere Schranken für die Komplexität Boolescher Funcktionen. *Computing 13 (1974) pp. 155-171.*

[Sc2] Schnorr C.P., The combinational complexity of equivalence. *Theoretical Computer Science 1 (1976) pp. 289-295.*

[Sh] Shannon C.E., The synthesis of two-terminal switching circuits. *Bell System Technical Journal 28 (1949) pp. 59-98.*

[So] Soprunenko E.P. Minimal realization of functions by circuits using functional elements. *Problemy Kibernetiki vol. 15 (1965) pp. 117-134.*

[St] Stockmeyer L., On the combinational complexity of certain symmetric boolean functions. *Mathematical Systems Theory vol. 10 (1977) pp. 323-336.*

[W] Wegener I., The complexity of boolean functions. *Wiley-Teubner Series in Computer Science (1987).*