Karchmer-Wigderson Games for Hazard-Free Computation

Christian Ikenmeyer ✉
University of Warwick, Coventry, UK

Balagopal Komarath ✉
IIT Gandhinagar, India

Nitin Saurabh ✉
IIT Hyderabad, India

Abstract

We present a Karchmer-Wigderson game to study the complexity of hazard-free formulas. This new game is both a generalization of the monotone Karchmer-Wigderson game and an analog of the classical Boolean Karchmer-Wigderson game. Therefore, it acts as a bridge between the existing monotone and general games.

Using this game, we prove hazard-free formula size and depth lower bounds that are provably stronger than those possible by the standard technique of transferring results from monotone complexity in a black-box fashion. For the multiplexer function we give (1) a hazard-free formula of optimal size and (2) an improved low-depth hazard-free formula of almost optimal size and (3) a hazard-free formula with alternation depth 2 that has optimal depth. We then use our optimal constructions to obtain an improved universal worst-case hazard-free formula size upper bound. We see our results as a step towards establishing hazard-free computation as an independent missing link between Boolean complexity and monotone complexity.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Communication complexity; Theory of computation → Concurrency; Hardware → Combinational circuits

Keywords and phrases Hazard-free computation, monotone computation, Karchmer-Wigderson games, communication complexity, lower bounds

Digital Object Identifier 10.4230/LIPIcs.ITCS.2023.74

Related Version *Full Version*: `https://arxiv.org/abs/2107.05128`

Funding *Nitin Saurabh*: This project has received funding from the the European Union's Horizon 2020 research and innovation programme under grant agreement No 802020-ERC-HARMONIC.

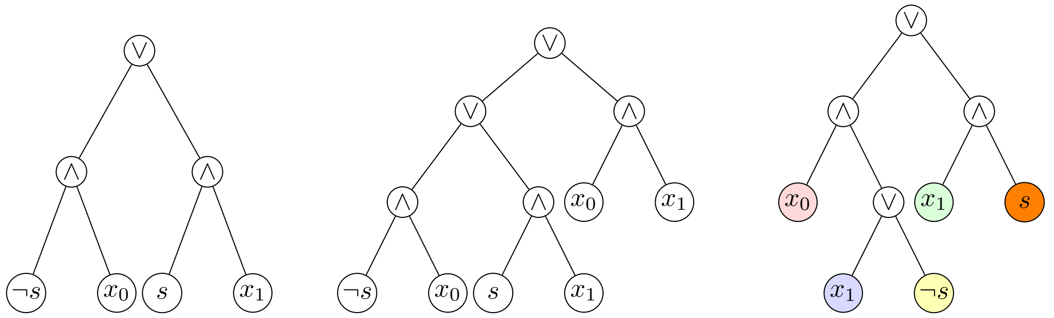
Acknowledgements We thank Igor Sergeev for his support with the literature. Nitin Saurabh would like to thank Yuval Filmus for helpful discussions. We would also like to thank anonymous reviewers for useful suggestions that greatly improved the presentation.

1 Introduction

In this paper we apply for the first time methods from communication complexity to the study of hazard-free complexity, which we see as a step towards bridging the gap between Boolean complexity and monotone complexity.

The study of the three-valued strong logic of indeterminacy dates back to Kleene ([31, p. 153], [32, §64]). It found numerous applications, for example in logic (see e.g. [33, 22]), in cybersecurity for information flow tracking at the gate level (see e.g. [66, 25, 4]), the design of real-world circuits that communicate between unsynchronized clock domains (see e.g. [16, 17, 63, 9]), and in the study of hazards in Boolean circuits (see e.g. [20, 10, 67, 13, 45, 46, 47, 48, 6, 5]). The languages in these areas is different, but the underlying three-valued

14th Innovations in Theoretical Computer Science Conference (ITCS 2023).
Editor: Yael Tauman Kalai; Article No. 74; pp. 74:1–74:25
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(a) A size-optimal formula, but with a hazard at $(s, x_0, x_1) = (\mathsf{u}, 1, 1)$.

(b) The common hazard-free formula. There is visible symmetry between x_0 and x_1.

(c) A size-optimal hazard-free formula. The symmetry is broken.

Figure 1 Different De Morgan formulas for MUX_1.

logic is the same and many questions and results can be readily transferred between areas. We will use the language of hazards in circuits in this paper. The use of three-valued logic to study hazards in Boolean circuits dates all the way back to Goto [20], who used 0 and 1 to denote the Boolean values and used the symbol $\frac{1}{2}$ to denote the third value, which stands for any undefined, oscillating, unstable, or otherwise somehow flawed state. In this paper we use the symbol $\mathsf{u} := \frac{1}{2}$ to denote this third state. Goto modeled the Boolean operations $\wedge$ (and) and $\vee$ (or) as min and max, respectively, and the $\neg$ (not) operation as $1-x$, which defines the behaviour of the three types of gates on inputs from $\{0, \mathsf{u}, 1\}$. Hence a Boolean circuit C on n inputs[1] computes a function $\{0, \mathsf{u}, 1\}^n \to \{0, \mathsf{u}, 1\}$ by induction over the circuit structure. The design of the gate behaviour as min, max, and $1 - x$ is the result of a more general construction principle that is called the *hazard-free extension*[2] $\widetilde{f} : \{0, \mathsf{u}, 1\}^n \to \{0, \mathsf{u}, 1\}$ of a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$. It is defined as follows.

A binary string $a \in \{0, 1\}^n$ is called a *resolution* of a ternary string $\alpha \in \{0, \mathsf{u}, 1\}^n$ if for all $1 \le i \le n$ with $\alpha_i \neq \mathsf{u}$ we have $\alpha_i = a_i$, i.e., all entries u are replaced by 0s and 1s. Note that the set of all resolutions a of α forms a subcube of $\{0, 1\}^n$. For a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ and for an input $\alpha \in \{0, \mathsf{u}, 1\}^n$ we define the evaluation of the function $\widetilde{f} : \{0, \mathsf{u}, 1\}^n \to \{0, \mathsf{u}, 1\}$ at α via

$$\widetilde{f}(\alpha) := \begin{cases} 1 & \text{if for all resolutions } a \text{ of } \alpha \text{ we have } f(a) = 1 \\ 0 & \text{if for all resolutions } a \text{ of } \alpha \text{ we have } f(a) = 0 \\ \mathsf{u} & \text{otherwise.} \end{cases} \tag{1}$$

A Boolean circuit C that computes a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ is called *hazard-free* if for all $\alpha \in \{0, \mathsf{u}, 1\}^n$ we have $C(\alpha) = \widetilde{f}(\alpha)$. An α where these two functions differ is called a *hazard*. For example, consider the circuit in Part (a) of Figure 1 that computes the multiplexer function $C(s, x_0, x_1) = \mathsf{MUX}(s, x_0, x_1) = x_s$ for all $(s, x_0, x_1) \in \{0, 1\}^3$. We observe that C has a hazard at $(\mathsf{u}, 1, 1)$, because $C(\mathsf{u}, 1, 1) = \mathsf{u} \vee \mathsf{u} = \mathsf{u}$, whereas $C(0, 1, 1) = C(1, 1, 1) = 1$. The circuit can be made hazard-free at the expense of using more gates, see Part (b) of Figure 1 (this construction can be found for example in [16, Fig. 6a] and [27, Fig. 1b]).

---

[1] All circuits in our paper have a single output.

[2] The function $\widetilde{f}$ is called the *hazard-free extension* of f (see [27]), or alternatively the *ternary extension* (see [44]) or the *metastable closure* (see [16]).

Designing small hazard-free circuits for computing Boolean functions is a fundamental goal in electronic circuit design. Huffman [26] proved that all Boolean functions can be implemented by hazard-free circuits and he already noted the large growth of the number of gates in his examples. Eichelberger proved the first lower bound on hazard-free complexity in the restricted model of DNF formulas, which is given by the number of prime implicants of the function that is computed. The very recent paper [27] formally defines the notion of hazard-free complexity and shows that for monotone functions the hazard-free complexity and the monotone complexity coincide. Fortunately, good lower bounds are known on the monotone complexity of monotone Boolean functions (see [55, 56, 2, 1, 57, 64, 21, 29, 54, 53, 23, 19, 49]). A direct consequence of [27] is that the exponential gap between Boolean circuit complexity and monotone circuit complexity transfers directly into an exponential gap between Boolean circuit complexity and the hazard-free circuit complexity. [28] proves that every Boolean circuit that computes a monotone function and that is optimal with respect to hazard-free complexity must automatically be a monotone circuit. Hence the study of hazard-free complexity does not yield any new insights into monotone functions, but it is a natural generalization of monotone complexity to the domain of *all* Boolean functions. This suggests that the study of hazard-free complexity, in particular of non-monotone functions, should be of independent interest (apart from its applicability in practice). As a first step in this direction, [27] prove lower bounds for non-monotone functions by using monotone circuit lower bounds for the *hazard-derivative* of the function, because the monotone complexity of the hazard-derivative of f is a lower bound on the hazard-free complexity of f. All existing lower bounds known for hazard-free computation [27, 28] are derived from this wealth of known monotone complexity lower bounds.

However, the hazard-derivative method cannot always prove optimal lower bounds, because some functions with high hazard-free complexity have hazard-derivatives of only low monotone complexity (compare Proposition 1 with Theorem 23). We call this problem the *monotone barrier*. In this paper we take a radically different approach than all previous papers and translate notions from communication complexity to the hazard-free setting. The result is a new type of the Karchmer-Wigderson game that exactly describes the hazard-free De Morgan formula size and depth. Our new game is at the same time a hazard-free analog of the classical Boolean Karchmer-Wigderson game (Remark 6) and a generalization of the monotone Karchmer-Wigderson game to the set of all Boolean functions: it coincides with the monotone Karchmer-Wigderson game when played on monotone functions (Theorem 11). In other words, the difference between the monotone Karchmer-Wigderson game and the Boolean Karchmer-Wigderson game is precisely the presence of hazards in the Boolean game. We use this new definition to precisely determine the hazard-free formula size (Theorems 19 and 23) and the depth of hazard-free formulas of alternation depth[3] 2 (Theorem 31) of the multiplexer function $\mathsf{MUX}_n : \{0,1\}^{n+2^n} \to \{0,1\}$, which is a (non-monotone) Boolean function on $n + 2^n$ input bits, defined via

$$\mathsf{MUX}_n(s_1, \ldots, s_n, x_0, x_1, \ldots, x_{2^n-1}) = x_{\mathrm{bin}(s_1,\ldots,s_n)},$$

where $\mathrm{bin}(s_1, \ldots, s_n)$ is the natural number represented by the binary number $s_1 s_2 \cdots s_n$.

---

[3] Alternation depth is one plus the maximum number of changes in the type of the gate in root-to-leaf paths.

Our result breaks the monotone barrier, i.e., the hazard-derivatives of the multiplexer have lower complexity than the bound we prove. To obtain matching upper and lower bounds on complexity we use the Karchmer-Wigderson game interpretation to give two new efficient hazard-free implementations of the multiplexer function: One is optimal for the formula size and one is optimal for the depth of hazard-free formulas of alternation depth 2.

In contrast to monotone complexity, which is mainly a theoretical concept, hazard-free complexity has applications in practice, not only in cybersecurity ([66], [25], [4]), but also for designing real-world circuits, for example when a distributed system of agents with unsynchronized clock domains performs a parallel computation, see [17, 16, 63, 35, 7, 8, 9]. The hazard-free circuit depth (which is equal to the hazard-free formula depth) is a main parameter in this research area, directly correlated to a circuit's execution time.

An interesting incremental approach towards proving super-polynomial formula size lower bounds for explicit functions, is to make progress by proving good lower bounds for formulas with more and more NOT gates [15, 62, 3]. In Section 8, we show that instead of considering all implicants and implicates, we can choose any subset of implicants and implicates to obtain upper and lower bounds on *limited* hazard-free formulas, formulas that are guaranteed to be hazard-free on some inputs but not others. That is, we can parameterize our game by the number of undefined inputs so that it interpolates between the hazard-free game and the general Boolean game. This gives us a natural way to make progress towards proving super-polynomial Boolean formula size lower bounds by proving super-polynomial lower bounds for more and more limited hazard-free formulas, until we prove a lower bound on formulas that may have hazards on any input. Limited hazard-free formulas are also of interest in practice, for example when it is known that the unstable bit can only appear in the position where two adjacent Gray code numbers differ [17, 35, 7, 8, 9]. We are not aware of any applications of limited negation circuits for designing real-world circuits.

The multiplexer function is also significant from the perspective of proving super-polynomial formula size lower bounds. Informally, it suffices to prove a lower bound for a composition of the multiplexer function with itself. For a formal statement, see [12, 43].

1.1 Exact Bounds

In Section 5 we determine the *exact* hazard-free formula complexity of the multiplexer function. We achieve this by using a combination of an improvement in the upper bound (Huffman's [26] construction gives only $\mathsf{size}^{\mathsf{u}}(\mathsf{MUX}_n) \leq 4^n + 2n3^{n-1}$) and an analysis of the hazard-free Karchmer-Wigderson game for the lower bound: $\mathsf{size}^{\mathsf{u}}(\mathsf{MUX}_n) = 2 \cdot 3^n - 1$.

It is known that there are De Morgan formulas (with hazards) of size $2^{n+1}(1 + o(1))$ computing MUX_n [37], i.e., $\mathsf{size}(\mathsf{MUX}_n) \leq 2^{n+1}\left(1 + \frac{1}{2n} + O(\frac{1}{n \log n})\right)$. Our upper bound construction is a recursive application of the improved implementation of MUX_1 in Figure 1(c). To prove the lower bound we reduce the Karchmer-Wigderson game for MUX_n from a communication game for the *subcube intersection problem.* Its communication matrix is highly structured, so that its rank can be determined and be used to find the lower bound. The subcube intersection problem is the hazard-free generalization of the classical equality problem from communication complexity and could be of independent interest, especially for proving other hazard-free formula lower bounds.

Since all derivatives of MUX_n have monotone formulas of size at most $(n+1)2^n$ (Proposition 1), the separation that we achieve breaks the monotone barrier. Therefore, our lower bound is the first to separate the Boolean complexity and the hazard-free complexity of a function while breaking the monotone barrier[4]. We consider this an important step forward towards establishing hazard-free computation as a new theoretical device that can serve as a true generalization of monotone circuit complexity.

Considering the depth (which is the same for circuits and formulas), we immediately obtain $\mathsf{depth}^{\mathsf{u}}(\mathsf{MUX}_n) \geq \log_2(3)n \geq 1.58n$. This lower bound separates the hazard-free circuit depth complexity and Boolean circuit depth complexity of MUX_n, because $\mathsf{depth}(\mathsf{MUX}_n) \leq n+3$ [65, 38]. (In fact, $\mathsf{depth}(\mathsf{MUX}_n) = n+2$ for all $n \geq 20$ [38].) Analogously to formula size, since all derivatives of MUX_n have monotone circuits of depth at most $n + \log_2(n) + 1$ (Proposition 1), our separation breaks the monotone barrier.

In Section 7 we focus on the depth of hazard-free formulas of alternation depth 2 for MUX_n. These formulas are interesting in practice because certain programmable logic arrays produce implementations that have alternation depth 2. We prove the exact complexity of the multiplexer function in this restricted model to be $2n+2$. For the proof we exploit an old result by Huffman: the fact that in this restricted model Alice must communicate her prime implicant to Bob before Bob starts communicating. Therefore small-depth formulas can exist if and only if there are short prefix codes that allow Alice to communicate her prime implicant efficiently to Bob. Then using Kraft's inequality from information theory we show that there are prefix codes that achieve a depth upper bound of $2n+2$, but they cannot achieve $2n+1$. One key idea is a distinction of cases between prime implicants of logarithmic size and prime implicants of super-logarithmic size.

For general hazard-free formula depth the upper bound of $2n+2$ is not optimal for MUX_n, because we show in Theorem 25 that the depth is at most $2n+1$. Note that this is significantly lower than the depth $3n$ achieved by the formula of optimal size in Theorem 19, and strictly lower than the depth that can be achieved by any formula of alternation depth 2. This gives a size-depth trade-off: the size of this formula is only a factor of $\frac{9}{8}$ more than the optimal size. This construction is done recursively using the hazard-free Karchmer-Wigderson game. It is crucial in this recursion that the induction hypothesis is *not* the monochromatic partitioning of the communication matrix of MUX_{n-1}, but of an enlarged matrix that can be partitioned monochromatically using the same depth.

All upper bounds and lower bounds are proved using the framework of hazard-free Karchmer-Wigderson games. The lower bound proofs rely heavily on this framework. The game also played a crucial role in deriving the upper bounds given in Theorems 25 and 31. The upper bound in Theorem 19 can also be be proved without using the game (see Remark 18).

1.2 Universal Upper Bounds

One of the most fundamental and oldest questions in electronic circuit design is finding an upper bound on the size of circuits or formulas that holds for all Boolean functions [60]. For Boolean circuits and formulas, this question has been very satisfactorily answered. It is known that any n-bit Boolean function has circuits of size $(1+o(1))2^n/n$ [39, 36] and almost

---

[4] Note that breaking the monotone barrier can also be achieved using Khrapchenko's method for the parity function [30], which was interpreted as a Karchmer-Wigderson game in [29], but for the parity function the hazard-free complexity and the Boolean complexity coincide (every implementation of parity is automatically hazard-free): Parity requires $\Theta(n^2)$ formula size, but the derivatives of parity are all equal to the OR function, which requires $\Theta(n)$ formula size. For the parity function the Boolean Karchmer-Wigderson game coincides with our hazard-free Karchmer-Wigderson game, so we obtain the same bounds.

all Boolean functions require circuits of size $(1+o(1))2^n/n$ [41]. For Boolean formulas, the lower bound is $(1-o(1))2^n/\log(n)$ [59], almost matched by the upper bound $(1+o(1))\frac{2^n}{\log n}$ [40, 36].

For hazard-free circuits, the situation is very similar to that of Boolean circuits: any n-bit Boolean function has a hazard-free circuit of size $O(2^n/n)$ (see, e.g., [28, Section 7]), thus matching Lupanov's upper bound [39] up to constants. Since hazard-free circuits are also Boolean circuits, the lower bound of $(1+o(1))2^n/n$ for almost all functions continues to hold for hazard-free circuits.

For hazard-free formulas, this question is still open. Huffman [26] gives hazard-free implementations for any function by representing it as a DNF where the set of terms is the set of all prime implicants of the function. Since a function on n variables may have as many as $\Omega(3^n/\sqrt{n})$ prime implicants [11] and each prime implicant may contain as many as n literals, this translates into a worst-case bound of $O(\sqrt{n}\cdot 3^n)$ on the hazard-free formula complexity.

We make progress on this question by studying the multiplexer function. In electronic circuit design, the multiplexer is often used as a programmable logic device. Indeed, given any Boolean function $f:\{0,1\}^n \mapsto \{0,1\}$, we can implement it as: $f(x_1,\dots,x_n) = \mathsf{MUX}_n(x_1,\dots,x_n, f(0,0,\dots,0),\dots,f(1,1,\dots,1))$. This implementation of f is hazard-free if the implementation of MUX_n is hazard-free. Therefore, any hazard-free formula upper bound for MUX_n gives an upper bound for the hazard-free formula complexity of *all* n-bit Boolean functions. Theorem 19 gives such an improved upper bound of $2\cdot 3^n - 1$ for the multiplexer function and hence our construction gives a new best worst-case hazard-free formula size implementation of size $2\cdot 3^n-1$, which was $O(\sqrt{n}\cdot 3^n)$ before.

Observe that in the world of Boolean circuits, Boolean formulas, and hazard-free circuits, the multiplexer upper bound is only a polynomial (in n) multiplicative factor away from the optimal bound. We show in Theorem 23 that our new bound is optimal for the multiplexer function. This means that we cannot improve the universal upper bound further by directly using the multiplexer function. However, the best known lower bound for hazard-free formulas for n-bit functions is still the $2^n/\log(n)$ given by a counting argument. This creates an interesting situation that is different from the other three settings described in this section.

- If there are n-bit functions such that the hazard-free formula size is asymptotically more than $2^n/\log(n)$, then a tight lower bound can be proved by only using some argument that exploits the *semantic* property of hazard-freeness, such as the hazard-free Karchmer-Wigderson game we introduce in this paper. This is in contrast to the other settings where tight lower bounds can be obtained using a counting argument that only exploits the structure (or syntax) of the model.
- Otherwise, all n-bit functions have hazard-free formulas that are smaller than the optimal hazard-free formula for the multiplexer function by a multiplicative factor that is exponential in n. This is also in stark contrast to the situation in the other three settings.

2 Preliminaries

Formulas

A *Boolean formula* is a Boolean circuit whose graph is a tree. That is, it is a formula over the De Morgan basis $\{\vee,\wedge,\neg\}$. The $\vee$ and $\wedge$ gates have fan-in two and $\neg$ gates have fan-in one. Using De Morgan's laws (which also work over the three-valued logic) the negations can be moved to the leaves: all internal nodes are labeled with $\vee$ or $\wedge$ and all leaves are labeled with literals x_i or $\neg x_i$. This is called a *De Morgan formula.* The *size* of a De Morgan formula F, denoted $\mathsf{size}(F)$, is defined to be the number of leaves in it[5]. The *depth* of a formula F, denoted $\mathsf{depth}(F)$, is defined to be the length of the longest root-to-leaf path in F. For a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, we denote the minimal size of a De Morgan formula computing f by $\mathsf{size}(f)$ and the minimal depth of a formula computing f by $\mathsf{depth}(f)$. Similarly, in the hazard-free setting, let $\mathsf{size}^{\mathsf{u}}(f)$ and $\mathsf{depth}^{\mathsf{u}}(f)$ denote the minimal size and minimal depth of a hazard-free De Morgan formula computing f, respectively. For a monotone function f let $\mathsf{size}^{+}(f)$ and $\mathsf{depth}^{+}(f)$ denote the minimal size and minimal depth of a monotone formula computing f, respectively. The *alternation depth* of a formula is one plus maximum number of changes to the type of the gate in the sequence of gates in a root-to-leaf path. For example, the alternation depth of the formula in Figure 1(b) is 2 and that of the formula in Figure 1(c) is 3. We denote the minimal size and depth of hazard-free formulas of alternation depth d using $\mathsf{size}^{\mathsf{u}}_d(f)$ and $\mathsf{depth}^{\mathsf{u}}_d(f)$, respectively.

Implicants and Implicates

For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ the preimage of a value $c \in \{0,1\}$ is denoted by $f^{-1}(c)$. For the hazard-free extension $\widetilde{f} : \{0,\mathsf{u},1\}^n \to \{0,\mathsf{u},1\}$ the preimage of $\gamma \in \{0,\mathsf{u},1\}$ is denoted by $\widetilde{f}^{-1}(\gamma)$. Elements $\alpha \in \widetilde{f}^{-1}(1)$ are called *implicants* of f. A *prime implicant* is an implicant in which no value from $\{0,1\}$ can be replaced by a u such that it is still an implicant, i.e., a prime implicant is an implicant that is minimal with respect to the *instability partial order*, the partial order on $\{0,1,\mathsf{u}\}$ defined by $\mathsf{u} \leq 0$ and $\mathsf{u} \leq 1$. Elements $\alpha \in \widetilde{f}^{-1}(0)$ are called *implicates* of f. A *prime implicate* is an implicate in which no value from $\{0,1\}$ can be replaced by a u such that it is still an implicate, i.e., a prime implicate is an implicate that is minimal with respect to the instability partial order. We occasionally identify an implicant α with the Boolean function that is 1 exactly on the hypercube of resolutions of α, and an implicate β with the Boolean function that is 0 exactly on the hypercube of resolutions of β.

Communication

We assume familiarity with the basic definitions of communication complexity (see, e.g., [34, 52]). Let $K\colon A \times B \to 2^O$ be a function that maps tuples to nonempty subsets of a set O. For the purposes of this paper we will only be interested in deterministic communication complexity where Alice gets $\alpha \in A$, Bob gets $\beta \in B$ and their goal is to determine some value in $K(\alpha,\beta)$ while minimizing the communication (number of bits exchanged). Let Π be a deterministic communication protocol solving K. Then the *communication cost* of Π, denoted $\mathsf{CC}(\Pi)$, is defined to be the maximum number of bits exchanged on any pair of inputs (α,β) when following Π. Let $\mathsf{CC}(K)$ denote the minimum cost over all protocols solving K. Recall that the leaves of a protocol induce a partition of $A \times B$ into combinatorial rectangles. We denote the number of such combinatorial rectangles in a protocol Π by $\mathsf{monorect}(\Pi)$ and the minimum number of leaves in a protocol solving K by $\mathsf{monorect}(K)$.

We will often work with the communication matrix M_K of dimensions $|A| \times |B|$ associated with a function K. The rows and columns of M_K are indexed by the elements of A and B, respectively. The (α,β)-th entry of M_K is defined to be $K(\alpha,\beta)$. The leaves of a protocol Π solving K partitions the communication matrix M_K into $\mathsf{monorect}(\Pi)$ many monochromatic

---

[5] If all $\wedge$ and $\vee$ gates have fan-in two, then the number of leaves is always exactly one more than the number of gates (not counting negation gates) in F, which is a measure often used to describe circuit size.

combinatorial rectangles, where a combinatorial rectangle $A' \times B'$ ($A' \subseteq A$, $B' \subseteq B$) is called monochromatic if there exists $o \in O$ with $\forall(\alpha, \beta) \in A' \times B' : o \in K(\alpha, \beta)$. We will often use K and M_K interchangeably.

3 Hazard-Derivatives and the Monotone Barrier

For $x, y \in \{0,1\}^n$, we define $x \oplus \mathsf{u} \cdot y$ to be the string $\alpha \in \{0, \mathsf{u}, 1\}^n$ such that for all $i \in [n]$, $\alpha_i = x_i$ if $y_i = 0$, and otherwise $\alpha_i = \mathsf{u}$. Let f be a Boolean function on n variables. Its *hazard derivative* is a Boolean function on $2n$ variables denoted $\mathsf{d}f(x; y)$ that evaluates to 1 if and only if $\widetilde{f}(x \oplus \mathsf{u} \cdot y) = \mathsf{u}$, i.e., there are two resolutions of $x \oplus \mathsf{u} \cdot y$, say a and b, such that $f(a) = 0$ and $f(b) = 1$. In other words, $\mathsf{d}f(x; y) = 1$ if and only if the function f is *not* constant on the subcube of all resolutions of $x \oplus \mathsf{u} \cdot y$. For example consider the multiplexer function $\mathsf{MUX}_n(s, x)$ which is defined as: $\mathsf{MUX}_n(s_1, \ldots, s_n, x_0, x_1, \ldots, x_{2^n-1}) = x_{\mathrm{bin}(s_1,\ldots,s_n)}$, where $\mathrm{bin}(s_1, \ldots, s_n)$ is the natural number represented by the binary number $s_1 s_2 \cdots s_n$. Its hazard derivative $\mathsf{dMUX}_n(s, x; t, y) = 1$ if and only if the string $x \oplus \mathsf{u} \cdot y$ restricted to the positions given by the subcube of all resolutions of $s \oplus \mathsf{u} \cdot t$ is neither the all zeroes nor the all ones string. Equivalently, we can say that either x restricted to the positions given by the subcube of all resolutions of $s \oplus \mathsf{u} \cdot t$ is not all zeroes or all ones, or y restricted to the positions given by the subcube of all resolutions of $s \oplus \mathsf{u} \cdot t$ is not all zeroes.

Notice that for a fixed value $a \in \{0,1\}^n$, the function $\mathsf{d}f(a; y) : \{0,1\}^n \to \{0,1\}$, is a monotone function, see [27, Lemma 4.6]. The key observation that connects hazard-free circuits to monotone circuits is that given a hazard-free circuit C for f and any Boolean string a, we can construct a monotone circuit for $\mathsf{d}f(a; y)$ that is no larger in size than C, see [27, Thm. 4.9]. Therefore, in order to prove lower-bounds for hazard-free circuits for f, one only needs to identify a Boolean string a such that $\mathsf{d}f(a; y)$ is a hard function for monotone circuits, i.e., has high monotone circuit complexity. This allows us to transfer a wealth of known monotone circuit lower bounds to the hazard-free world. This works analogously for hazard-free De Morgan formulas (in this case the hazard-derivative can be constructed as a monotone De Morgan formula instead of a monotone circuit).

The best known construction for hazard-free De Morgan formulas for MUX_n has size $2 \cdot 3^n - 1$ (See Theorem 19). Can we use derivatives to prove that this is optimal? No. We show that *all* derivatives of MUX_n have monotone De Morgan formulas of size at most $(n+1)2^n$. This is an instance of the *monotone barrier.*

▶ **Proposition 1.** *Fix any* $(s, x) \in \{0,1\}^{n+2^n}$. *The function* $\mathsf{dMUX}_n(s, x; t, y) : \{0,1\}^{n+2^n} \to \{0,1\}$ *has monotone De Morgan formulas of size at most* $(n+1)2^n$.

Proof. Fix $(s, x) \in \{0,1\}^{n+2^n}$. Recall $\mathsf{dMUX}_n(s, x; t, y) = 1$ if and only if x restricted to the positions given by the subcube of all resolutions of $s \oplus \mathsf{u} \cdot t$ is not all zeroes or all ones, or y restricted to the positions given by the subcube of all resolutions of $s \oplus \mathsf{u} \cdot t$ is not all zeroes.

For the sake of clarity we first give a monotone implementation of dMUX_n when (s, x) is fixed to the all zeroes input. When $(s, x) = (0, 0)$ (i.e., the all zeroes string), then x restricted to the positions given by the subcube of all resolutions of $s \oplus \mathsf{u} \cdot t$ is all zeroes. Therefore, $\mathsf{dMUX}_n(0, 0; t, y) = 1$ if and only if y restricted to the positions given by the subcube of all resolutions of $s \oplus \mathsf{u} \cdot t$ is not the all zeroes string. Furthermore, since $s = 0$, the subcube of all resolutions of $s \oplus \mathsf{u} \cdot t$ is the set of points $b \in \{0,1\}^n$ such that $b \leq t$. Thus, $\mathsf{dMUX}_n(0, 0; t, y) = \bigvee_{b \in \{0,1\}^n} y_{\mathrm{bin}(b)} \wedge (\bigwedge_{i: b_i = 1} t_i)$.

For the general case when $(s, x) \in \{0,1\}^{n+2^n}$ is arbitrary (but fixed), we need to also verify whether x restricted to the positions given by the subcube of all resolutions of $s \oplus \mathsf{u} \cdot t$ is neither all zeroes nor all ones. Equivalently, for $b \in \{0,1\}^n$, when $x_{\mathrm{bin}(b)} \neq \mathsf{MUX}_n(s, x) = x_{\mathrm{bin}(s)}$, we need not check if $y_{\mathrm{bin}(b)} = 1$. Thus we have the following monotone implementation

$$\mathsf{dMUX}_n(s,x;t,y) = \left(\bigvee_{\substack{b\in\{0,1\}^n:\\ x_{\mathrm{bin}(b)}=x_{\mathrm{bin}(s)}}} y_{\mathrm{bin}(b)} \wedge \left(\bigwedge_{i:\, b_i\neq s_i} t_i\right)\right) \vee \left(\bigvee_{\substack{b\in\{0,1\}^n:\\ x_{\mathrm{bin}(b)}\neq x_{\mathrm{bin}(s)}}} \bigwedge_{i:\, b_i\neq s_i} t_i\right).$$

The size bound easily follows. ◀

The above proposition shows that the derivative method cannot yield a lower bound bigger than $(n+1)2^n$ for hazard-free De Morgan formulas for MUX_n. We now proceed to develop a framework that will allow us to prove that $2\cdot 3^n - 1$ is the optimal size for MUX_n. This is the first result that proves a hazard-free circuit lower bound without relying on an existing monotone circuit lower bound, i.e., that breaks the monotone barrier.

4 A Karchmer-Wigderson Game for Hazard-free Computation

In this section we give a natural generalization of the classical Karchmer-Wigderson game, which captures the complexity of *hazard-free* computation. We begin with recalling the framework of Karchmer-Wigderson games [29].

▶ **Definition 2** ([29]). *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. The* Karchmer-Wigderson game *of f, denoted KW_f, is the following communication problem: Alice gets $a \in \{0,1\}^n$ with $f(a)=1$, Bob gets $b\in\{0,1\}^n$ with $f(b)=0$ and their goal is to determine a coordinate $i\in[n]$ such that $a_i \neq b_i$.*

They also gave the following monotone version of the game.

▶ **Definition 3** ([29]). *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a monotone Boolean function. The* monotone Karchmer-Wigderson game *of f, denoted KW^+_f, is the following communication problem: Alice gets $a\in\{0,1\}^n$ with $f(a)=1$, Bob gets $b\in\{0,1\}^n$ with $f(b)=0$ and their goal is to determine a coordinate $i\in[n]$ such that $1 = a_i \neq b_i = 0$.*

The seminal work of Karchmer and Wigderson [29] showed that the communication complexity of the KW_f game (resp., KW^+_f game) characterizes the size and depth complexity of De Morgan formulas (resp., monotone formulas).

▶ **Theorem 4** ([29]). *Let $f\colon\{0,1\}^n\to\{0,1\}$ be a Boolean function. Then, $\mathsf{depth}(f) = \mathsf{CC}(\mathsf{KW}_f)$ and $\mathsf{size}(f) = \mathsf{monorect}(\mathsf{KW}_f)$. Let $f\colon\{0,1\}^n\to\{0,1\}$ be a monotone Boolean function. Then, $\mathsf{depth}^+(f) = \mathsf{CC}(\mathsf{KW}^+_f)$ and $\mathsf{size}^+(f) = \mathsf{monorect}(\mathsf{KW}^+_f)$.*

We now extend the Karchmer-Wigderson games to the hazard-free setting. For a Boolean function $f\colon\{0,1\}^n\to\{0,1\}$, recall from (1) that $\widetilde{f}\colon\{0,\mathsf{u},1\}^n\to\{0,\mathsf{u},1\}$ is the hazard-free extension of f.

▶ **Definition 5** (Hazard-free Karchmer-Wigderson game). *Let $f\colon\{0,1\}^n\to\{0,1\}$ be a Boolean function. The* hazard-free Karchmer-Wigderson game *of f, denoted $\mathsf{KW}^{\mathsf{u}}_f$, is the following communication problem: Alice gets $\alpha\in\{0,\mathsf{u},1\}^n$ with $\widetilde{f}(\alpha)=1$, Bob gets $\beta\in\{0,\mathsf{u},1\}^n$ with $\widetilde{f}(\beta)=0$ and their goal is to determine a coordinate $i\in[n]$ such that $\alpha_i\neq\beta_i$ and furthermore $\alpha_i\neq\mathsf{u}$ and $\beta_i\neq\mathsf{u}$.*

▶ Remark 6. Note that the wordy condition "$\alpha_i\neq\beta_i$ and $\alpha_i\neq\mathsf{u}$ and $\beta_i\neq\mathsf{u}$" is equivalent to the simple $\alpha_i\oplus\beta_i=1$, which is in complete analogy to $a_i\oplus b_i=1$ in the classical Karchmer-Wigderson game, see Def. 2, whereas we will show in Theorem 11 that our game is actually a generalization of the monotone Karchmer-Wigderson game to the domain of all Boolean functions.

Now using this generalized game $\mathsf{KW}^{\mathsf{u}}_f$ we characterize the complexity of hazard-free De Morgan formulas for Boolean functions.

▶ **Theorem 7.** *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *be a Boolean function. Then,* $\mathsf{depth}^{\mathsf{u}}(f) = \mathsf{CC}(\mathsf{KW}^{\mathsf{u}}_f)$ *and* $\mathsf{size}^{\mathsf{u}}(f) = \mathsf{monorect}(\mathsf{KW}^{\mathsf{u}}_f)$*.*

The proof is a natural generalization of the proof of Theorem 4 and is provided in Section 6.

▶ Remark 8. We remark that a variant of the game $\mathsf{KW}^{\mathsf{u}}_f$ has been considered in prior works [24, 14]. In this variant, the inputs to Alice and Bob remains the same but the goal is *different.* More formally, Alice gets $\alpha \in \widetilde{f}^{-1}(1)$, Bob gets $\beta \in \widetilde{f}^{-1}(0)$ and their goal is to determine a coordinate $i \in [n]$ such that $\alpha_i \neq \beta_i$. That is, now a coordinate where one of them has u and the other has 0 or 1 is a valid answer. This is the subtle but crucial difference with respect to our game (Definition 5), where we forbid such answers by requiring that $\alpha_i \neq \mathsf{u}$ and $\beta_i \neq \mathsf{u}$.

▶ Remark 9. Building on [58] and [51], [61] generalizes the Karchmer-Wigderson result from formulas to De Morgan circuits (i.e., Boolean circuits with negations only at the inputs). The corresponding communication games are played on directed acyclic graphs. We remark that their proofs also work for hazard-free complexity, so we get a tight correspondence between hazard-free De Morgan circuit size and the correspoding hazard-free KW-game.

4.1 Restriction to prime implicants and prime implicates

We now prove that we can restrict our attention to small (in some cases significantly smaller) submatrices of the communication matrix. We will use this restricted version of the game to show that for monotone functions the hazard-free Karchmer-Wigderson game is *equivalent* to the monotone Karchmer-Wigderson game, see Theorem 11.

▶ **Theorem 10.** *For any function f, the complexity (works for size and also for depth) of the game* $\mathsf{KW}^{\mathsf{u}}_f$ *remains unchanged even if we restrict Alice's input to prime implicants and Bob's input to prime implicates.*

Proof. The complexity of the restricted game is obviously at most the complexity of the original game, since the game is now being played on a submatrix of the original matrix. For the other direction, observe that given an arbitrary implicant α and an arbitrary implicate β, Alice can choose a prime implicant α' that is obtained by flipping some stable bits in α to u and Bob can choose a prime implicate β' that is obtained by flipping some stable bits in β to u, and play the restricted game on the input (α', β'). Any valid answer in the restricted game is also a valid answer in the original game, since we are only flipping stable bits to u. This proves that the complexity of the original game is at most the complexity of the restricted game. Therefore, both games have the same complexity. ◀

We will use this equivalent reduced form of the hazard-free Karchmer-Wigderson game in the rest of the paper.

There is a natural counterpart to Theorem 10 in the monotone world: in Definition 3, we can assume without loss of generality that Alice's input has minimal number of ones and Bob's input has maximal number of ones. We show that we can view the hazard-free Karchmer-Wigderson game as a generalization of the monotone Karchmer-Wigderson game.

▶ **Theorem 11.** *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *be a* monotone *function. Then, the games* $\mathsf{KW}^{\mathsf{u}}_f$ *and* KW^{+}_f *are equivalent.*

Proof. First, we show that the complexity of $\mathsf{KW}^{\mathsf{u}}_f$ is at most that of KW^{+}_f. Using Theorem 10, we can assume that Alice's input is a prime implicant and Bob's input is a prime implicate. Since f is monotone, any prime implicant of f contains only 1s and u's. Similarly, any prime implicate of f contains only 0s and u's. Now, Alice can flip every u in her input to 0 and Bob can flip every u in his input to 1 and play the game KW^{+}_f. Notice that by Definition 3, the output of this game will be a position where Alice's and Bob's input had different stable values originally.

For the other direction, Alice can flip every 0 in her input to u and Bob can flip every 1 in his input to u. Since f is monotone, Alice still has an input in $\widetilde{f}^{-1}(1)$ and Bob still has an input in $\widetilde{f}^{-1}(0)$. Now, the output of the game $\mathsf{KW}^{\mathsf{u}}_f$ on these new inputs will also be a valid output for the game KW^{+}_f since all stable bits in Alice's input are 1 and all stable bits in Bob's input are 0. ◀

▶ Remark 12. We note another perspective on hazard-free KW-game through the lens of monotone KW-game. For this purpose we need to associate sets with implicants and implicates. For an implicant $\alpha \in \widetilde{f}^{-1}(1)$ we define the set S_α of literals as follows: if $\alpha_i = 1$, then $x_i \in S_\alpha$ and if $\alpha_i = 0$, then $\neg x_i \in S_\alpha$. In particular, if $\alpha_i = \mathsf{u}$ then neither x_i nor $\neg x_i$ belongs to S_α. Furthermore, α is said to be a *prime* implicant if S_α is *minimal* with respect to set inclusion. That is, no strict subset of S_α is also an implicant. Analogously, for an implicate $\beta \in \widetilde{f}^{-1}(0)$ we define the set T_β of literals as follows: if $\beta_i = 1$, then $\neg x_i \in T_\beta$ and if $\beta_i = 0$, then $x_i \in T_\beta$. Furthermore, β is said to be a *prime* implicate if T_β is *minimal* with respect to set inclusion.

By using Theorem 10 and its counterpart in the monotone world, we observe that *both* the monotone KW-game and the hazard-free KW-game has the exact same definition: Alice gets a set of literals corresponding to a prime implicant and Bob gets a set of literals corresponding to a prime implicate. Their goal is to find a literal in the intersection of the two sets.

This means that playing (this reduced version of) the hazard-free KW-game is the same as playing the monotone KW-game on a function that is not necessarily monotone.

▶ Remark 13. It is instructive to gain yet another perspective on hazard-free KW-games via an application of a well-known result in communication complexity that informally says, for every communication total search problem S (in particular also for the hazard-free KW-game) there exists a partial monotone function g such that the monotone KW-game for g is equivalent to the communication problem S. For a formal statement see [18, Lemma 2.3] or [50, Proposition 2.10] (and references therein). In our setting the partial monotone function $g : \{0,1\}^{2n} \to \{0,1,*\}$ is defined as follows: an input $(y_1, \ldots, y_n, z_1, \ldots, z_n)$ is said to be admissible if for all $i \in [n]$ we have $y_i \cdot z_i = 0$, otherwise we call the input inadmissible. On an inadmissible input g is undefined, which is denoted by $*$. Given an admissible input (y, z), let us define $\alpha \in \{0, \mathsf{u}, 1\}^n$ such that for all $i \in [n]$, $\alpha_i = y_i$ if $y_i \oplus z_i = 1$, and otherwise $\alpha_i = \mathsf{u}$. Then, $g(y, z) = \Psi(\widetilde{f}(\alpha))$, where $\Psi(0) = 0$, $\Psi(1) = 1$, and $\Psi(\mathsf{u}) = *$.

Now it can be shown (see, e.g., [50, Proposition 2.10]) that the hazard-free KW-game for the Boolean function f is equivalent to the monotone KW-game for the partial function g. Thus the monotone KW-games *on partial functions* are expressive enough to capture hazard-free KW-games. Note that g has twice as many inputs as f.

5 Hazard-Free Formulas for the Multiplexer Function

We now use the hazard-free Karchmer-Wigderson game to give improved constructions of hazard-free formulas as well as proofs of their optimality. Our starting point is the observation that the commonly used hazard-free formula for MUX_1 in Figure 1(b) is *not* optimal w.r.t.

size. We find an optimal formula for it (Figure 1c) which in turn leads to an optimal formula of size $2 \cdot 3^n - 1$ for MUX_n. Following the discussion in Subsection 1.2, this upper bound also applies to all n-bit Boolean functions and improves upon Huffman's construction [26]. However, a gap between the upper and lower bound still remains. We begin with some necessary basics on the multiplexer function.

5.1 The Multiplexer Function and its Communication Matrix

Recall, the multiplexer function $\mathsf{MUX}_n \colon \{0,1\}^{n+2^n} \to \{0,1\}$ is a Boolean function on $n + 2^n$ variables defined as

$$\mathsf{MUX}_n(s_1, \ldots, s_n, x_0, x_1, \ldots, x_{2^n-1}) = x_{\mathrm{bin}(s_1,\ldots,s_n)}, \tag{2}$$

where $\mathrm{bin}(s_1, \ldots, s_n)$ is the natural number represented by the binary number $s_1 s_2 \cdots s_n$. We will be studying the communication matrix $M_{\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}}$ of the hazard-free game $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$. Following Theorem 10, we will restrict our attention to the submatrix given by prime implicants and implicates of MUX_n. The following proposition gives the structure of the prime implicants and prime implicates.

▶ **Proposition 14.** *For any $n \geq 1$ and any string $\alpha \in \{0, \mathsf{u}, 1\}^n$, there exist unique strings $\alpha' \in \{\mathsf{u}, 1\}^{2^n}$ and $\beta' \in \{0, \mathsf{u}\}^{2^n}$ such that $\alpha\alpha' \in \{0, \mathsf{u}, 1\}^{n+2^n}$ is a prime implicant of MUX_n and $\alpha\beta' \in \{0, \mathsf{u}, 1\}^{n+2^n}$ is a prime implicate of MUX_n.*

Proof. For $\alpha \in \{0, \mathsf{u}, 1\}^n$, consider the string $\alpha' \in \{\mathsf{u}, 1\}^{2^n}$ that has 1s at positions indexed by the resolutions of α and that has u's elsewhere. We have $\widetilde{\mathsf{MUX}}_n(\alpha\alpha') = 1$ showing that $\alpha\alpha'$ is an implicant. We now show it is a prime implicant. If any 1s in α' are made a u, then the output becomes a u, because a resolution of α now indexes into a u. If any Boolean value in α is made a u, then at least one resolution of the selector bits is a position in the data bits that is a u. Therefore this implicant is minimal. This is also that only prime implicant that can be obtained by extending α because the α' part is minimal and all 1s in it are necessary. The argument for prime implicates is symmetric. ◀

The following proposition states the inductive structure of communication matrices of MUX_n.

▶ **Proposition 15.** *The communication matrix of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$, when restricted to prime implicants and prime implicates, has the following inductive structure:*

- *For $n = 1$,*

$$M_{\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_1}} = \begin{array}{c} \\ 01\mathsf{u} \\ \mathsf{u}11 \\ 1\mathsf{u}1 \end{array} \!\!\begin{array}{c} \begin{array}{ccc} 00\mathsf{u} & \mathsf{u}00 & 1\mathsf{u}0 \end{array} \\ \begin{pmatrix} x_0 & x_0 & s \\ x_0 & x_0, x_1 & x_1 \\ s & x_1 & x_1 \end{pmatrix} \end{array}.$$

- *For $n \geq 2$,*

$$M_{\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}} = \begin{array}{c} \\ 0 \\ \mathsf{u} \\ 1 \end{array} \!\!\begin{array}{c} \begin{array}{ccc} 0 & \mathsf{u} & 1 \end{array} \\ \begin{pmatrix} M_0 & M_0 & s_1 \\ M_0 & M_0 \cup M_1 & M_1 \\ s_1 & M_1 & M_1 \end{pmatrix} \end{array},$$

where the row (resp., column) labeled $\gamma \in \{0, \mathsf{u}, 1\}$ represents the set of prime implicants (resp., prime implicates) with $s_1 = \gamma$. We define the formulas

$$\begin{aligned} F_0 &= \mathsf{MUX}_{n-1}(s_2 \ldots, s_{n-1}, x_0, \ldots, x_{2^{n-1}-1}) = \mathsf{MUX}_n(0, s_2, \ldots, s_n, x_0, x_1, \ldots, x_{2^n-1}), \\ F_1 &= \mathsf{MUX}_{n-1}(s_2, \ldots, s_n, x_{2^{n-1}}, \ldots, x_{2^n-1}) = \mathsf{MUX}_n(1, s_2, \ldots, s_n, x_0, x_1, \ldots, x_{2^n-1}), \end{aligned}$$

and matrices $M_0 := M_{\mathsf{KW}^{\mathsf{u}}_{F_0}}$, $M_1 := M_{\mathsf{KW}^{\mathsf{u}}_{F_1}}$, s_1 *stands for a block matrix of all entries* s_1*, and* $M_0 \cup M_1$ *is obtained by taking entry-wise union of* M_0 *and* M_1*. In other words,* $M_0 \cup M_1$ *represents the matrix where the* (i,j) *entry equals* $(M_0)_{i,j} \cup (M_1)_{i,j}$*.*

Note that in the communication matrix we have changed the entries from indices of variables to their labels, as in instead of $1, 2, 3$ *we write the more intuitive symbols* $s_1, \ldots, s_n, x_0, x_1, \ldots, x_{2^n-1}$ *for better readability.*

Proof. For $n = 1$ the proof follows by inspection and when $n \geq 2$ it follows from the following recursive decomposition: $\mathsf{MUX}_n = \mathsf{MUX}_1(s_1, F_0, F_1)$. ◀

We also need the following well-known general technique used in communication complexity that allows us to exploit repeated submatrices within a communication matrix.

▶ **Proposition 16.** *Let* M *be a communication matrix such that* $M = \begin{pmatrix} A & A \\ A & A \end{pmatrix}$ *or* $M = \begin{pmatrix} A & A \end{pmatrix}$*. Then,* $\mathsf{CC}(M) = \mathsf{CC}(A)$ *and* $\mathsf{monorect}(M) = \mathsf{monorect}(A)$*.*

5.2 Size optimal hazard-free formula

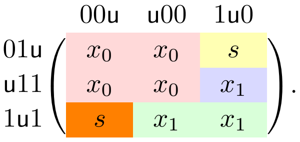
We now give the size optimal hazard-free formula for the multiplexer function. As a simple application of Theorem 7, we begin with finding optimal formulas for MUX_1.

▶ **Proposition 17.** *The optimal (size and depth) hazard-free De Morgan formula for* $\mathsf{MUX}_1(s, x_0, x_1)$ *has size* 5 *and depth* 3*.*

Proof. Consider the communication matrix of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_1}$ shown below,

$$\begin{array}{c} \\ 01\mathsf{u} \\ \mathsf{u}11 \\ 1\mathsf{u}1 \end{array} \begin{array}{c} \begin{array}{ccc} 00\mathsf{u} & \mathsf{u}00 & 1\mathsf{u}0 \end{array} \\ \begin{pmatrix} x_0 & x_0 & s \\ x_0 & x_0, x_1 & x_1 \\ s & x_1 & x_1 \end{pmatrix} \end{array}.$$

We find the following protocol for $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_1}$ by inspection:



$$\begin{array}{c} \\ 01\mathsf{u} \\ \mathsf{u}11 \\ 1\mathsf{u}1 \end{array} \begin{array}{c} \begin{array}{ccc} 00\mathsf{u} & \mathsf{u}00 & 1\mathsf{u}0 \end{array} \\ \begin{pmatrix} x_0 & x_0 & s \\ x_0 & x_0 & x_1 \\ s & x_1 & x_1 \end{pmatrix} \end{array}. \tag{3}$$

Using Lemma 30 with the above protocol, we obtain the hazard-free formula for MUX_1 shown in Figure 1c. The optimality of depth follows from the optimality of size. We defer the proof of the optimality of size to Theorem 23, the general case of MUX_n. ◀

▶ Remark 18. To demystify the construction in Figure 1c we note that it is simply the hazard-free DNF of MUX_1, the formula $(s \wedge x_1) \vee (\neg s \wedge x_0) \vee (x_0 \wedge x_1)$, with an application of distributivity of $\wedge$ over $\vee$ to reduce the size.

Now, using the recursive decomposition of MUX_n we obtain the following upper bound.

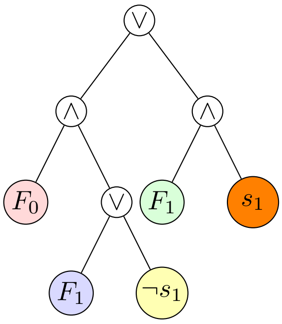


Figure 2 F: a size-optimal hazard-free formula for MUX_n.

▶ **Theorem 19.** *The multiplexer function* MUX_n *has hazard-free formulas of size* $2 \cdot 3^n - 1$ *and depth* $3n$ *for all* $n \geq 1$*.*

Proof. We construct the formula inductively. The construction for MUX_1 is given by Proposition 17. Recall that we can write $\mathsf{MUX}_n(s_1, \ldots, s_n, x_0, \ldots, x_{2^n-1})$ recursively as the formula $F = \mathsf{MUX}_1(s_1, F_0, F_1)$, where

$$F_0 = \mathsf{MUX}_{n-1}(s_2, \ldots, s_n, x_0, \ldots, x_{2^{n-1}-1}) \quad \text{and} \quad F_1 = \mathsf{MUX}_{n-1}(s_2, \ldots, s_n, x_{2^{n-1}}, \ldots, x_{2^n-1}).$$

By the induction hypothesis, both F_0 and F_1 have hazard-free formulas of size $2 \cdot 3^{n-1} - 1$ and depth $3(n-1)$. Using the hazard-free formula for MUX_1, given in Figure 1c, to implement F yields a formula of size $2 \cdot 3^n - 1$ and depth $3n$ for MUX_n.

It remains to prove that the constructed formula F is hazard-free. Using Lemma 29, it suffices to show that the protocol using F correctly solves the hazard-free KW-game $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$. In other words, the communication matrix of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ is partitioned into monochromatic rectangles by the protocol given by F. We will prove it by induction on n. The base case, $n = 1$, is given by Proposition 17. Now consider the inductive formula F for MUX_n shown in Figure 2. Following Lemma 29, when Alice and Bob reach the colored nodes in F (see Figure 2), then we obtain the following partition of $M_{\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}}$ as a block matrix, where $M_i := M_{\mathsf{KW}^{\mathsf{u}}_{F_i}}$ for $i \in \{0, 1\}$, and s_1 stands for a block matrix of all entries s_1:

$$\begin{array}{c|ccc|} & 0 & \mathsf{u} & 1 \\ \hline 0 & M_0 & M_0 & s_1 \\ \mathsf{u} & M_0 & M_0 & M_1 \\ 1 & s_1 & M_1 & M_1 \\ \hline \end{array},$$

where the row (resp., column) labeled $\gamma \in \{0, \mathsf{u}, 1\}$ represents the set of prime implicants (resp., implicates) with $s_1 = \gamma$. Now from the monochromatic partition of M_i using F_i, $i \in \{0, 1\}$, and Proposition 16, we get a monochromatic partition of the communication matrix of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$. ◀

▶ Remark 20. We remark that the construction in Theorem 19 can be made into an *alternating* formula: the communication matrix is symmetric and hence the monochromatic partition can be transposed so that Bob starts the communication. We now use this transposed partition on the blue F_1 node in Figure 2.

We now prove that the above construction for MUX_n is optimal with respect to size. For this purpose, we study the communication problem associated with the following *subcube intersection* function: $\mathsf{subcube\text{-}intersect}_n \colon \{0, \mathsf{u}, 1\}^n \times \{0, \mathsf{u}, 1\}^n \to \{0, 1\}$, where

$\mathsf{subcube\text{-}intersect}_n(\alpha,\beta) = 1$ if and only if the subcubes defined by α and β in $\{0,1\}^n$ intersect, i.e., if α and β have a common resolution. We note that the subcube intersection function is the same as the equality function when restricting its domain of definition to Boolean values only. The equality function is widely used in classical communication complexity for proving lower bounds. We also note that the subcube intersection function cannot be implemented by any circuit over $\{0,\mathsf{u},1\}$ (and hence in particular is not the hazard-free extension of any Boolean function), even for $n = 1$, because $\mathsf{subcube\text{-}intersect}_1(\mathsf{u},\mathsf{u}) = 1$, but $\mathsf{subcube\text{-}intersect}_1(0,1) = 0$ [6]. Let us see how the subcube intersection problem helps in capturing the complexity of the hazard-free game $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$.

▶ **Lemma 21.** *The* $\mathsf{subcube\text{-}intersect}_n$ *communication problem reduces to the communication problem* $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ *with no extra cost.*

Proof. Given inputs $\alpha,\beta \in \{0,\mathsf{u},1\}^n$ to the $\mathsf{subcube\text{-}intersect}_n$ problem, Alice and Bob modify their inputs as follows *without* communication.

- Alice constructs $\alpha' \in \{\mathsf{u},1\}^{2^n}$ such that α' has ones only at the positions indexed by the subcube of resolutions of α.
- Bob constructs $\beta' \in \{\mathsf{u},0\}^{2^n}$ such that β' has zeroes only at the positions indexed by the subcube of resolutions of β.

Now they can solve the game $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ on inputs $\alpha\alpha'$ and $\beta\beta'$. Observe that if the subcubes α and β intersect then answers to $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ lie in the set of data variables $\{x_0,\ldots,x_{2^n-1}\}$, otherwise they lie in the set of selector variables $\{s_1,\ldots,s_n\}$. Therefore, from the answers to the $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ game they can deduce whether the subcubes intersect or not, again without communication. ◀

Using the rank lower bound technique of [42] (See also [34, Lemma 1.28] and the discussion following the lemma.), we know that

$$\mathsf{monorect}(\mathsf{subcube\text{-}intersect}_n) \geq 2\cdot \mathsf{rank}(M_{\mathsf{subcube\text{-}intersect}_n}) - 1, \tag{4}$$

where $M_{\mathsf{subcube\text{-}intersect}_n}$ is interpreted as a matrix over $\mathbb{R}$ with 0s and 1s as entries. We prove the following tight bound on the rank of $M_{\mathsf{subcube\text{-}intersect}_n}$:

▶ **Lemma 22.** *The communication matrix of* $\mathsf{subcube\text{-}intersect}_n$ *is of* full *rank. That is, the rank of* $M_{\mathsf{subcube\text{-}intersect}_n}$ *equals* 3^n *for all* $n \geq 1$*.*

This immediately implies our size lower bound:

▶ **Theorem 23.** *Any hazard-free formula for* MUX_n *requires* $2\cdot 3^n - 1$ *leaves for all* $n \geq 1$*.*

Proof. Using Theorem 7, it is sufficient to show that the communication matrix of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ requires $2\cdot 3^n - 1$ monochromatic rectangles, i.e., $\mathsf{monorect}(\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}) \geq 2\cdot 3^n - 1$. This is readily checked:

$$\begin{aligned}\mathsf{monorect}(\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}) &\overset{\text{Lem. 21}}{\geq} \mathsf{monorect}(\mathsf{subcube\text{-}intersect}_n)\\ &\overset{(4)}{\geq} 2\cdot \mathsf{rank}(M_{\mathsf{subcube\text{-}intersect}_n}) - 1 \overset{\text{Lem. 22}}{=} 2\cdot 3^n - 1.\end{aligned}$$ ◀

---

[6] In any circuit implementation, if $C(\alpha) = 1$, then for all resolutions a of α we also have $C(a) = 1$, which is easily seen by induction. Alternatively, this can be seen by the fact that all gates (and hence the whole circuit) are monotone with respect to the partial order of stability ($\mathsf{u} \sqsubseteq 0$, $\mathsf{u} \sqsubseteq 1$, 0 and 1 incomparable), so switching unstable inputs to stable inputs can only keep an output u or switch an output from u to a stable value, but not change a stable output.

It now remains to prove Lemma 22.

Proof of Lemma 22. We prove it by induction on n. For the base case, $n = 1$ and the communication matrix $M_{\textsf{subcube-intersect}_1}$ is as follows:

$$\begin{array}{c|ccc} & 0 & \mathsf{u} & 1 \\ \hline 0 & 1 & 1 & 0 \\ \mathsf{u} & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{array}.$$

Clearly $\mathsf{rank}(M_{\textsf{subcube-intersect}_1}) = 3$. Now consider the communication matrix of $M_{\textsf{subcube-intersect}_n}$. We claim that it looks as follows:

$$\begin{array}{c|ccc} & 0 & \mathsf{u} & 1 \\ \hline 0 & M_{\textsf{subcube-intersect}_{n-1}} & M_{\textsf{subcube-intersect}_{n-1}} & 0 \\ \mathsf{u} & M_{\textsf{subcube-intersect}_{n-1}} & M_{\textsf{subcube-intersect}_{n-1}} & M_{\textsf{subcube-intersect}_{n-1}} \\ 1 & 0 & M_{\textsf{subcube-intersect}_{n-1}} & M_{\textsf{subcube-intersect}_{n-1}} \end{array},$$

where the row labeled $\gamma \in \{0, \mathsf{u}, 1\}$ represents the set of rows labeled with $\alpha \in \{0, \mathsf{u}, 1\}^n$ such that $\alpha_1 = \gamma$ and similarly for the columns. The validity of the claim follows from inspection that on fixing the first variables we either know the answer or have self-reduced it to a smaller instance. Therefore, we obtain:

$$M_{\textsf{subcube-intersect}_n} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \otimes M_{\textsf{subcube-intersect}_{n-1}},$$

where $\otimes$ is the Kronecker product of matrices. Hence, using the fact that rank is multiplicative with respect to Kronecker product, we have

$$\mathsf{rank}(M_{\textsf{subcube-intersect}_n}) = \mathsf{rank}(M_{\textsf{subcube-intersect}_1}) \cdot \mathsf{rank}(M_{\textsf{subcube-intersect}_{n-1}}).$$

Now using the induction hypothesis completes the proof. ◀

Translating the size lower bound to depth gives the following corollary.

▶ **Corollary 24.** *For all $n \geq 1$, $\mathsf{depth}^{\mathsf{u}}(\mathsf{MUX}_n) \geq \lceil \log_2(2 \cdot 3^n - 1) \rceil$.*

5.3 Formulas of improved depth

The lower bound from Corollary 24 on hazard-free formula depth is at least $1 + (\log_2 3) \cdot n$ for large n. However, our construction in Theorem 19 gives an upper bound of $3n$. We now give an improved construction (Theorem 25) with respect to depth while increasing the size by a factor of $\frac{9}{8}$. In contrast, the depth-optimal version of Huffman's construction is larger than the optimal size hazard-free formula by a multiplicative factor that is exponential in n.

▶ **Theorem 25.** *The multiplexer function MUX_n has hazard-free formulas of depth $2n + 1$ and size at most $2.25 \cdot 3^n - \frac{n}{2} - 1.25$ for all $n \geq 1$.*

From Theorem 7 we know it is sufficient to give a protocol Π solving the hazard-free Karchmer-Wigderson game of MUX_n such that $\mathsf{CC}(\Pi) \leq 2n + 1$ and $\mathsf{monorect}(\Pi) \leq 2.25 \cdot 3^n - \frac{n}{2} - 1.25$. We consider a monochromatic extension of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$. We extend the communication matrix of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ as follows to define the extended version $\mathsf{e}\text{-}\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$:

$$\begin{array}{cc} & \text{prime implicates} \\ \text{prime implicants} & \left(\begin{array}{c} M_{\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}} \\ \hline 0 \end{array}\right). \end{array} \tag{5}$$

In other words, the communication matrix of the extended version $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ is obtained by adding a rectangular block with all 0s to either the set of rows of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ (as shown above) or the set of columns. We note that the added block could have any number of rows in the former case or any number of columns in the latter. Further we will *always* require that the extended part be filled with a number that does not appear in $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$. Observe that 0 doesn't appear in the communication matrix of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$. However we could have used any other number that doesn't appear in $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$. We will denote all such extensions by $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$.

Clearly the following proposition holds.

▶ **Proposition 26.** *A protocol* Π *for* $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ *gives a protocol* Π' *for* $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ *such that* $\mathsf{CC}(\Pi') \leq \mathsf{CC}(\Pi)$ *and* $\mathsf{monorect}(\Pi') \leq \mathsf{monorect}(\Pi)$.

Proof. Follows from the definition (5) of $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$. ◀

Therefore, to prove the depth bound in Theorem 25 we will give a protocol for $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ with communication cost at most $2n + 1$.

▶ **Lemma 27.** *There is a protocol solving* $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ *such that its communication cost is at most* $2n + 1$.

Proof. From Proposition 15 we know that the communication matrix of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ looks as follows

$$\begin{array}{cc} & \begin{array}{ccc} 0 & \mathsf{u} & 1 \end{array} \\ \begin{array}{c} 0 \\ \mathsf{u} \\ 1 \end{array} & \left(\begin{array}{ccc} M_0 & M_0 & s_1 \\ M_0 & M_0 \cup M_1 & M_1 \\ s_1 & M_1 & M_1 \end{array}\right), \end{array}$$

where we define the formulas:

$$F_0 = \mathsf{MUX}_{n-1}(s_2, \ldots, s_n, x_0, \ldots, x_{2^{n-1}-1})$$
$$F_1 = \mathsf{MUX}_{n-1}(s_2, \ldots, s_n, x_{2^{n-1}}, \ldots, x_{2^n-1})$$

and matrices $M_i := M_{\mathsf{KW}^{\mathsf{u}}_{F_i}}$ for $i \in \{0, 1\}$. Therefore, the matrix of extended version $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ looks as follows

$$\begin{array}{cc} & \begin{array}{ccc} 0 & \mathsf{u} & 1 \end{array} \\ \begin{array}{c} 0 \\ \mathsf{u} \\ 1 \\ \\ \end{array} & \left(\begin{array}{ccc} M_0 & M_0 & s_1 \\ M_0 & M_0 \cup M_1 & M_1 \\ s_1 & M_1 & M_1 \\ 0 & 0 & 0 \end{array}\right). \end{array}$$

We now give a protocol to partition this matrix into monochromatic rectangles. This will be done inductively.

Base case: $n = 1$. The matrix of $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_1}$ can be monochromatically partitioned as follows

$$\begin{array}{c|ccc|}
 & 00\mathsf{u} & \mathsf{u}00 & 1\mathsf{u}0\\
\hline
01\mathsf{u} & x_0 & x_0 & s\\
\mathsf{u}11 & x_0 & x_0, x_1 & x_1\\
1\mathsf{u}1 & s & x_1 & x_1\\
 & 0 & 0 & 0
\end{array}.$$

This is obtained from the protocol where Alice sends the first bit indicating whether her input lies in the top or bottom part of red line. Then Bob sends a bit indicating whether his input lies in the left part or the right part of the blue line(s). Finally Alice sends one last bit to indicate whether her input lies in the top or bottom part of the orange line(s). Clearly the communication cost is 3.

Induction step: $n \geq 2$. Now Alice and Bob send one bit of communication each to reduce $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ to the following partition

$$\begin{array}{c|ccc|}
 & 0 & \mathsf{u} & 1\\
\hline
0 & M_0 & M_0 & s_1\\
\mathsf{u} & M_0 & M_0 \cup M_1 & M_1\\
1 & s_1 & M_1 & M_1\\
 & 0 & 0 & 0
\end{array}.$$

Observe that the top-right block $\begin{pmatrix} s_1 \\ M_1 \end{pmatrix}$ is the matrix of an $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_{n-1}}$. The bottom-right block $\begin{pmatrix} M_1 & M_1 \\ 0 & 0 \end{pmatrix}$ can be solved, using Proposition 16, at the cost of solving $\begin{pmatrix} M_1 \\ 0 \end{pmatrix}$ which is a block of $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_{n-1}}$. Similarly, the top-left block $\begin{pmatrix} M_0 & M_0 \\ M_0 & M_0 \cup M_1 \end{pmatrix}$ can be solved, using Proposition 16, at the cost of solving M_0 which is a block of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_{n-1}}$. Therefore, by Proposition 26 it has less complexity (size and depth) than $\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_{n-1}}$. Finally, note that the bottom-left block $\begin{pmatrix} s_1 \\ 0 \end{pmatrix}$ just needs one bit of communication to monochromatically partition it. Therefore, we have

$$\begin{aligned}
\mathsf{CC}(\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_n}) &\leq 2 + \max\left\{\mathsf{CC}(\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_{n-1}}), 1\right\}\\
&\leq 2 + \mathsf{CC}(\mathsf{e\text{-}KW}^{\mathsf{u}}_{\mathsf{MUX}_{n-1}}),\\
&\leq 2n + 1,
\end{aligned}$$

where the second inequality follows because $n \geq 2$ and the third follows from the induction hypothesis. ◀

We are now ready to prove Theorem 25.

Proof of Theorem 25. As mentioned in the beginning we will give a protocol to partition the communication matrix of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ into monochromatic rectangles such that the communication cost of this protocol is at most $2n + 1$ and the number of monochromatic rectangles is at most $2.25 \cdot 3^n - \frac{n}{2} - 1.25$ for all $n \geq 1$. Our protocol is the same as the one given in the proof of Lemma 27. Thus the upper bound on depth follows readily. To bound the size we count the number of monochromatic rectangles in the partition given by the protocol.

Define $T(n)$ to be the number of monochromatic rectangles in the partition of the communication matrix of $\text{e-KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ given by the protocol. Further define $S(n)$ to be the number of monochromatic rectangles covering only the entries of $\mathsf{KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$ in the partition of the communication matrix of $\text{e-KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$. In particular, $S(n) < T(n)$. Note that $S(n)$ gives the bound on the size we are interested in. We now write recurrences for $S(n)$ and $T(n)$ using the partition given by the induction step and the base case in Lemma 27. Recall the partition in the induction step looks as shown below:

$$\begin{array}{c@{}c} & \begin{array}{ccc} 0 & \mathsf{u} & 1 \end{array} \\ \begin{array}{c} 0\\ \mathsf{u}\\ 1\\ \end{array} & \left(\begin{array}{cc|c} M_0 & M_0 & s_1 \\ M_0 & M_0\cup M_1 & M_1 \\ \hline \multicolumn{1}{c|}{s_1} & M_1 & \multicolumn{1}{c}{M_1} \\ \multicolumn{1}{c|}{0} & 0 & \multicolumn{1}{c}{0} \end{array}\right) \end{array}.$$

It follows that $S(n)$ and $T(n)$ satisfies the following recurrences for $n \geq 2$:

$$S(n) = 2 \cdot S(n-1) + T(n-1) + 1, \text{ and} \tag{6}$$

$$T(n) = S(n-1) + 2 \cdot T(n-1) + 2, \tag{7}$$

where $S(1) = 5$ and $T(1) = 7$. For solving these recurrences it is helpful to first solve the two auxiliary recurrences $S(n) + T(n)$ and $S(n) - T(n)$. One ultimately obtains $S(n) = 2.25 \cdot 3^n - \frac{n}{2} - 1.25$. ◀

▶ Remark 28. We note that in the induction step we are reducing to an instance of $\text{e-KW}^{\mathsf{u}}_{\mathsf{MUX}}$, in particular the top-right part of the matrix in the proof of Theorem 25. Therefore, we need to strengthen our induction hypothesis and begin with an instance of $\text{e-KW}^{\mathsf{u}}_{\mathsf{MUX}_n}$.

6 Proofs for the hazard-free Karchmer-Wigderson game

In this section we prove Theorem 7. We split the proof into two lemmas.

▶ **Lemma 29.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function and F be a hazard-free De Morgan formula computing it. Then,*

$$\mathsf{CC}(\mathsf{KW}^{\mathsf{u}}_f) \leq \mathsf{depth}(F), \quad \textit{and} \quad \mathsf{monorect}(\mathsf{KW}^{\mathsf{u}}_f) \leq \mathsf{size}(F).$$

▶ **Lemma 30.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function and Π be a protocol for the hazard-free* KW*-game* $\mathsf{KW}^{\mathsf{u}}_f$*. Then,*

$$\mathsf{depth}^{\mathsf{u}}(f) \leq \mathsf{CC}(\Pi), \quad \textit{and} \quad \mathsf{size}^{\mathsf{u}}(f) \leq \mathsf{monorect}(\Pi).$$

The proofs of Lemmas 29 and 30 are natural generalizations of their corresponding counterparts in the original setting of the KW-game. See the full version for the proofs.

7 Alternation Depth Two and Two-round Protocols

The following theorem determines the exact depth of hazard-free formulas for MUX_n of alternation depth two.

▶ **Theorem 31.** *For $n \geq 2$, we have $\mathsf{depth}^{\mathsf{u}}_2(\mathsf{MUX}_n) = 2n + 2$.*

Theorem 31 is proved by analyzing *two-round communication protocols* of the form: Alice sends some string, Bob replies with some string, and they settle on an answer. We then connect the existence of low-depth, hazard-free formulas of alternation depth two to the existence of short, prefix codes. We then apply the well-known Kraft's inequality that gives a necessary and sufficient condition for the existence of prefix codes to obtain a lower bound on the hazard-free formula depth. This is the only depth lower bound in this paper that does not follow directly from a size lower bound. See the full version for details.

8 Limited hazard-freeness

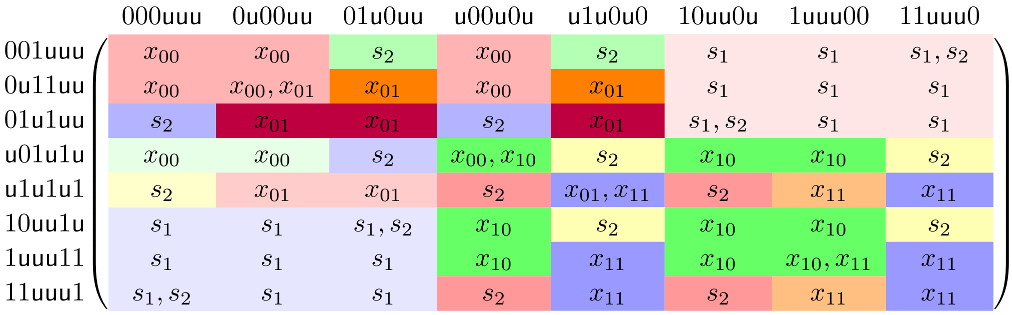
Avoiding all hazards can be very expensive (see [27, 28]) and sometimes is not needed, because we might have additional information about the input, for example when composing circuits. Under the (physically realistic in that setting) assumption that the input contains at most one u, the counter in [17] outputs the number of 1s in the input, encoded in binary Gray code with a single u, such that both resolutions of the output correspond to numbers whose difference is 1. Note that if Gray codes are not used, then in the usual binary bit representation one has to flip 4 bits to go from $7 = 0111_2$ to $8 = 1000_2$, hence any hazard-free counter would have to output uuuu on input 1111111u. When composing circuits, this additional information about the position of the u can be useful, which is shown for the task of sorting Gray code numbers in [35, 7, 8, 9]. Another interesting class of hazards that should be avoided are the inputs where the number of u's is bounded from above. This is the setting of k-bit hazard-freeness from [27].

Theorem 7 as stated is not directly applicable in these settings, as it only characterizes hazard-free formulas, i.e., formulas that are hazard-free with respect to *all* inputs. However, we can also treat formulas that avoid only certain hazards.

▶ **Proposition 32.** *Given sets A and B with $f^{-1}(1) \subseteq A \subseteq \widetilde{f}^{-1}(1)$ and $f^{-1}(0) \subseteq B \subseteq \widetilde{f}^{-1}(0)$. The hazard-free* KW*-game where Alice gets input from A and Bob gets input from B characterizes formulas that is hazard-free on inputs in $A \uplus B$.*

The proof is a straightforward generalization of the proof of Theorem 7 (and in particular of Lemmas 29 and 30).

▶ **Example 33.** Consider the function $\mathsf{MUX}_2(s_1, s_2, x_{00}, x_{01}, x_{10}, x_{11})$. We proved in Theorem 23 that any hazard-free formula for MUX_2 requires 17 leaves. Suppose we only want to be hazard-free on inputs where at least one selector bit is stable. Note that every other possible hazard is covered by the eight prime implicants and eight prime implicates of MUX_2 labeling the rows and columns of the following matrix. Therefore, we can obtain an improved upper bound for this task by showing that the following game has a protocol of size smaller than 17.



$$\begin{array}{c|cccccccc}
 & \text{000uuu} & \text{0u00uu} & \text{01u0uu} & \text{u00u0u} & \text{u1u0u0} & \text{10uu0u} & \text{1uuu00} & \text{11uuu0} \\
\hline
\text{001uuu} & x_{00} & x_{00} & s_2 & x_{00} & s_2 & s_1 & s_1 & s_1, s_2 \\
\text{0u11uu} & x_{00} & x_{00}, x_{01} & x_{01} & x_{00} & x_{01} & s_1 & s_1 & s_1 \\
\text{01u1uu} & s_2 & x_{01} & x_{01} & s_2 & x_{01} & s_1, s_2 & s_1 & s_1 \\
\text{u01u1u} & x_{00} & x_{00} & s_2 & x_{00}, x_{10} & s_2 & x_{10} & x_{10} & s_2 \\
\text{u1u1u1} & s_2 & x_{01} & x_{01} & s_2 & x_{01}, x_{11} & s_2 & x_{11} & x_{11} \\
\text{10uu1u} & s_1 & s_1 & s_1, s_2 & x_{10} & s_2 & x_{10} & x_{10} & s_2 \\
\text{1uuu11} & s_1 & s_1 & s_1 & x_{10} & x_{11} & x_{10} & x_{10}, x_{11} & x_{11} \\
\text{11uuu1} & s_1, s_2 & s_1 & s_1 & s_2 & x_{11} & s_2 & x_{11} & x_{11}
\end{array}$$

Indeed, the above colouring yields a protocol of size 16. We can also show that we cannot do better. Substitute $s_1 = s_2 = \left(\begin{smallmatrix}1 & 0\\ 0 & 0\end{smallmatrix}\right)$ and $x_{00} = x_{01} = x_{10} = x_{11} = \left(\begin{smallmatrix}0 & 0\\ 0 & 1\end{smallmatrix}\right)$. This yields the following 16×16 (block) matrix. Note that if the original communication matrix had a protocol that yields fewer than 16 combinatorial rectangles, then this block matrix must have rank less than 16 since all substituted matrices are rank one. However, the block matrix is full-rank, showing that 16 leaves are required.

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0\\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0\\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0\\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0\\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1\\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0\\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1\\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1
\end{pmatrix}$$

For a natural number k, a Boolean circuit C is said to have a k-bit hazard if there exists $\alpha \in \{0, \mathsf{u}, 1\}^n$ such that C has hazard at α and the number of u's in α is at most k. We can also obtain a k-bit hazard-free construction similar to [27, Theorem 5.3] using Proposition 32.

▶ **Theorem 34.** *For any Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, there exists a k-bit hazard-free formula of depth at most $2\log\left(\sum_{i=0}^{k}\binom{n}{i}\right) + 2k + \mathsf{depth}(f)$ and size at most $\left(\sum_{i=0}^{k}\binom{n}{i}\right)^2 \cdot 4^k \cdot \mathsf{size}(f)$.*

Proof. Following Proposition 32 it suffices to give a protocol for the hazard-free KW-game where Alice gets an implicant α with at most k u's and Bob gets an implicate β with at most k u's. The protocol is as follows:

1. Alice sends the set $A \subseteq [n]$ of positions of the u's in α.
2. Bob then sends the set $B \subseteq [n]$ of positions of the u's in β and the bits in β at all positions from $A \setminus B$.
3. Alice then sends the bits in α for the positions in $B \setminus A$.
4. For the positions in $A \cap B$, they decide beforehand to set them to a fixed constant, say 0. Thus no communication is required. They now have an instance of the classical game KW_f, which they solve optimally.

The correctness of the protocol follows easily. The total number of bits exchanged to reduce to KW_f is at most $2\log\left(\sum_{i=0}^{k}\binom{n}{i}\right) + 2k$. Alice and Bob decide beforehand that the bits for positions in $A \setminus B$ and $B \setminus A$ are sent in sorted (say, increasing) order of positions. Thus they need to exchange at most $2k$ bits to send the bits across. ◀

References

1 N. Alon and R. B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, January 1987. `doi:10.1007/BF02579196`.

2 A. E. Andreev. On one method of obtaining constructive lower bounds for the monotone circuit size. *Algebra and Logics*, 26(1):3–26, 1987.

3 R. Beals, T. Nishino, and K. Tanaka. On the complexity of negation-limited boolean networks. *SIAM Journal on Computing*, 27(5):1334–1347, 1998.

4 A. Becker, W. Hu, Y. Tai, P. Brisk, R. Kastner, and P. Ienne. Arbitrary precision and complexity tradeoffs for gate-level information flow tracking. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2017. `doi:10.1145/3061639.3062203`.

5 J. Brzozowski, Z. Esik, and Y. Iland. Algebras for hazard detection. In *Proc. 31st International Symposium on Multiple-Valued Logic*, 2001.

6 J. A. Brzozowski and C-J. H. Seger. *Asynchronous circuits.* Springer New York, 1995.

7 J. Bund, C. Lenzen, and M. Medina. Near-optimal metastability-containing sorting networks. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 226–231, 2017. `doi:10.23919/DATE.2017.7926987`.

8 J. Bund, C. Lenzen, and M. Medina. Optimal metastability-containing sorting networks. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 521–526, 2018. `doi:10.23919/DATE.2018.8342063`.

9 J. Bund, C. Lenzen, and M. Medina. Optimal metastability-containing sorting via parallel prefix computation. *IEEE Transactions on Computers*, 69(2):198–211, 2020. `doi:10.1109/TC.2019.2939818`.

10 S. H. Caldwell. *Switching Circuits and Logical Design.* John Wiley & Sons Inc, 1958.

11 A. K. Chandra and G. Markowsky. On the number of prime implicants. *Discrete Mathematics*, 24(1):7–11, 1978.

12 J. Edmonds, R. Impagliazzo, S. Rudich, and J. Sgall. Communication complexity towards lower bounds on circuit depth. *Comput. Complex.*, 10(3):210–246, 2001. `doi:10.1007/s00037-001-8195-x`.

13 E. B. Eichelberger. Hazard detection in combinational and sequential switching circuits. *IBM J. Res. Dev.*, 9(2):90–99, March 1965. `doi:10.1147/rd.92.0090`.

14 Y. Filmus, O. Meir, and A. Tal. Shrinkage under random projections, and cubic formula lower bounds for AC0 (extended abstract). In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021*, volume 185 of *LIPIcs*, pages 89:1–89:7. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.ITCS.2021.89`.

15 M. J. Fischer. The complexity of negation-limited networks—a brief survey. In *Automata Theory and Formal Languages*, pages 71–82. Springer, 1975.

16 S. Friedrichs, M. Függer, and C. Lenzen. Metastability-containing circuits. *IEEE Transactions on Computers*, 67(08):1167–1183, August 2018. `doi:10.1109/TC.2018.2808185`.

17 M. Függer, A. Kinali, C. Lenzen, and T. Polzer. Metastability-aware memory-efficient time-to-digital converters. In *2017 23rd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 49–56, 2017. `doi:10.1109/ASYNC.2017.12`.

18 A. Gál. A characterization of span program size and improved lower bounds for monotone span programs. *Computational Complexity*, 10(4):277–296, 2001. `doi:10.1007/s000370100001`.

19 M. Göös and T. Pitassi. Communication lower bounds via critical block sensitivity. In *Symposium on Theory of Computing, STOC 2014*, pages 847–856. ACM, 2014. `doi:10.1145/2591796.2591838`.

20 M. Goto. Application of logical mathematics to the theory of relay networks (in Japanese). *J. Inst. Elec. Eng. of Japan*, 64(726):125–130, 1949.

21 M. Grigni and M. Sipser. Monotone separation of logarithmic space from logarithmic depth. *Journal of Computer and System Sciences*, 50(3):433–437, 1995.

22 M. Grzegorz. Kleene logic and inference. *Bulletin of the Section of Logic*, 43(1/2):42–52, 2014.

23 D. Harnik and R. Raz. Higher lower bounds on monotone size. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 378–387. ACM, 2000. `doi:10.1145/335305.335349`.

24 J. Håstad. The shrinkage exponent of de morgan formulas is 2. *SIAM Journal on Computing*, 27(1):48–64, 1998. `doi:10.1137/S0097539794261556`.

25 W. Hu, J. Oberg, A. Irturk, M. Tiwari, T. Sherwood, D. Mu, and R. Kastner. On the complexity of generating gate level information flow tracking logic. *IEEE Transactions on Information Forensics and Security*, 7(3):1067–1080, June 2012. `doi:10.1109/TIFS.2012.2189105`.

26 D. A. Huffman. The design and use of hazard-free switching networks. *J. ACM*, 4(1):47–62, January 1957. `doi:10.1145/320856.320866`.

27 C. Ikenmeyer, B. Komarath, C. Lenzen, V. Lysikov, A. Mokhov, and K. Sreenivasaiah. On the complexity of hazard-free circuits. *J. ACM*, 66(4), 2019. `doi:10.1145/3320123`.

28 S. Jukna. Notes on hazard-free circuits. *SIAM Journal on Discrete Mathematics*, 35(2):770–787, 2021. `doi:10.1137/20M1355240`.

29 M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990. `doi:10.1137/0403021`.

30 V. M. Khrapchenko. Complexity of the realization of a linear function in the class of Π-circuits. *Mathematical Notes of the Academy of Sciences of the USSR*, 9(1):21–23, 1971.

31 S. C. Kleene. On notation for ordinal numbers. *The Journal of Symbolic Logic*, 3(4):150–155, 1938. URL: `http://www.jstor.org/stable/2267778`.

32 S. C. Kleene. *Introduction to Metamathematics*. North Holland, 1952.

33 S. Körner. *Experience and theory : an essay in the philosophy of science*. International library of philosophy and scientific method. Routledge & Kegan Paul, London, 1966.

34 E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1996. `doi:10.1017/CBO9780511574948`.

35 C. Lenzen and M. Medina. Efficient metastability-containing gray code 2-sort. In *2016 22nd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 49–56, 2016. `doi:10.1109/ASYNC.2016.18`.

36 S. Lozhkin. Tighter bounds on the complexity of control systems from some classes. *Mat. Voprosy Kibernetiki*, 6:189–214, 1996. in Russian.

37 S. Lozhkin and N. Vlasov. On multiplexer function complexity in the π-schemes class. *Kazan. Gos. Univ. Uchen. Zap. Ser. Fiz.-Mat. Nauki*, 151(2):98–106, 2009. in Russian. URL: `http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=uzku&paperid=750&option_lang=eng`.

38 S. Lozhkin and N. Vlasov. On the depth of the storage access function. *Moscow University Computational Mathematics and Cybernetics*, 35(2):97–104, 2011.

39 O. B. Lupanov. On a method of circuit synthesis. *Izvestia VUZ*, 1:120–140, 1958.

40 O. B. Lupanov. On the complexity of realization of functions of propositional calculus by formulas. *Problemy Kibernetiki*, 3:61–80, 1960. in Russian.

41 O. B. Lupanov. On the synthesis of some classes of control systems. *Problemy Kibernetiki*, 10:63–97, 1963. in Russian.

42 K. Mehlhorn and E. M. Schmidt. Las vegas is better than determinism in VLSI and distributed computing. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 330–337, 1982.

43 O. Meir. Toward better depth lower bounds: Two results on the multiplexor relation. *Comput. Complex.*, 29(1):4, 2020. `doi:10.1007/s00037-020-00194-8`.

44 M. Mendler, T. R. Shiple, and G. Berry. Constructive boolean circuits and the exactness of timed ternary simulation. *Formal Methods in System Design*, 40:283–329, 2012. `doi:10.1007/s10703-012-0144-6`.

45 M. Mukaidono. On the b-ternary logical function - a ternary logic considering ambiguity. *Systems, Computers, Controls*, 3(3):27–36, 1972.

46 M. Mukaidono. Advanced results on application of fuzzy switching functions to hazard detection. In P.P. Wong, editor, *Advances in Fuzzy Sets, Possibility Theory and Applications*, pages 335–349. Plenum Publishing Corporation, 1983.

47 M. Mukaidono. Regular ternary logic functions – ternary logic functions suitable for treating ambiguity. In *Proc. 13th International Symposium on Multiple-Valued Logic*, pages 286–291. IEEE Computer Society Press, 1983.

48 S. M. Nowick and D. L. Dill. Exact two-level minimization of hazard-free logic with multiple-input changes. In *1992 IEEE/ACM International Conference on Computer-Aided Design*, pages 626–630, November 1992. `doi:10.1109/ICCAD.1992.279301`.

49 T. Pitassi and R. Robere. Strongly exponential lower bounds for monotone computation. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*, pages 1246–1255. ACM, 2017. `doi:10.1145/3055399.3055478`.

50 T. Pitassi and R. Robere. Lifting nullstellensatz to monotone span programs over any field. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 1207–1219. ACM, 2018. `doi:10.1145/3188745.3188914`.

51 P. Pudlák. On extracting computations from propositional proofs (a survey). In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010*, volume 8 of *LIPIcs*, pages 30–41. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010. `doi:10.4230/LIPIcs.FSTTCS.2010.30`.

52 A. Rao and A. Yehudayoff. *Communication Complexity: and Applications.* Cambridge University Press, 2020. `doi:10.1017/9781108671644`.

53 R. Raz and P. McKenzie. Separation of the monotone NC hierarchy. *Comb.*, 19(3):403–435, 1999. `doi:10.1007/s004930050062`.

54 R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM (JACM)*, 39(3):736–744, 1992.

55 A. A. Razborov. Lower bounds for the monotone complexity of some boolean functions. In *Soviet Math. Dokl.*, volume 31, pages 354–357, 1985.

56 A. A. Razborov. Lower bounds on monotone complexity of the logical permanent. *Mathematical Notes of the Academy of Sciences of the USSR*, 37(6):485–493, 1985.

57 A. A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

58 A. A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya: Mathematics*, 59(1):205–227, 1995. `doi:10.1070/im1995v059n01abeh000009`.

59 J. Riordan and C. E. Shannon. The number of two-terminal series-parallel networks. *Journal of Mathematics and Physics*, 21(1-4):83–93, 1942.

60 C. E. Shannon. The synthesis of two-terminal switching circuits. *The Bell System Technical Journal*, 28(1):59–98, 1949.

61 D. Sokolov. Dag-like communication and its applications. In *International Computer Science Symposium in Russia*, pages 294–307. Springer, 2017.

62 K. Tanaka, T. Nishino, and R. Beals. Negation-limited circuit complexity of symmetric functions. *Information processing letters*, 59(5):273–279, 1996.

63 G. Tarawneh, M. Függer, and C. Lenzen. Metastability tolerant computing. In *2017 23rd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 25–32, 2017. `doi:10.1109/ASYNC.2017.9`.

64 E. Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.

65 G. Tardos and U. Zwick. The communication complexity of the universal relation. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity, 1997*, pages 247–259. IEEE Computer Society, 1997. `doi:10.1109/CCC.1997.612320`.

66 M. Tiwari, H. M. G. Wassel, B. Mazloom, S. Mysore, F. T. Chong, and T. Sherwood. Complete information flow tracking from the gates up. *SIGARCH Comput. Archit. News*, 37(1):109–120, March 2009. `doi:10.1145/2528521.1508258`.

67 M. Yoeli and S. Rinon. Application of ternary algebra to the study of static hazards. *J. ACM*, 11(1):84–97, January 1964. `doi:10.1145/321203.321214`.