

# Karchmer-Wigderson Games for Hazard-free Computation

Christian Ikenmeyer\*, Balagopal Komarath†, Nitin Saurabh‡

July 2021

## 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 significant step towards establishing hazard-free computation as an independent missing link between Boolean complexity and monotone complexity.

**Keywords:** Hazard-free computation, monotone computation, Karchmer-Wigderson games, communication complexity, lower bounds, distributed computing

**ACM subject classification:** Theory of Comput. → Models of Computation → Concurrency  
Theory of Comput. → Comput. Compl. and Cryptogr. → Communication Complexity  
Theory of Comput. → Comput. Compl. and Cryptogr. → Circuit Complexity  
Hardware → Integrated Circuits → Logic Circuits → Combinational Circuits

**Acknowledgments:** Nitin Saurabh would like to thank Yuval Filmus for helpful discussions.

---

\*University of Liverpool, United Kingdom. email: christian.ikenmeyer@liverpool.ac.uk

†IIT Gandhinagar, India. email: bkomarath@rbgo.in

‡Technion-IIT, Haifa, Israel. email: nitinsau@cs.technion.ac.il. This project has received funding from the the European Union's Horizon 2020 research and innovation programme under grant agreement No 802020-ERC-HARMONIC.

# 1 Introduction

The study of the three-valued strong logic of indeterminacy dates back to Kleene ([Kle38, p. 153], [Kle52, §64]). It found numerous applications, for example in logic (see e.g. [Kör66], [Mal14]), in cybersecurity for information flow tracking at the gate level (see e.g. [TWM<sup>+</sup>09], [HOI<sup>+</sup>12], [BHT<sup>+</sup>17]), the design of real-world circuits that communicate between unsynchronized clock domains (see e.g. [FFL18], [FKLP17], [TFL17], [BLM20]), and in the study of hazards in Boolean circuits (see e.g. [Got49, Cal58, YR64, Eic65, Muk72, Muk83a, Muk83b, ND92, BS95, BEI01]). The languages in these areas is different, but the underlying three-valued 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 [Got49], 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  $u := \frac{1}{2}$  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  $\{0, u, 1\}$ . Hence a Boolean circuit  $C$  on  $n$  inputs<sup>1</sup> computes a function  $\{0, u, 1\}^n \rightarrow \{0, u, 1\}$  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*<sup>2</sup>  $\bar{f} : \{0, u, 1\}^n \rightarrow \{0, u, 1\}$  of a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ . It is defined as follows.

A binary string  $a \in \{0, 1\}^n$  is called a *resolution* of a ternary string  $\alpha \in \{0, u, 1\}^n$  if for all  $1 \leq i \leq n$  with  $\alpha_i \neq u$  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  $\alpha$  forms a subcube of  $\{0, 1\}^n$ . For a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and for an input  $\alpha \in \{0, u, 1\}^n$  we define the evaluation of the function  $\bar{f} : \{0, u, 1\}^n \rightarrow \{0, u, 1\}$  at  $\alpha$  via

$$\bar{f}(\alpha) := \begin{cases} 1 & \text{if for all resolutions } a \text{ of } \alpha \text{ we have } f(a) = 1 \\ 0 & \text{if for all resolutions } a \text{ of } \alpha \text{ we have } f(a) = 0 \\ u & \text{otherwise.} \end{cases} \quad (1.1)$$

A Boolean circuit  $C$  that computes a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is called *hazard-free* if for all  $\alpha \in \{0, u, 1\}^n$  we have  $C(\alpha) = \bar{f}(\alpha)$ . An  $\alpha$  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  $C(s, x_0, x_1) = \text{MUX}(s, x_0, x_1) = x_s$  for all  $(s, x_0, x_1) \in \{0, 1\}^3$ . We observe that  $C$  has a hazard at  $(u, 1, 1)$ , because  $C(u, 1, 1) = u \vee u = u$ , 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 [FFL18, Fig. 6a] and [IKL<sup>+</sup>19, Fig. 1b]).

Designing small hazard-free circuits for computing Boolean functions is a fundamental goal in electronic circuit design. Huffman [Huf57] 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 [IKL<sup>+</sup>19] formally defines the notion of hazard-free complexity and shows that for monotone functions the hazard-free complexity and the monotone complexity coincide. Fortunately,

<sup>1</sup>All circuits in our paper have a single output.

<sup>2</sup>The function  $\bar{f}$  is called the *hazard-free extension* of  $f$  (see [IKL<sup>+</sup>19]), or alternatively the *ternary extension* (see [MSB12]) or the *metastable closure* (see [FFL18]).

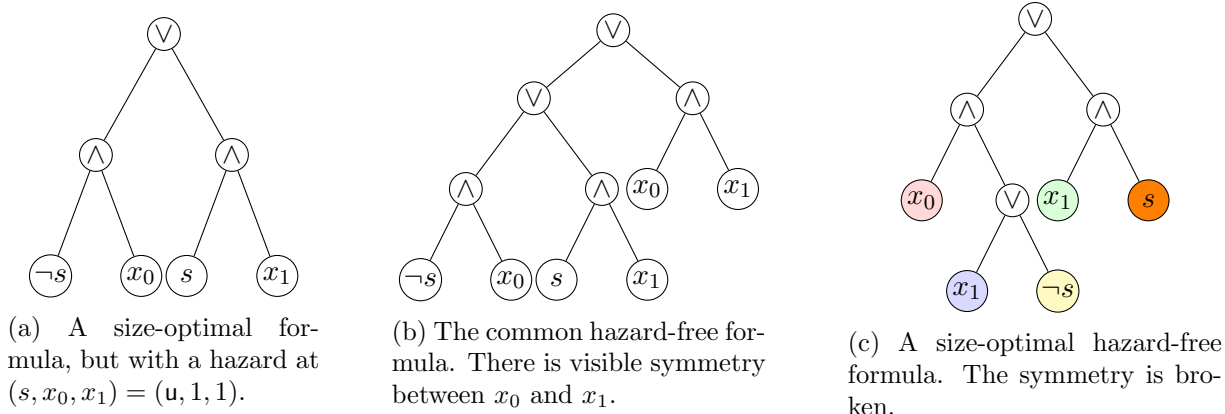


Figure 1: Different De Morgan formulas for  $\text{MUX}_1$

good lower bounds are known on the monotone complexity of monotone Boolean functions (see [Raz85a, Raz85b, And87, AB87, Raz87, Tar88, GS95, KW90, RW92, RM99, HR00, GP14, PR17]). A direct consequence of [IKL<sup>+</sup>19] 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. [Juk21] 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, [IKL<sup>+</sup>19] 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 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 3.1 with Theorem 5.12). 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 4.5) 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 4.10). 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 5.8 and 5.12) and the depth of hazard-free formulas of alternation depth<sup>3</sup> 2 (Theorem 6.5) of the multiplexer function  $\text{MUX}_n : \{0, 1\}^{n+2^n} \rightarrow \{0, 1\}$ , which is a (non-monotone)

<sup>3</sup>Alternation depth is one plus the maximum number of changes in the type of the gate in root-to-leaf paths.

Boolean function on  $n + 2^n$  input bits, defined via

$$\text{MUX}_n(s_1, \dots, s_n, x_{0,0,\dots,0}, x_{0,0,\dots,0,1}, x_{0,0,\dots,1,0}, \dots, x_{1,1,\dots,1}) := x_{s_1,\dots,s_n}.$$

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 ([TWM<sup>+</sup>09], [HOI<sup>+</sup>12], [BHT<sup>+</sup>17]), but also for designing real-world circuits, for example when a distributed system of agents with unsynchronized clock domains performs a parallel computation, see [FKLP17, FFL18, TFL17, LM16, BLM17, BLM18, BLM20]. 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 [Fis75, TNB96, BNT98]. 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 [FKLP17, LM16, BLM17, BLM18, BLM20]. We are not aware of any applications of limited negation circuits for designing real-world circuits.

## 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 [Huf57] construction gives only  $\text{size}^u(\text{MUX}_n) \leq 4^n + 2n3^{n-1}$ ) and an analysis of the hazard-free Karchmer-Wigderson game for the lower bound:

$$\text{size}^u(\text{MUX}_n) = 2 \cdot 3^n - 1.$$

We note that there are De Morgan formulas (with hazards) of size  $3 \cdot 2^n - 2$  computing  $\text{MUX}_n$  [Weg87], i.e.,  $\text{size}(\text{MUX}_n) \leq 3 \cdot 2^n - 2$ . Our upper bound construction is a recursive application of the improved implementation of  $\text{MUX}_1$  in Figure 1(c). To prove the lower bound we reduce the Karchmer-Wigderson game for  $\text{MUX}_n$  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  $\text{MUX}_n$  have monotone formulas of size at most  $(n+1)2^n$  (Proposition 3.1), 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<sup>4</sup>. 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  $\text{depth}^u(\text{MUX}_n) \geq \log_2(3)n \geq 1.58n$ . This lower bound separates the hazard-free circuit depth complexity and Boolean circuit depth complexity of  $\text{MUX}_n$ , because  $\text{depth}(\text{MUX}_n) \leq \text{depth}_2(\text{MUX}_n) \leq n + \log_2(n)$  (see Theorem 5.1, Section 3.5 in [Weg87]). Analogously to formula size, since all derivatives of  $\text{MUX}_n$  have monotone circuits of depth at most  $n + \log_2(n) + 1$  (Proposition 3.1), our separation breaks the monotone barrier.

In Section 6 we focus on the depth of hazard-free formulas of alternation depth 2 for  $\text{MUX}_n$ . 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 using a combination of a new protocol for the upper bound and a hazard-free Karchmer-Wigderson game for the lower bound:

$$\text{depth}_2^u(\text{MUX}_n) = 2n + 2.$$

Hence, in this restricted model we can precisely determine the depth complexity. 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 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  $\text{MUX}_n$ , because we show in Theorem 5.14 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 5.8, and strictly lower than the depth that can be achieved by any formula of alternation depth 2. Moreover, the size of this formula is only a factor of  $\frac{5}{4}$  more than the optimal size, so we think that this is an excellent size-depth trade-off. 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  $\text{MUX}_{n-1}$ , 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 5.14 and 6.5. The upper bound in Theorem 5.8 can also be proved without using the game (see Remark 5.7).

## 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 [Sha49]. For Boolean circuits and formulas, this question has been very satisfactorily answered. It is known that any

---

<sup>4</sup>Note that breaking the monotone barrier can also be achieved using Khrapchenko's method for the parity function [Khr71], which was interpreted as a Karchmer-Wigderson game in [KW90], 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.

$n$ -bit Boolean function has circuits of size  $(1 + o(1))2^n/n$  [Lup58] and almost all Boolean functions require circuits of size  $(1 + o(1))2^n/n$  [RS42, Sha49, Lut92, FM05]. For Boolean formulas, the lower bound is  $2^n/\log(n)$  [RS42, Sha49], almost matched by the upper bound  $(1 + o(1))\frac{2^n}{\log n}$  [Lup60].

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., [Juk21, Section 7]), thus matching Lupanov’s upper bound [Lup58] up to constants. Since hazard-free circuits are also Boolean circuits, Riordon and Shannon’s lower bound of  $2^n/n$  for almost all functions continues to hold for hazard-free circuits.

For hazard-free formulas, this question is still open. Huffman [Huf57] 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 [CM78] 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:  $f(x_1, \dots, x_n) = \text{MUX}_n(x_1, \dots, x_n, f(0, 0, \dots, 0), \dots, f(1, 1, \dots, 1))$ . This implementation of  $f$  is hazard-free if the implementation of  $\text{MUX}_n$  is hazard-free. Therefore, any hazard-free formula upper bound for  $\text{MUX}_n$  gives an upper bound for the hazard-free formula complexity of *all*  $n$ -bit Boolean functions. Theorem 5.8 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 5.12 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 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  $\text{size}(F)$ , is defined to be the number of leaves in it<sup>5</sup>. The *depth* of a formula  $F$ , denoted  $\text{depth}(F)$ , is defined to be the length of the longest root-to-leaf path in  $F$ . For a Boolean function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ , we denote the minimal size of a De Morgan formula computing  $f$  by  $\text{size}(f)$  and the minimal depth of a formula computing  $f$  by  $\text{depth}(f)$ . Similarly, in the hazard-free setting, let  $\text{size}^u(f)$  and  $\text{depth}^u(f)$  denote the minimal size and minimal depth of a hazard-free De Morgan formula computing  $f$ , respectively. For a monotone function  $f$  let  $\text{size}^+(f)$  and  $\text{depth}^+(f)$  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  $\text{size}_d^u(f)$  and  $\text{depth}_d^u(f)$ , respectively.

**Implicants and Implicates.** For a Boolean function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  the preimage of a value  $c \in \{0, 1\}$  is denoted by  $f^{-1}(c)$ . For the hazard-free extension  $\bar{f}: \{0, u, 1\}^n \rightarrow \{0, u, 1\}$  the preimage of  $\gamma \in \{0, u, 1\}$  is denoted by  $\bar{f}^{-1}(\gamma)$ . Elements  $\alpha \in \bar{f}^{-1}(1)$  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 maximal with respect to the number of  $u$ s. Elements  $\alpha \in \bar{f}^{-1}(0)$  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 maximal with respect to the number of  $u$ s. We occasionally identify an implicant  $\alpha$  with the Boolean function that is 1 exactly on the hypercube of resolutions of  $\alpha$ , and an implicate  $\beta$  with the Boolean function that is 0 exactly on the hypercube of resolutions of  $\beta$ . An implicant or implicate  $\alpha$  can also be identified with the set of literals  $\{x_i \mid \alpha_i = 1\} \cup \{\neg x_i \mid \alpha_i = 0\}$ .

**Communication.** We assume familiarity with the basic definitions of communication complexity (see, e.g., [KN96, RY20]). Let  $K: A \times B \rightarrow 2^O$  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  $\Pi$  be a deterministic communication protocol solving  $K$ . Then the *communication cost* of  $\Pi$ , denoted  $\text{CC}(\Pi)$ , is defined to be the maximum number of bits exchanged on any pair of inputs  $(\alpha, \beta)$  when following  $\Pi$ . Let  $\text{CC}(K)$  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  $\Pi$  by  $\text{monorect}(\Pi)$  and the minimum number of leaves in a protocol solving  $K$  by  $\text{monorect}(K)$ .

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  $(\alpha, \beta)$ -th entry of  $M_K$  is defined to be  $K(\alpha, \beta)$ . The leaves of a protocol  $\Pi$  solving  $K$  partitions the communication matrix  $M_K$  into  $\text{monorect}(\Pi)$  many monochromatic 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.

---

<sup>5</sup>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.

### 3 Hazard-Derivatives and the Monotone Barrier

Let  $f$  be a Boolean function on  $n$  variables. Its *hazard derivative* is a Boolean function on  $2n$  variables denoted  $\text{df}(x; y)$  that evaluates to 1 if and only if  $\bar{f}(x \oplus u \cdot y) = u$ , i.e., there are two resolutions of  $x \oplus u \cdot y$ , say  $a$  and  $b$ , such that  $f(a) = 0$  and  $f(b) = 1$ . In other words, the function  $f$  is *not* constant in the subcube of all resolutions of  $x \oplus u \cdot y$ . Notice that for a fixed value  $a$  of  $x$ , the restriction of  $\text{df}$  to  $y$ , i.e., the function  $\text{df}(a; y) : \{0, 1\}^n \rightarrow \{0, 1\}$ , is a monotone function.

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  $\text{df}(a; y)$  that is no larger in size than  $C$ , see [IKL<sup>+</sup>19, 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  $\text{df}(a; y)$  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.

The best known construction for hazard-free De Morgan formulas for  $\text{MUX}_n$  has size  $2 \cdot 3^n - 1$  (See Theorem 5.8). Can we use derivatives to prove that this is optimal? No. We show that *all* derivatives of  $\text{MUX}_n$  have monotone De Morgan formulas of size at most  $(n + 1)2^n$ . This is an instance of the *monotone barrier*.

**3.1 Proposition.** *Fix any  $a \in \{0, 1\}^{n+2^n}$ . The function  $\text{dMUX}_n(a; y) : \{0, 1\}^{n+2^n} \rightarrow \{0, 1\}$  has monotone De Morgan formulas of size at most  $(n + 1)2^n$ .*

*Proof.* Let  $(s^a, x^a) \in \{0, 1\}^n \times \{0, 1\}^{2^n}$  denote the selector and data bits of  $a$ . Analogously, we partition the sequence of Boolean variables  $y$  into  $n$  variables  $s^y$  and  $2^n$  variables  $x^y$ . For  $b \in \{0, 1\}^n$ , we define the formula  $D_b(y) = \bigwedge_{i: b_i \neq s_i^a} s_i^y$ . Note that  $D_b(y) = 1$  if and only if  $b$  is a resolution of  $s^a \oplus u \cdot s^y$ . Suppose  $\text{MUX}_n(a) = 1$  (The other case is symmetric). We now define  $F_b(y)$ . If  $x_b^a = 1$ , we define  $F_b(y) := D_b(y) \wedge x_b^y$ . If  $x_b^a = 0$ , we define  $F_b(y) := D_b(y)$ . We have  $F_b(y) = 1$  if and only if  $[b$  is a resolution of  $s^a \oplus u \cdot s^y$  and  $x_b^a \oplus u \cdot x_b^y \neq 1]$ . Hence  $F_b(y) = 1$  if and only if there exists  $b' \in \{0, 1\}^{2^n}$  such that  $(b, b')$  is a resolution of  $a \oplus u \cdot y$  with  $\text{MUX}(b, b') = 0$ . We claim that the formula  $\bigvee_{b \in \{0, 1\}^n} F_b(y)$  is an implementation of  $\text{dMUX}_n(a; y)$ . This is because  $\text{dMUX}_n(a; y) = 1$  exactly when there is a resolution of  $a \oplus u \cdot y$  that evaluates to zero under  $\text{MUX}_n$ . Finally, this formula is a monotone De Morgan formula of size at most  $(n + 1)2^n$ , because each  $F_b(y)$  has size at most  $n + 1$ .  $\square$

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  $\text{MUX}_n$ . We now proceed to develop a framework that will allow us to prove that  $2 \cdot 3^n - 1$  is the optimal size for  $\text{MUX}_n$ . 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 [KW90].

**4.1 Definition** ([KW90]). *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. The Karchmer-Wigderson game of  $f$ , denoted  $\text{KW}_f$ , 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  $a_i \neq b_i$ .*



	000	00u	001	u00	100	1u0	110
010	2	2	2,3	2	1,2	1	1
01u	2	2	2	2	1,2	1	1
011	2,3	2	2	2,3	1,2,3	1,3	1,3
u11	2,3	2	2	2,3	2,3	3	3
101	1,3	1	1	3	3	3	2,3
1u1	1,3	1	1	3	3	3	3
111	1,2,3	1,2	1,2	2,3	2,3	3	3

	000	00u	001	u00	100	1u0	110
010	2	2	2	2	1	1	1
01u	2	2	2	2	1	1	1
011	2	2	2	2	1	1	1
u11	2	2	2	2	3	3	3
101	1	1	1	3	3	3	3
1u1	1	1	1	3	3	3	3
111	1	1	1	3	3	3	3

(a) The communication matrix  $M_{\text{KW}_{\text{MUX}_1}^u}$ .

(b) The monochromatic partition of  $M_{\text{KW}_{\text{MUX}_1}^u}$ . It is a result of applying the construction from Lemma 7.1 to Figure 1(c).

Figure 2: The communication matrix  $M_{\text{KW}_{\text{MUX}_1}^u}$  and a monochromatic partition.

They also gave the following monotone version of the game.

**4.2 Definition** ([KW90]). *Let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. The monotone Karchmer-Wigderson game of  $f$ , denoted  $\text{KW}_f^+$ , 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 [KW90] showed that the communication complexity of the  $\text{KW}_f$  game (resp.,  $\text{KW}_f^+$  game) characterizes the size and depth complexity of De Morgan formulas (resp., monotone formulas).

**4.3 Theorem** ([KW90]). *Let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. Then,*

$$\text{depth}(f) = \text{CC}(\text{KW}_f), \quad \text{and} \quad \text{size}(f) = \text{monorect}(\text{KW}_f).$$

*Let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. Then,*

$$\text{depth}^+(f) = \text{CC}(\text{KW}_f^+), \quad \text{and} \quad \text{size}^+(f) = \text{monorect}(\text{KW}_f^+).$$

We now extend the Karchmer-Wigderson games to the hazard-free setting. For a Boolean function  $f: \{0, 1\}^n \rightarrow \{0, 1\}$ , recall from (1.1) that  $\bar{f}: \{0, u, 1\}^n \rightarrow \{0, u, 1\}$  is the hazard-free extension of  $f$ .

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

An example of a communication matrix for this game is shown in Figure 2. Observe that, for all  $(\alpha, \beta) \in \bar{f}^{-1}(1) \times \bar{f}^{-1}(0)$ , there exists an  $i \in [n]$  such that  $\alpha_i \neq \beta_i$ ,  $\alpha_i \neq u$  and  $\beta_i \neq u$ <sup>6</sup>, which implies that all cells in the communication matrix are nonempty.

**4.5 Remark.** *Note that the wordy condition “ $\alpha_i \neq \beta_i$  and  $\alpha_i \neq u$  and  $\beta_i \neq u$ ” 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*

<sup>6</sup>Assume the contrary. Then  $\alpha$  and  $\beta$  must have some resolution in common, which we call  $a$ . Hence  $f(a) = 1$  and  $f(a) = 0$ , which is a contradiction.

game, see Def. 4.1, whereas we will show in Theorem 4.10 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  $\text{KW}_f^u$  we characterize the complexity of hazard-free De Morgan formulas for Boolean functions.

**4.6 Theorem.** *Let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function. Then,*

$$\text{depth}^u(f) = \text{CC}(\text{KW}_f^u), \quad \text{and} \quad \text{size}^u(f) = \text{monorect}(\text{KW}_f^u).$$

The proof is a natural generalization of the proof of Theorem 4.3 and is provided in Section 7.

**4.7 Remark.** *We remark that a variant of the game  $\text{KW}_f^u$  has been considered in prior works [Hås98, FMT21]. In this variant, the inputs to Alice and Bob remains the same but the goal is different. More formally, Alice gets  $\alpha \in \bar{f}^{-1}(1)$ , Bob gets  $\beta \in \bar{f}^{-1}(0)$  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 4.4), where we forbid such answers by requiring that  $\alpha_i \neq u$  and  $\beta_i \neq u$ .*

#### 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 4.10.

**4.8 Theorem.** *For any function  $f$ , the complexity (works for size and also for depth) of the game  $\text{KW}_f^u$  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  $\alpha$  and an arbitrary implicate  $\beta$ , Alice can choose a prime implicant  $\alpha'$  that is obtained by flipping some stable bits in  $\alpha$  to  $u$  and Bob can choose a prime implicate  $\beta'$  that is obtained by flipping some stable bits in  $\beta$  to  $u$ , and play the restricted game on the input  $(\alpha', \beta')$ . 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.  $\square$

For example, consider the communication matrix of  $\text{KW}_{\text{MUX}_1}^u$  given below, where we restricted the rows and columns to prime implicants and prime implicates. It is a submatrix of Figure 2(a).

$$\begin{array}{ccc} & 00u & u00 & 1u0 \\ \begin{array}{l} 01u \\ u11 \\ 1u1 \end{array} & \begin{pmatrix} 2 & 2 & 1 \\ 2 & 2,3 & 3 \\ 1 & 3 & 3 \end{pmatrix} & & \end{array} \quad (4.9)$$

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 4.8 in the monotone world: in Definition 4.2, 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.

**4.10 Theorem.** *Let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone function. Then, the games  $\text{KW}_f^u$  and  $\text{KW}_f^+$  are equivalent.*

*Proof.* First, we show that the complexity of  $\text{KW}_f^u$  is at most that of  $\text{KW}_f^+$ . Using Theorem 4.8, 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  $\text{KW}_f^+$ . Notice that by Definition 4.2, 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  $\bar{f}^{-1}(1)$  and Bob still has an input in  $\bar{f}^{-1}(0)$ . Now, the output of the game  $\text{KW}_f^u$  on these new inputs will also be a valid output for the game  $\text{KW}_f^+$  since all stable bits in Alice's input are 1 and all stable bits in Bob's input are 0.  $\square$

## 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  $\text{MUX}_1$  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  $\text{MUX}_n$ . Following the discussion in Subsection 1-2, this upper bound also applies to all  $n$ -bit Boolean functions and improves upon Huffman's construction [Huf57]. 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  $\text{MUX}_n: \{0, 1\}^{n+2^n} \rightarrow \{0, 1\}$  is a Boolean function on  $n+2^n$  variables defined as

$$\text{MUX}_n(s_1, \dots, s_n, x_0, x_1, \dots, x_{2^n-1}) = x_{\text{bin}(s_1, \dots, s_n)}, \quad (5.1)$$

where  $\text{bin}(s_1, \dots, s_n)$  is the natural number represented by the binary number  $s_1 s_2 \dots s_n$ . We will be studying the communication matrix  $M_{\text{KW}_{\text{MUX}_n}^u}$  of the hazard-free game  $\text{KW}_{\text{MUX}_n}^u$ . Following Theorem 4.8, we will restrict our attention to the submatrix given by prime implicants and implicates of  $\text{MUX}_n$ . The following proposition gives the structure of the prime implicants and prime implicates.

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

*Proof.* For  $\alpha \in \{0, u, 1\}^n$ , consider the string  $\alpha' \in \{u, 1\}^{2^n}$  that has 1s at positions indexed by the resolutions of  $\alpha$  and that has  $u$ s elsewhere. We have  $\text{MUX}_n(\alpha\alpha') = 1$  showing that  $\alpha\alpha'$  is an implicant. We now show it is a prime implicant. If any 1s in  $\alpha'$  are made a  $u$ , then the output becomes a  $u$ , because a resolution of  $\alpha$  now indexes into a  $u$ . If any Boolean value in  $\alpha$  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  $\alpha$  because the  $\alpha'$  part is minimal and all 1s in it are necessary. The argument for prime implicates is symmetric.  $\square$

The following proposition states the inductive structure of communication matrices of  $\text{MUX}_n$ .

	000uuu	0u00uu	01u0uu	u00u0u	uu0000	u1u0u0	10uu0u	1uuu00	11uuu0
001uuu	$x_0$	$x_0$	$s_2$	$x_0$	$x_0$	$s_2$	$s_1$	$s_1$	$s_1, s_2$
0u11uu	$x_0$	$x_0, x_1$	$x_1$	$x_0$	$x_0, x_1$	$x_1$	$s_1$	$s_1$	$s_1$
01u1uu	$s_2$	$x_1$	$x_1$	$s_2$	$x_1$	$x_1$	$s_1, s_2$	$s_1$	$s_1$
u01u1u	$x_0$	$x_0$	$s_2$	$x_0, x_2$	$x_0, x_2$	$s_2$	$x_2$	$x_2$	$s_2$
uu1111	$x_0$	$x_0, x_1$	$x_1$	$x_0, x_2$	$x_0, x_1, x_2, x_3$	$x_1, x_3$	$x_2$	$x_2, x_3$	$x_3$
u1u1u1	$s_2$	$x_1$	$x_1$	$s_2$	$x_1, x_3$	$x_1, x_3$	$s_2$	$x_3$	$x_3$
10uu1u	$s_1$	$s_1$	$s_1, s_2$	$x_2$	$x_2$	$s_2$	$x_2$	$x_2$	$s_2$
1uuu11	$s_1$	$s_1$	$s_1$	$x_2$	$x_2, x_3$	$x_3$	$x_2$	$x_2, x_3$	$x_3$
11uuu1	$s_1, s_2$	$s_1$	$s_1$	$s_2$	$x_3$	$x_3$	$s_2$	$x_3$	$x_3$

Figure 3: The communication matrix for  $KW_{MUX_2}^u$ .

**5.3 Proposition.** *The communication matrix of  $KW_{MUX_n}^u$ , when restricted to prime implicants and prime implicates, has the following inductive structure:*

- For  $n = 1$ ,

$$M_{KW_{MUX_1}^u} = \begin{matrix} & & 00u & u00 & 1u0 \\ \begin{matrix} 01u \\ u11 \\ 1u1 \end{matrix} & \begin{pmatrix} x_0 & x_0 & s \\ x_0 & x_0, x_1 & x_1 \\ s & x_1 & x_1 \end{pmatrix} \end{matrix}.$$

- For  $n \geq 2$ ,

$$M_{KW_{MUX_n}^u} = \begin{matrix} & & 0 & u & 1 \\ \begin{matrix} 0 \\ u \\ 1 \end{matrix} & \begin{pmatrix} M_0 & M_0 & s_1 \\ M_0 & M_0 \cup M_1 & M_1 \\ s_1 & M_1 & M_1 \end{pmatrix} \end{matrix},$$

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

$$F_0 = MUX_{n-1}(s_2, \dots, s_{n-1}, x_0, \dots, x_{2^{n-1}-1}) = MUX_n(0, s_2, \dots, s_n, x_0, x_1, \dots, x_{2^n-1}),$$

$$F_1 = MUX_{n-1}(s_2, \dots, s_n, x_{2^{n-1}}, \dots, x_{2^n-1}) = MUX_n(1, s_2, \dots, s_n, x_0, x_1, \dots, x_{2^n-1}),$$

and matrices  $M_0 := M_{KW_{F_0}^u}$ ,  $M_1 := M_{KW_{F_1}^u}$ ,  $s_1$  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  $s_1, \dots, s_n, x_0, x_1, \dots, x_{2^n-1}$  for better readability (cp. (4.9)).

See Figure 3 for the example of  $M_{KW_{MUX_2}^u}$ .

*Proof of Proposition 5.3.* For  $n = 1$  the proof follows by inspection and when  $n \geq 2$  it follows from the following recursive decomposition:  $MUX_n = MUX_1(s_1, F_0, F_1)$ .  $\square$

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

**5.4 Proposition.** *Let  $M$  be a communication matrix such that  $M = \begin{pmatrix} A & A \\ A & A \end{pmatrix}$  or  $M = (A \ A)$ .*

Then,

$$\text{CC}(M) = \text{CC}(A) \quad \text{and} \quad \text{monorect}(M) = \text{monorect}(A).$$

*Proof.* Since  $A$  is a submatrix of  $M$ , the following inequalities are self-evident

$$\text{CC}(M) \geq \text{CC}(A) \quad \text{and} \quad \text{monorect}(M) \geq \text{monorect}(A).$$

For the other direction we treat only the case  $M = \begin{pmatrix} A & A \\ A & A \end{pmatrix}$ , as the other case follows because

$\begin{pmatrix} A & A \\ A & A \end{pmatrix}$  is a submatrix of  $\begin{pmatrix} A & A \\ A & A \end{pmatrix}$ . We consider a protocol  $\Pi$  of  $A$ . Using  $\Pi$  we give a protocol  $\Pi'$  for  $M$  such that  $\text{CC}(\Pi') = \text{CC}(\Pi)$  and  $\text{monorect}(\Pi') = \text{monorect}(\Pi)$ .

Without loss of generality, assume Alice is the first player to start in  $\Pi$  and she sends a bit to indicate whether her input lies in the set of rows  $R_1$  or  $R_2$  such that the disjoint union  $R_1 \uplus R_2$  is the set of all rows in  $A$ . Now let  $R'_1$  be the union of rows  $R_1$  from each copy of  $A$  within  $M$  and  $R'_2$  be the union of rows  $R_2$  from each copy of  $A$  within  $M$ . Clearly,  $R'_1 \uplus R'_2$  is the set of all rows in  $M$ . Then in  $\Pi'$  too, Alice will start by sending a bit to indicate whether her input lies in  $R'_1$  or  $R'_2$ . Upon receiving the message from Alice, Bob now communicates using completely analogous adjustments to the protocol  $\Pi$ . The two players proceed in this way and keep making these adjustments until they reach the end of  $\Pi$ . From the protocol it follows that at the end of  $\Pi'$  each rectangle in  $M$  is a union of the same rectangles from each copy of  $A$ , and thus monochromatic. Clearly, we also have  $\text{CC}(\Pi') = \text{CC}(\Pi)$  and  $\text{monorect}(\Pi') = \text{monorect}(\Pi)$ . Since  $\Pi$  is an arbitrary protocol for  $A$ , we obtain  $\text{CC}(M) \leq \text{CC}(A)$  and  $\text{monorect}(M) \leq \text{monorect}(A)$ .  $\square$

## 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 4.6, we begin with finding optimal formulas for  $\text{MUX}_1$ .

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

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

$$\begin{array}{c} 00u \quad u00 \quad 1u0 \\ 01u \begin{pmatrix} x_0 & x_0 & s \\ u11 \begin{pmatrix} x_0 & x_0, x_1 & x_1 \\ 1u1 \begin{pmatrix} s & x_1 & x_1 \end{pmatrix} \end{pmatrix} \end{pmatrix} \end{array}.$$

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

$$\begin{array}{c} 00u \quad u00 \quad 1u0 \\ 01u \begin{pmatrix} x_0 & x_0 & s \\ u11 \begin{pmatrix} x_0 & x_0 & x_1 \\ 1u1 \begin{pmatrix} s & x_1 & x_1 \end{pmatrix} \end{pmatrix} \end{pmatrix} \end{array}. \quad (5.6)$$

Using Lemma 7.2 with the above protocol, we obtain the hazard-free formula for  $\text{MUX}_1$  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 5.12, the general case of  $\text{MUX}_n$ .  $\square$

**5.7 Remark.** *To demystify the construction in Figure 1c we note that it is simply the hazard-free DNF of  $\text{MUX}_1$ , 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  $\text{MUX}_n$  we obtain the following upper bound.

**5.8 Theorem.** *The multiplexer function  $\text{MUX}_n$  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  $\text{MUX}_1$  is given by Proposition 5.5. Recall that we can write  $\text{MUX}_n(s_1, \dots, s_n, x_0, \dots, x_{2^n-1})$  recursively as the formula  $F = \text{MUX}_1(s_1, F_0, F_1)$ , where

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

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  $\text{MUX}_1$ , given in Figure 1c, to implement  $F$  yields a formula of size  $2 \cdot 3^n - 1$  and depth  $3n$  for  $\text{MUX}_n$ .

It remains to prove that the constructed formula  $F$  is hazard-free. Using Lemma 7.1, it suffices to show that the protocol using  $F$  correctly solves the hazard-free KW-game  $\text{KW}_{\text{MUX}_n}^u$ . In other words, the communication matrix of  $\text{KW}_{\text{MUX}_n}^u$  is partitioned into monochromatic rectangles by the protocol given by  $F$ . From the monochromatic partition of  $\text{KW}_{\text{MUX}_1}^u$  in (5.6), the structure of the communication matrix of  $\text{KW}_{\text{MUX}_n}^u$  (Proposition 5.3) and Proposition 5.4, we obtain the following monochromatic partition of  $M_{\text{KW}_{\text{MUX}_n}^u}$  as a block matrix, where  $M_i := M_{\text{KW}_{F_i}^u}$  for  $i \in \{0, 1\}$ , and  $s_1$  stands for a block matrix of all entries  $s_1$ :

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

where the row (resp., column) labeled  $\gamma \in \{0, u, 1\}$  represents the set of prime implicants (resp., implicates) with  $s_1 = \gamma$ . Using Proposition 5.4, we can partition the whole  $\begin{pmatrix} M_0 & M_0 \\ M_0 & M_0 \end{pmatrix}$  at the same cost for partitioning a single  $M_0$ . The same is true for  $M_1$  with the block  $\begin{pmatrix} M_1 & M_1 \end{pmatrix}$ . This gives a monochromatic partition of  $M_{\text{KW}_{\text{MUX}_n}^u}$  with size  $3 \cdot (2 \cdot 3^{n-1} - 1) + 2 = 2 \cdot 3^n - 1$  and depth  $3(n-1) + 3 = 3n$  as claimed.  $\square$

We now prove that the above construction for  $\text{MUX}_n$  is optimal with respect to size. For this purpose, we study the communication problem associated with the following *subcube intersection* function,

$$\text{subcube-intersect}_n: \{0, u, 1\}^n \times \{0, u, 1\}^n \rightarrow \{0, 1\},$$

where  $\text{subcube-intersect}_n(\alpha, \beta) = 1$  iff the subcubes defined by  $\alpha$  and  $\beta$  in  $\{0, 1\}^n$  intersect, i.e., if  $\alpha$  and  $\beta$  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  $\{0, u, 1\}$  (and hence in particular is not the hazard-free extension of any Boolean function), even for  $n = 1$ , because  $\text{subcube-intersect}_1(u, u) = 1$ , but  $\text{subcube-intersect}_1(0, 1) = 0$ <sup>7</sup>. Let us see how the subcube intersection problem helps in capturing the complexity of the hazard-free game  $\text{KW}_{\text{MUX}_n}^u$ .

<sup>7</sup>In any circuit implementation, if  $C(\alpha) = 1$ , then for all resolutions  $a$  of  $\alpha$  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 ( $u \sqsubseteq 0$ ,  $u \sqsubseteq 1$ ,  $0$  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.

**5.9 Lemma.** *The subcube-intersect<sub>n</sub> communication problem reduces to the communication problem KW<sub>MUX<sub>n</sub></sub><sup>u</sup> with no extra cost. That is, a monochromatic partition of the communication matrix of KW<sub>MUX<sub>n</sub></sub><sup>u</sup> is also a monochromatic partition for the communication matrix of subcube-intersect<sub>n</sub>.*

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

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

Now they can solve the game KW<sub>MUX<sub>n</sub></sub><sup>u</sup> on inputs  $\alpha\alpha'$  and  $\beta\beta'$ . Observe that if the subcubes  $\alpha$  and  $\beta$  intersect then answers to KW<sub>MUX<sub>n</sub></sub><sup>u</sup> lie in the set of data variables  $\{x_0, \dots, x_{2^n-1}\}$ , otherwise they lie in the set of selector variables  $\{s_1, \dots, s_n\}$ . Therefore, from the answers to the KW<sub>MUX<sub>n</sub></sub><sup>u</sup> game they can deduce whether the subcubes intersect or not, again without communication.  $\square$

Using the rank lower bound technique of [MS82] (See also [KN96, Lemma 1.28] and the discussion following the lemma.), we know that

$$\text{monorect}(\text{subcube-intersect}_n) \geq 2 \cdot \text{rank}(M_{\text{subcube-intersect}_n}) - 1, \quad (5.10)$$

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

**5.11 Lemma.** *The communication matrix of subcube-intersect<sub>n</sub> is of full rank. That is, the rank of  $M_{\text{subcube-intersect}_n}$  equals  $3^n$  for all  $n \geq 1$ .*

This immediately implies our size lower bound:

**5.12 Theorem.** *Any hazard-free formula for MUX<sub>n</sub> requires  $2 \cdot 3^n - 1$  leaves for all  $n \geq 1$ .*

*Proof.* Using Theorem 4.6, it is sufficient to show that the communication matrix of KW<sub>MUX<sub>n</sub></sub><sup>u</sup> requires  $2 \cdot 3^n - 1$  monochromatic rectangles, i.e.,  $\text{monorect}(\text{KW}_{\text{MUX}_n}^u) \geq 2 \cdot 3^n - 1$ . This is readily checked:

$$\begin{aligned} \text{monorect}(\text{KW}_{\text{MUX}_n}^u) &\stackrel{\text{Lem. 5.9}}{\geq} \text{monorect}(\text{subcube-intersect}_n) \\ &\stackrel{(5.10)}{\geq} 2 \cdot \text{rank}(M_{\text{subcube-intersect}_n}) - 1 \stackrel{\text{Lem. 5.11}}{=} 2 \cdot 3^n - 1. \quad \square \end{aligned}$$

It now remains to prove Lemma 5.11.

*Proof of Lemma 5.11.* We prove it by induction on  $n$ . For the base case,  $n = 1$  and the communication matrix  $M_{\text{subcube-intersect}_1}$  is as follows:

$$\begin{array}{ccc} & 0 & u & 1 \\ 0 & \left( \begin{array}{ccc} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{array} \right) \\ u & & & \\ 1 & & & \end{array}$$

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

$$\begin{array}{ccc} & 0 & u & 1 \\ 0 & \left( \begin{array}{ccc} M_{\text{subcube-intersect}_{n-1}} & M_{\text{subcube-intersect}_{n-1}} & 0 \\ M_{\text{subcube-intersect}_{n-1}} & M_{\text{subcube-intersect}_{n-1}} & M_{\text{subcube-intersect}_{n-1}} \\ 0 & M_{\text{subcube-intersect}_{n-1}} & M_{\text{subcube-intersect}_{n-1}} \end{array} \right), \\ u & & & \\ 1 & & & \end{array}$$

where the row labeled  $\gamma \in \{0, u, 1\}$  represents the set of rows labeled with  $\alpha \in \{0, u, 1\}^n$  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:

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

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

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

Now using the induction hypothesis completes the proof.  $\square$

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

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

### 5-3 Formulas of improved depth

The lower bound from Corollary 5.13 on hazard-free formula depth is at least  $1 + (\log_2 3) \cdot n$  for large  $n$ . However, our construction in Theorem 5.8 gives an upper bound of  $3n$ . We now give an improved construction (Theorem 5.14) with respect to depth while increasing the size by a factor of  $\frac{5}{4}$ . In contrast, the depth-optimal version of Huffman's construction (Proposition 6.2) is larger than the optimal size hazard-free formula by a multiplicative factor that is exponential in  $n$ .

**5.14 Theorem.** *The multiplexer function  $\text{MUX}_n$  has hazard-free formulas of depth  $2n + 1$  and size at most  $\frac{5}{2} \cdot 3^n - \frac{3}{2}$  for all  $n \geq 1$ .*

From Theorem 4.6 we know it is sufficient to give a protocol  $\Pi$  solving the hazard-free Karchmer-Wigderson game of  $\text{MUX}_n$  such that  $\text{CC}(\Pi) \leq 2n + 1$  and  $\text{monorect}(\Pi) \leq \frac{5}{2} \cdot 3^n - \frac{3}{2}$ . We consider a monochromatic extension of  $\text{KW}_{\text{MUX}_n}^u$ . We extend the communication matrix of  $\text{KW}_{\text{MUX}_n}^u$  as follows to define the extended version  $\text{e-KW}_{\text{MUX}_n}^u$ :

$$\text{prime implicants} \begin{pmatrix} \text{prime implicates} \\ M_{\text{KW}_{\text{MUX}_n}^u} \\ \hline 0 \end{pmatrix}. \quad (5.15)$$

In other words, the communication matrix of the extended version  $\text{e-KW}_{\text{MUX}_n}^u$  is obtained by adding a block of rectangle with all 0s to either the set of rows of  $\text{KW}_{\text{MUX}_n}^u$  (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  $\text{KW}_{\text{MUX}_n}^u$ . Observe that 0 doesn't appear in the communication matrix of  $\text{KW}_{\text{MUX}_n}^u$ . However we could have used any other number that doesn't appear in  $\text{KW}_{\text{MUX}_n}^u$ . We will denote all such extensions by  $\text{e-KW}_{\text{MUX}_n}^u$ . The following matrix is an



example of  $e\text{-KW}_{\text{MUX}_2}^u$ .

	000uuu	0u00uu	01u0uu	u00u0u	uu0000	u1u0u0	10uu0u	1uuu00	11uuu0
001uuu	$x_0$	$x_0$	$s_2$	$x_0$	$x_0$	$s_2$	$s_1$	$s_1$	$s_1, s_2$
0u11uu	$x_0$	$x_0, x_1$	$x_1$	$x_0$	$x_0, x_1$	$x_1$	$s_1$	$s_1$	$s_1$
01u1uu	$s_2$	$x_1$	$x_1$	$s_2$	$x_1$	$x_1$	$s_1, s_2$	$s_1$	$s_1$
u01ulu	$x_0$	$x_0$	$s_2$	$x_0, x_2$	$x_0, x_2$	$s_2$	$x_2$	$x_2$	$s_2$
uu1111	$x_0$	$x_0, x_1$	$x_1$	$x_0, x_2$	$x_0, x_1, x_2, x_3$	$x_1, x_3$	$x_2$	$x_2, x_3$	$x_3$
u1ulu1	$s_2$	$x_1$	$x_1$	$s_2$	$x_1, x_3$	$x_1, x_3$	$s_2$	$x_3$	$x_3$
10uu1u	$s_1$	$s_1$	$s_1, s_2$	$x_2$	$x_2$	$s_2$	$x_2$	$x_2$	$s_2$
1uuu11	$s_1$	$s_1$	$s_1$	$x_2$	$x_2, x_3$	$x_3$	$x_2$	$x_2, x_3$	$x_3$
11uuu1	$s_1, s_2$	$s_1$	$s_1$	$s_2$	$x_3$	$x_3$	$s_2$	$x_3$	$x_3$
	0	0	0	0	0	0	0	0	0

Clearly the following proposition holds.

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

*Proof.* Follows from the definition (5.15) of  $e\text{-KW}_{\text{MUX}_n}^u$ .  $\square$

Therefore, to prove the depth bound in Theorem 5.14 we will give a protocol for  $e\text{-KW}_{\text{MUX}_n}^u$  with communication cost at most  $2n + 1$ .

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

*Proof.* From Proposition 5.3 we know that the communication matrix of  $\text{KW}_{\text{MUX}_n}^u$  looks as follows

$$\begin{matrix} & 0 & u & 1 \\ \begin{matrix} 0 \\ u \\ 1 \end{matrix} & \begin{pmatrix} M_0 & M_0 & s_1 \\ M_0 & M_0 \cup M_1 & M_1 \\ s_1 & M_1 & M_1 \end{pmatrix}, \end{matrix}$$

where we define the formulas  $F_0 = \text{MUX}_{n-1}(s_2, \dots, s_n, x_0, \dots, x_{2^{n-1}-1})$  and  $F_1 = \text{MUX}_{n-1}(s_2, \dots, s_n, x_{2^{n-1}}, \dots, x_{2^n-1})$  and matrices  $M_i := M_{\text{KW}_{E_i}^u}$  for  $i \in \{0, 1\}$ . Therefore, the matrix of extended version  $e\text{-KW}_{\text{MUX}_n}^u$  looks as follows

$$\begin{matrix} & 0 & u & 1 \\ \begin{matrix} 0 \\ u \\ 1 \end{matrix} & \begin{pmatrix} M_0 & M_0 & s_1 \\ M_0 & M_0 \cup M_1 & M_1 \\ s_1 & M_1 & M_1 \\ 0 & 0 & 0 \end{pmatrix}. \end{matrix}$$

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

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

$$\begin{matrix} & 00u & u00 & 1u0 \\ \begin{matrix} 01u \\ u11 \\ 1u1 \end{matrix} & \begin{pmatrix} x_0 & x_0 & s \\ x_0 & x_0, x_1 & x_1 \\ s & x_1 & x_1 \\ 0 & 0 & 0 \end{pmatrix}. \end{matrix}$$

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  $\text{e-KW}_{\text{MUX}_n}^u$  to the following partition

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

Observe that the top-right block  $\begin{pmatrix} s_1 \\ M_1 \end{pmatrix}$  is the matrix of an  $\text{e-KW}_{\text{MUX}_{n-1}}^u$ . The bottom-right block  $\begin{pmatrix} M_1 & M_1 \\ 0 & 0 \end{pmatrix}$  can be solved, using Proposition 5.4, at the cost of solving  $\begin{pmatrix} M_1 \\ 0 \end{pmatrix}$  which is a block of  $\text{e-KW}_{\text{MUX}_{n-1}}^u$ . 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 5.4, at the cost of solving  $M_0$  which is a block of  $\text{KW}_{\text{MUX}_{n-1}}^u$ . Therefore, by Proposition 5.16 it has less complexity (size and depth) than  $\text{e-KW}_{\text{MUX}_{n-1}}^u$ . 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

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

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

As an illustration of the induction step in the proof above, we present a detailed example of  $\text{e-KW}_{\text{MUX}_2}^u$  with a decomposition into one block of  $\text{e-KW}_{\text{MUX}_1}^u$  (red), two blocks of  $\text{e-KW}_{\text{MUX}_1}^u$  (green) that can be solved at the cost of solving a single  $\text{e-KW}_{\text{MUX}_1}^u$  using Proposition 5.4, four blocks of  $\text{MUX}_1$  (blue) that can be solved at the cost of solving a single  $\text{MUX}_1$  using Proposition 5.4, and one (yellow) block of two identities that can be solved with depth 1. Therefore, the total depth is at most 5.

	000uuu	0u00uu	01u0uu	u00u0u	uu0000	u1u0u0	10uu0u	1uuu00	11uuu0
001uuu	$x_0$	$x_0$	$s_2$	$x_0$	$x_0$	$s_2$	$s_1$	$s_1$	$s_1, s_2$
0u11uu	$x_0$	$x_0, x_1$	$x_1$	$x_0$	$x_0, x_1$	$x_1$	$s_1$	$s_1$	$s_1$
01u1uu	$s_2$	$x_1$	$x_1$	$s_2$	$x_1$	$x_1$	$s_1, s_2$	$s_1$	$s_1$
u01ulu	$x_0$	$x_0$	$s_2$	$x_0, x_2$	$x_0, x_2$	$s_2$	$x_2$	$x_2$	$s_2$
uu1111	$x_0$	$x_0, x_1$	$x_1$	$x_0, x_2$	$x_0, x_1, x_2, x_3$	$x_1, x_3$	$x_2$	$x_2, x_3$	$x_3$
u1ulul	$s_2$	$x_1$	$x_1$	$s_2$	$x_1, x_3$	$x_1, x_3$	$s_2$	$x_3$	$x_3$
10uu1u	$s_1$	$s_1$	$s_1, s_2$	$x_2$	$x_2$	$s_2$	$x_2$	$x_2$	$s_2$
1uuu11	$s_1$	$s_1$	$s_1$	$x_2$	$x_2, x_3$	$x_3$	$x_2$	$x_2, x_3$	$x_3$
11uuu1	$s_1, s_2$	$s_1$	$s_1$	$s_2$	$x_3$	$x_3$	$s_2$	$x_3$	$x_3$
	0	0	0	0	0	0	0	0	0

We are now ready to prove Theorem 5.14.

*Proof of Theorem 5.14.* As mentioned in the beginning we will give a protocol to partition the communication matrix of  $\text{KW}_{\text{MUX}_n}^u$  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  $\frac{5}{2} \cdot 3^n - \frac{3}{2}$  for all  $n \geq 1$ .

Our protocol is the same as the one given in the proof of Lemma 5.17 with the small twist that we can save 1 monochromatic rectangle each time we encounter a  $\text{KW}_{\text{MUX}_n}^u$  matrix instead of a  $\text{e-KW}_{\text{MUX}_n}^u$  matrix. More formally:  $\text{size}^u(\text{KW}_{\text{MUX}_n}^u) = \text{size}^u(\text{e-KW}_{\text{MUX}_n}^u) - 1$ , because the additional entries in the matrix for  $\text{e-KW}_{\text{MUX}_{n-1}}^u$  are not present in the matrix for  $\text{KW}_{\text{MUX}_{n-1}}^u$ . Therefore the upper bound on depth follows readily and is not affected by this change.

To bound the size we consider the following recurrence for the number of monochromatic rectangles that we get in our partition of the communication matrix of  $\text{e-KW}_{\text{MUX}_n}^u$ :

$$T(n) = (T(n-1) - 1) + T(n-1) + 2 + T(n-1).$$

We first argue that  $T(n)$  indeed counts the number of monochromatic rectangles in the partition of  $\text{e-KW}_{\text{MUX}_n}^u$ . We recall the partition given by the induction step in Lemma 5.17.

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

The first summand  $(T(n-1) - 1)$  on the right in the recurrence is the contribution from the top-left part of the above matrix. There is a saving of 1 because we are only interested in counting the rectangles that cover the  $\text{KW}_{\text{MUX}_n}^u$  entries. This saving of 1 is possible, because the additional entries in the matrix for  $\text{e-KW}_{\text{MUX}_{n-1}}^u$  are not present in the matrix for  $\text{KW}_{\text{MUX}_{n-1}}^u$ .

The second summand  $T(n-1)$  on the right in the recurrence is the contribution from the top-right part of the above matrix. There are no savings here.

The third summand 2 on the right in the recurrence is the contribution from the bottom-left part of the above matrix. There are no savings here.

Finally, the fourth summand  $T(n-1)$  on the right in the recurrence is the contribution from the bottom-right part of the above matrix. There are no savings here. Thus, solving the recurrence  $T(n) = \frac{5}{2} \cdot 3^n - \frac{1}{2}$ . (from the base case in Lemma 5.17). We save an additional 1 monochromatic rectangle at the very end, because we are interested in  $\text{KW}_{\text{MUX}_n}^u$  and not  $\text{e-KW}_{\text{MUX}_n}^u$  gives an upper bound of  $\frac{5}{2} \cdot 3^n - \frac{3}{2}$  on the size.  $\square$

## 6 Alternation Depth Two and Two-round Protocols

In this section, we determine the hazard-free depth complexity of formulas of alternation depth 2 computing  $\text{MUX}_n$ . We assume without loss of generality that the output gate is an  $\vee$  gate since the prime implicants and prime implicates of  $\text{MUX}_n$  are symmetric. We exploit the correspondence between hazard-free formulas of alternation depth 2 and two-round communication protocols of the form: Alice sends some string, Bob replies with some string, and they settle on an answer (See Lemmas 7.1 and 7.2).

We begin by proving a property of hazard-free formulas of alternation depth 2 that has been observed in [Huf57]. We present a proof for completeness.

**6.1 Proposition.** *Let  $n \geq 1$ . In any two-round communication protocol for  $\text{MUX}_n$ , for different prime implicants, Alice must send different strings to Bob in the first round. Equivalently, in any hazard-free formula of alternation depth 2 computing  $\text{MUX}_n$ , for any prime implicant  $\alpha$ , there is at least one  $\wedge$  gate  $g$  in the formula such that the subformula at  $g$  is exactly the  $\wedge$  of literals in the prime implicant  $\alpha$ .*

*Proof.* Let  $\alpha, \beta \in \{0, u, 1\}^n$  be two distinct prime implicants of  $\text{MUX}_n$  (See Proposition 5.2). Then Bob must receive different strings from Alice for these two inputs to Alice, which can be seen as follows. Suppose Bob receives the same string from Alice for prime implicants  $\alpha$  and  $\beta$ . Without loss of generality (i.e., we can swap  $\alpha$  and  $\beta$ )  $\alpha \neq u^n$  and there exists a position  $i$  such that  $\{0, u, 1\} \ni \beta_i \neq \alpha_i \in \{0, 1\}$ . Let  $\alpha'$  be the prime implicate obtained from  $\alpha$  by flipping the  $i^{\text{th}}$  bit. Clearly, the only answer to the input  $(\alpha, \alpha')$  is  $i$  and  $i$  is a wrong answer for the input  $(\beta, \alpha')$ . Therefore, Alice must send different strings to Bob for  $\alpha$  and  $\beta$ .

Now, we prove the equivalent statement for formulas. Consider an arbitrary hazard-free formula  $F$  for  $\text{MUX}_n$  of alternation depth 2. For any prime implicant  $\alpha$  of  $\text{MUX}_n$ , since  $F$  is hazard-free, we have  $F(\alpha) = 1$ . This is possible only if there is a topmost  $\wedge$  gate  $g$  in  $F$  such that  $g(\alpha) = 1$ . Since the formula has alternation depth 2, the subformula at  $g$  is simply an  $\wedge$  of literals. Since it evaluates to 1 on  $\alpha$ , this set of literals has to be a subset of the literals in  $\alpha$ . It cannot be a proper subset because of minimality of prime implicants.  $\square$

We now observe a slightly weaker (than Theorem 6.5) depth lower bound using known results. We define the *size* of a prime implicant  $\alpha$  as  $|\{i \mid \alpha_i \neq u\}|$ .

**6.2 Proposition.** *For  $n \geq 1$ , we have  $\text{size}_2^u(\text{MUX}_n) = 4^n + 2n3^{n-1}$  and  $\text{depth}_2^u(\text{MUX}_n) \geq 2n + 1$ .*

*Proof.* Consider an arbitrary hazard-free formula  $F$  of alternation depth 2 computing  $\text{MUX}_n$ . By Proposition 6.1, for each prime implicant  $\alpha$  of  $\text{MUX}_n$ , there is at least one subformula that computes  $\alpha$ . Therefore, the size of  $F$  must be at least the sum of the sizes of all prime implicants. We now compute the sum of the sizes of all prime implicants of  $\text{MUX}_n$ . The size of any prime implicant where the selector bits have exactly  $i$  us is  $n - i + 2^i$ . This is because the other  $n - i$  selector bits must have Boolean values, and the subcube indexed by these selector bits contains  $2^i$  points, which must all have value 1 in the data bits. There are  $\binom{n}{i}$  ways to choose  $i$  positions for the selector bits with unstable values. For each such choice, there are  $2^{n-i}$  ways to set the remaining selector bits. Therefore, the sum of the sizes of all prime implicants in  $\text{MUX}_n$  is  $\sum_{i=0}^n \binom{n}{i} 2^{n-i} (2^i + n - i) = 4^n + 2n3^{n-1}$ , giving us the required size lower bound (This is also an upper bound using [Huf57]). We readily conclude  $\text{depth}_2^u(f) \geq \lceil \log_2(4^n + 2n3^{n-1}) \rceil = 2n + 1$ .  $\square$

We now show that the existence of low-depth hazard-free formulas of alternation depth 2 is linked to the existence of certain short prefix codes.

**6.3 Lemma.** *Let  $n \geq 1$ . For  $d \geq 0$ , there is a hazard-free formula of alternation depth 2 and depth  $d$  for  $\text{MUX}_n$  if and only if there is a prefix code for the set of all prime implicants of  $\text{MUX}_n$  such that for any  $i$  where  $0 \leq i \leq n$ , each prime implicant with exactly  $i$  us in the selector bits is encoded using at most  $d - \lceil \log_2(2^i + n - i) \rceil$  bits.*

*Proof.* Let  $i$  be the number of us in the selector bits of the prime implicant given to Alice. First, we determine a tight bound on the number of bits that Bob must transfer based on  $i$  (after Alice has transferred her prime implicant to Bob). Once Bob has received the prime implicant from Alice, the final answer could be any of the selector bits with Boolean values ( $n - i$  possibilities) or any of the data bits in the subcube indexed by Alice's selector bits ( $2^i$  possibilities). Since there are  $2^i + n - i$  distinct answers possible, Bob must use at least  $\lceil \log_2(2^i + n - i) \rceil$  bits in his reply. This

bound is tight. Once Bob receives the prime implicant from Alice, he can reply with the answer using at most  $\lceil \log_2(2^i + n - i) \rceil$  bits. Therefore, a two-round protocol of depth  $d$  exists if and only if there is a prefix code for the prime implicants that uses at most  $d - \lceil \log_2(2^i + n - i) \rceil$  bits to encode prime implicants with  $i$  us in the selector bits.  $\square$

We now prove that the optimal depth for hazard-free formulas with alternation depth 2 is  $2n + 2$ . This is the only depth lower bound in this paper that does not follow directly from a size lower bound. We will need the well-known Kraft's inequality giving a necessary and sufficient condition for the existence of a prefix code.

**6.4 Theorem** ([CT91, Theorem 5.2.1]). *For any binary prefix code, the codeword lengths  $\ell_1, \dots, \ell_m$  must satisfy the inequality*

$$\sum_{i=1}^m 2^{-\ell_i} \leq 1.$$

*Conversely, given a set of codeword lengths that satisfy this inequality, there exists a prefix code with these codeword lengths.*

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

*Proof.* From Lemma 6.3 we know that it suffices to find the minimal  $d$  for which there exists a prefix code for the set of all prime implicants of  $\text{MUX}_n$  such that for any  $i$ ,  $0 \leq i \leq n$ , each prime implicant with exactly  $i$  us in the selector bits is encoded using at most  $d - \lceil \log_2(2^i + n - i) \rceil$  bits. We know that there are  $\binom{n}{i} 2^{n-i}$  many prime implicants with exactly  $i$  us. Now using Kraft's inequality (Theorem 6.4) we have that the lengths of the encoding for each prime implicant must satisfy the following inequality

$$\sum_{i=0}^n \binom{n}{i} 2^{n-i} \cdot 2^{-(d - \lceil \log_2(2^i + n - i) \rceil)} \leq 1.$$

Rearranging we obtain

$$\sum_{i=0}^n \binom{n}{i} 2^{n-i} \cdot 2^{\lceil \log_2(2^i + n - i) \rceil} \leq 2^d.$$

For a fixed  $n$ , define the function  $\Psi(i) := \lceil \log_2(2^i + n - i) \rceil - i$ , where  $0 \leq i \leq n$ . Note that for all  $i \in \{0, 1, \dots, n\}$ ,  $\Psi(i) \geq 0$  and further  $\Psi(i)$  is an integer. Now consider  $\psi(i) := \log_2(2^i + n - i) - i$ . Clearly from the definitions we have  $\Psi(i) = \lceil \psi(i) \rceil$  for all  $i \in \{0, 1, \dots, n\}$ . From elementary calculus it follows that  $\psi(i)$  is a continuous function that is decreasing in the interval  $[0, n]$ . Furthermore observe that at  $i = 0$ ,  $\psi(0) = \log_2(n + 1)$  and  $\Psi(0) = \lceil \log_2(n + 1) \rceil$ , and at  $i = n - 1$ ,  $\Psi(n - 1) = \lceil \psi(n - 1) \rceil = 1$ . Therefore when  $n \geq 2$ ,  $\Psi(0) \geq 2$ , and thus by the continuity of  $\psi$  there exists a  $t \in \{1, \dots, n - 1\}$  such that for all  $i \in \{t, \dots, n - 1\}$ ,  $\Psi(i) = 1$ . Choose the minimal such  $t$ . In the following we will work with this minimal  $t$ .

Now breaking the summation at  $t - 1$  we get

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{n-i} \cdot 2^{\Psi(i)+i} + \sum_{i=t}^{n-1} \binom{n}{i} 2^{n-i} \cdot 2^{\Psi(i)+i} + \binom{n}{n} 2^n \leq 2^d.$$

Dividing by  $2^{n+1}$  on both sides we obtain

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} + \sum_{i=t}^{n-1} \binom{n}{i} 2^{\Psi(i)-1} + \binom{n}{n} \frac{1}{2} \leq 2^{d-n-1}. \quad (6.6)$$

Up to this point we only reformulated the property of the existence of a prefix code with the

desired properties. We now prove the theorem by first considering the lower bound and then the upper bound.

Now by the minimality of  $t$  and the integrality of  $\Psi$ , for  $0 \leq i \leq t-1$ ,  $\Psi(i) \geq 2$ , and for  $i \in \{t, \dots, n-1\}$ ,  $\Psi(i) = 1$ . Thus, plugging these values in the left hand side of the above inequality we see that the left hand side is strictly greater than  $2^n$ . Hence, we have

$$2^n < 2^{d-n-1}.$$

Therefore, we get that  $d > 2n+1$ , when  $n \geq 2$ .

We further observe that  $t$  is at most  $\lceil \log_2 n \rceil$ , because for all  $i \in \{\lceil \log_2 n \rceil, \dots, n-1\}$ , we have  $\Psi(i) = 1$ . We now claim that using  $t \leq \lceil \log_2 n \rceil$  and  $d = 2n+2$ , the inequality (6.6) is satisfied. Plugging the value of  $d$  and using the fact that  $\Psi(i) - 1 = 0$  for  $i \in \{t, \dots, n-1\}$  we can rewrite inequality (6.6) as follows

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} + \sum_{i=t}^{n-1} \binom{n}{i} + \binom{n}{n} \frac{1}{2} \leq 2^{n+1}. \quad (6.7)$$

Now using  $\sum_{i=0}^{n+1} \binom{n+1}{i} = 2^{n+1}$ , we obtain

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} + \sum_{i=t}^{n-1} \binom{n}{i} + \binom{n}{n} \frac{1}{2} \leq \sum_{i=0}^{n+1} \binom{n+1}{i}.$$

Moving the second summand to the right hand side we have

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} + \binom{n}{n} \frac{1}{2} \leq \sum_{i=0}^{n+1} \binom{n+1}{i} - \sum_{i=t}^{n-1} \binom{n}{i}.$$

Rewriting again we get

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} + \binom{n}{n} \frac{1}{2} \leq \sum_{i=0}^{t-1} \binom{n+1}{i} + \sum_{i=t}^{n+1} \binom{n+1}{i} - \sum_{i=t}^{n-1} \binom{n}{i}.$$

Further rewriting leads to

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} + \binom{n}{n} \frac{1}{2} \leq \sum_{i=0}^{t-1} \binom{n+1}{i} + \sum_{i=t}^{n-1} \left[ \binom{n+1}{i} - \binom{n}{i} \right] + \binom{n+1}{n} + \binom{n+1}{n+1}.$$

Now using the Pascal's rule,  $\binom{n+1}{i} - \binom{n}{i} = \binom{n+1}{i-1}$ , to simplify the second summand on the right hand side we have

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} + \binom{n}{n} \frac{1}{2} \leq \sum_{i=0}^{t-1} \binom{n+1}{i} + \sum_{i=t}^{n-1} \binom{n}{i-1} + \binom{n+1}{n} + \binom{n+1}{n+1}.$$

Rewriting and simplifying we obtain,

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} + \binom{n}{n} \frac{1}{2} \leq \sum_{i=0}^{t-1} \binom{n+1}{i} + \sum_{i=t-1}^{n-2} \binom{n}{i} + n+1+1.$$

Rewriting the right hand side again gives us

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} + \binom{n}{n} \frac{1}{2} \leq \sum_{i=0}^{t-1} \binom{n+1}{i} + \sum_{i=t-1}^n \binom{n}{i} + 1.$$

Moving the second summand on the left hand side to the right and simplifying we have

$$\sum_{i=0}^{t-1} \binom{n}{i} 2^{\Psi(i)-1} \leq \sum_{i=0}^{t-1} \binom{n+1}{i} + \sum_{i=t-1}^n \binom{n}{i} + \frac{1}{2}.$$

Now we first observe that the right hand side is strictly greater than  $2^n$ . Using  $t \leq \lceil \log_2 n \rceil$  we now show that the left hand side is at most  $2^{\log_2^2 n + 2 \log_2 n}$ . Thus showing that the inequality is satisfied. Using the fact that  $\Psi(i) - 1 \leq \Psi(0) - 1 \leq \lceil \log_2(n+1) \rceil - 1 \leq \log_2 n$  and simplifying the left hand side, we obtain the following upper bound on it

$$n \sum_{i=0}^{t-1} \binom{n}{i}.$$

Now using  $\sum_{i=0}^k \binom{n}{i} \leq 2^{\mathbb{H}(k/n) \cdot n}$  for  $k/n \leq 1/2$  and  $\mathbb{H}$  being the Shannon entropy function, we can further bound it by

$$n \cdot 2^{\mathbb{H}(\frac{t-1}{n}) \cdot n}.$$

Now since  $t - 1 \leq \log_2 n$ , we obtain the following upper bound on it

$$2^{\log_2 n + \log_2^2 n + \log_2 n},$$

where we used  $\mathbb{H}(\log_2 n/n) \cdot n \leq \log_2^2 n + \log_2 n$ . For  $n \in \{1, \dots, 38\}$  we verified eq. (6.7) with a computer calculation, see Section 10. The fact that  $n \geq \log_2^2 n + 2 \log_2 n$  for all  $n \geq 39$  completes the proof.  $\square$

## Appendix

### 7 Proofs for the hazard-free Karchmer-Wigderson game

In this section we prove Theorem 4.6. We split the proof into two lemmas.

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

$$\text{CC}(\text{KW}_f^u) \leq \text{depth}(F), \quad \text{and} \quad \text{size}^P(\text{KW}_f^u) \leq \text{size}(F).$$

**7.2 Lemma.** *Let  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  be a Boolean function and  $\Pi$  be a protocol for the hazard-free KW-game  $\text{KW}_f^u$ . Then,*

$$\text{depth}^u(f) \leq \text{CC}(\Pi), \quad \text{and} \quad \text{size}^u(f) \leq \text{size}^P(\Pi).$$

The proofs of Lemmas 7.1 and 7.2 are natural generalizations of their corresponding counterparts in the original setting of the KW-game. However, for the sake of completeness and to highlight the differences, we present the proofs below.

*Proof of Lemma 7.1. (Formula to Protocol.)* Let  $F$  be a hazard-free De Morgan formula computing the function  $f$ . It suffices to show a protocol  $\Pi$  solving  $\text{KW}_f^u$  such that  $\text{CC}(\Pi) \leq \text{depth}(F)$  and  $\text{size}^P(\Pi) \leq \text{size}(F)$ . Let  $\alpha \in \bar{f}^{-1}(1)$  be Alice's input and  $\beta \in \bar{f}^{-1}(0)$  be Bob's input.

In the protocol  $\Pi$  both players keep track of a subformula  $G$  of  $F$  such that  $G(\alpha) = 1$  and  $G(\beta) = 0$ . We begin at the root of  $F$ , i.e.,  $G = F$ . By the hazard-free property of  $F$ , we have  $G(\alpha) = F(\alpha) = \bar{f}(\alpha) = 1$  and  $G(\beta) = F(\beta) = \bar{f}(\beta) = 0$ . Now depending on the type of gate at the root of  $G$ , we decide which player sends the message.

If  $G = G_0 \vee G_1$ , then it is Alice's turn. Observe that, since  $G(\alpha) = 1$ , there exists  $i \in \{0, 1\}$  such that  $G_i(\alpha) = 1$ . Again by a similar reasoning, we also have  $G_0(\beta) = G_1(\beta) = 0$ . Thus, Alice can send a single bit to Bob indicating a child which evaluates to 1. Then they both move to the corresponding subformula.

If  $G = G_0 \wedge G_1$ , then it is Bob's turn. Again by a similar reasoning,  $G(\alpha) = 1$  and  $G(\beta) = 0$ , we deduce that  $G_0(\alpha) = G_1(\alpha) = 1$  and there exists  $i \in \{0, 1\}$  such that  $G_i(\beta) = 0$ . Thus, Bob can send a single bit to Alice indicating a child which evaluates to 0 and they both move to the

corresponding subformula.

If  $G$  is a leaf of  $F$  and is labeled with literal  $x_i$  or  $\neg x_i$  for some  $i \in [n]$ , then the protocol returns the coordinate  $i$  as answer.

Clearly, the number of bits exchanged on any input equals the depth of the leaf reached and the number of leaves in the protocol  $\Pi$  equals the number of leaves in  $F$ . That is,  $\text{CC}(\Pi) = \text{depth}(F)$  and  $\text{size}^P(\Pi) = \text{size}(F)$ . The correctness of the protocol follows from observing that when  $G$  is a literal and  $G(\alpha) = 1$  and  $G(\beta) = 0$  then the variable corresponding to the literal has different (Boolean) values in the certificates  $\alpha$  and  $\beta$ .  $\square$

We now give the translation from protocols to formulas.

*Proof of Lemma 7.2.* (Protocol to Formula.) Let  $\Pi$  be a protocol solving  $\text{KW}_f^u$ . It suffices to show a hazard-free formula  $F$  computing  $f$  such that  $\text{depth}(F) \leq \text{CC}(\Pi)$  and  $\text{size}(F) \leq \text{size}^P(\Pi)$ .

Let  $T$  be the protocol tree of  $\Pi$ . We will convert the tree  $T$  into a hazard-free formula  $F$  for  $f$ . Every internal node in the tree is associated with a player whose turn it is to send the message. To obtain  $F$  we first replace every internal node in  $T$  as follows:

- If it is associated with Alice, then replace it with  $\vee$ ,
- If it is associated with Bob, then replace it with  $\wedge$ .

Now consider a leaf  $\ell \in T$  and suppose that the output at this leaf is some coordinate  $i \in [n]$ . Let  $A_\ell \times B_\ell$  be the set of inputs that reaches this leaf  $\ell$ . This is a monochromatic combinatorial rectangle. By definition of  $\text{KW}_f^u$ , exactly one of the following cases holds:

- (i) for all  $\alpha \in A_\ell$ ,  $\alpha_i = 1$  and for all  $\beta \in B_\ell$ ,  $\beta_i = 0$ , or
- (ii) for all  $\alpha \in A_\ell$ ,  $\alpha_i = 0$  and for all  $\beta \in B_\ell$ ,  $\beta_i = 1$ .

In the first case we label the leaf  $\ell$  in  $F$  with the literal  $x_i$ , while in the second we label it with  $\neg x_i$ . Clearly the constructed formula  $F$  has depth and size equal to  $\text{CC}(\Pi)$  and  $\text{size}^P(\Pi)$ , respectively. So it remains to argue the correctness of the transformation. That is, we need to verify that  $F$  is indeed a hazard-free formula for  $f$ . It suffices to show the following:

for every node  $v \in F$ , the subformula  $G$  rooted at  $v$  satisfies  $G(\alpha) = 1$  for all  $\alpha \in A$  and  $G(\beta) = 0$  for all  $\beta \in B$ , where  $A \times B$  is the set of the inputs that reach the node corresponding to  $v$  in the protocol tree  $T$ .

The correctness now follows by applying this claim to the root of  $F$  (note that for the root we have  $A = \bar{f}^{-1}(1)$  and  $B = \bar{f}^{-1}(0)$ ). We prove the claim by induction on the depth of nodes in the formula.

**Base case:**  $\text{depth} = 0$ . The claim holds for the leaf nodes by our construction (i.e., by our choices of their labels).

**Induction step:** Suppose the claim holds for the children  $v_0$  and  $v_1$  of a certain node  $v \in F$ . We will now show that it also holds for  $v$ . Let  $G, G_0, G_1$  be the subformulas rooted at  $v, v_0, v_1$ , respectively. We assume, w.l.o.g., that  $G = G_0 \vee G_1$ . (The other case being symmetric.) Let  $v', v'_0, v'_1 \in T$  be the corresponding nodes to  $v, v_0, v_1 \in F$ . Let  $A \times B$  be the inputs reaching  $v'$  in  $T$ . Since  $G = G_0 \vee G_1$ , it is Alice's turn to send the message. Therefore, Alice's message partitions  $A$  into  $A_0$  and  $A_1$  such that  $A_0 \times B$  is the inputs reaching  $v'_0$  and  $A_1 \times B$  is the inputs reaching  $v'_1$ . Thus, by induction hypothesis, we have

- for all  $\alpha \in A_0$ ,  $G_0(\alpha) = 1$  and for all  $\beta \in B$ ,  $G_0(\beta) = 0$ ,
- for all  $\alpha \in A_1$ ,  $G_1(\alpha) = 1$  and for all  $\beta \in B$ ,  $G_1(\beta) = 0$ .

This in turn implies that for all  $\alpha \in A$ ,  $G(\alpha) = G_0(\alpha) \vee G_1(\alpha) = 1$  and for all  $\beta \in B$ ,  $G(\beta) = G_0(\beta) \vee G_1(\beta) = 0$ .  $\square$



## 8 Limited hazard-freeness

Avoiding all hazards can be very expensive (see [IKL<sup>+</sup>19, Juk21]) 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 [FKLP17] 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 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 output  $uuuu$  on input  $1111111u$ , so the Gray code is used. 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 [LM16, BLM17, BLM18, BLM20]. Another interesting class of hazards that should be avoided are the inputs where the number of  $u$  is bounded from above. This is the setting of  $k$ -bit hazard-freeness from [IKL<sup>+</sup>19].

Theorem 4.6 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.

**8.1 Proposition.** *Given sets  $A$  and  $B$  with  $f^{-1}(1) \subseteq A \subseteq \bar{f}^{-1}(1)$  and  $f^{-1}(0) \subseteq B \subseteq \bar{f}^{-1}(0)$ . 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 4.6.

**8.2 Example.** *Consider the function  $\text{MUX}_2(s_1, s_2, x_{00}, x_{01}, x_{10}, x_{11})$ . We proved in Theorem 5.12 that any hazard-free formula for  $\text{MUX}_2$  requires 17 leaves. Suppose we only want to avoid hazards where both selector bits are unstable. Note that every other possible hazard is covered by the eight prime implicants and eight prime impicates of  $\text{MUX}_2$  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.*

	000uuu	0u00uu	01u0uu	u00u0u	u1u0u0	10uu0u	1uuu00	11uuu0
001uuu	$x_{00}$	$x_{00}$	$s_2$	$x_{00}$	$s_2$	$s_1$	$s_1$	$s_1, s_2$
0u11uu	$x_{00}$	$x_{00}, x_{01}$	$x_{01}$	$x_{00}$	$x_{01}$	$s_1$	$s_1$	$s_1$
01uluu	$s_2$	$x_{01}$	$x_{01}$	$s_2$	$x_{01}$	$s_1, s_2$	$s_1$	$s_1$
u01ulu	$x_{00}$	$x_{00}$	$s_2$	$x_{00}, x_{10}$	$s_2$	$x_{10}$	$x_{10}$	$s_2$
u1ulul	$s_2$	$x_{01}$	$x_{01}$	$s_2$	$x_{01}, x_{11}$	$s_2$	$x_{11}$	$x_{11}$
10uu1u	$s_1$	$s_1$	$s_1, s_2$	$x_{10}$	$s_2$	$x_{10}$	$x_{10}$	$s_2$
1uuu11	$s_1$	$s_1$	$s_1$	$x_{10}$	$x_{11}$	$x_{10}$	$x_{10}, x_{11}$	$x_{11}$
11uuu1	$s_1, s_2$	$s_1$	$s_1$	$s_2$	$x_{11}$	$s_2$	$x_{11}$	$x_{11}$

Indeed, the above colouring yields a protocol of size 16. We can also show that we cannot do better. Substitute  $s_1 = s_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$  and  $x_{00} = x_{01} = x_{10} = x_{11} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ . This yields the following  $16 \times 16$  (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.

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

## 9 Hazard-free formula depth reduction

In this section we show that the standard depth reduction process for Boolean circuits and formulas works in a hazard-free way. Let  $F$  be a hazard-free formula. In the Boolean depth reduction process we take a gate that has roughly the same distance to the root as it has to its deepest leaf and we write

$$F(x) = F'(G(x), x)$$

where  $G$  is the subformula of that gate. Now we observe that

$$F(x) = F'(G(x), x) = \text{MUX}_1(G(x), F'(0, x), F'(1, x)) =: \tilde{F}(x) \quad (9.1)$$

which can be used iteratively to convert a Boolean formula to logarithmic (in the size of  $F$ ) depth. If the implementation of  $\text{MUX}_1$  is hazard-free, then this depth-reduction process preserves the hazard-freeness of a formula, as the next claim shows.

**9.2 Claim.** *Assume that a hazard-free implementation of  $\text{MUX}_1$  is used in (9.1). If  $F$  is hazard-free, then  $\tilde{F}$  is also hazard-free.*

*Proof.* Consider the case when  $\tilde{F}(\alpha) = u$ . It remains to show that  $F(\alpha) = u$ . The hazard-freeness of the multiplexer implementation implies that there are the following three cases:

- $G(\alpha) = 0$  and  $F'(0, \alpha) = u$
- $G(\alpha) = 1$  and  $F'(1, \alpha) = u$
- $G(\alpha) = u$  and  $F'(0, \alpha)$  and  $F'(1, \alpha)$  are not both equal to the same Boolean value (in fact, they potentially have value  $u$ ).

In the first two cases, by definition of  $F'$  and  $G$  we see that  $F(\alpha) = F'(G(\alpha), \alpha) = u$ . In the third case we see that  $F'(u, \alpha) = u$ . Using the fact that  $G(\alpha) = u$  implies that  $F(\alpha) = F'(G(\alpha), \alpha) = u$ .  $\square$

By using standard techniques for finding the subformula  $G$  (See Lemma 1.3 in [Juk12]), we can show that the balanced formula has depth  $3 \log_{3/2}(m) + O(1)$  and size  $O(m^{2.92})$  where  $m$  is the size of the original formula.

## 10 Sagemath Source Code for the Finite Cases

The following sagemath (version 9) code is used to verify Equation (6.7) up to  $n = 38$ .

```
def blog(k):
    return float(log(k,2))

def Psi(i,n):
    return int(ceil( blog(2i+n-i ) )-i)

def t(n):
    return int(ceil(blog(n)))

def LHS(n):
    return sum([binomial(n,i)*2(Psi(i,n)-1) for i in [0..t(n)-1]]) + \
        sum([binomial(n,i) for i in [t(n)..n-1] ]) + 0.5

def RHS(n):
    return 2(n+1)

for n in [1..38]:
    if RHS(n)>=LHS(n):
        print("ok")
    else:
        print("problem")
```

## References

- [AB87] N. Alon and R. B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, January 1987.
- [And87] AE Andreev. On one method of obtaining constructive lower bounds for the monotone circuit size. *Algebra and Logics*, 26(1):3–26, 1987.
- [BEI01] J. Brzozowski, Z. Esik, and Y. Iland. Algebras for hazard detection. In *Proc. 31st International Symposium on Multiple-Valued Logic*, 2001.
- [BHT<sup>+</sup>17] Andrew Becker, Wei Hu, Yu Tai, Philip Brisk, Ryan Kastner, and Paolo 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.
- [BLM17] Johannes Bund, Christoph Lenzen, and Moti Medina. Near-optimal metastability-containing sorting networks. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, pages 226–231, 2017.
- [BLM18] Johannes Bund, Christoph Lenzen, and Moti Medina. Optimal metastability-containing sorting networks. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 521–526, 2018.

- [BLM20] Johannes Bund, Christoph Lenzen, and Moti Medina. Optimal metastability-containing sorting via parallel prefix computation. *IEEE Transactions on Computers*, 69(2):198–211, 2020.
- [BNT98] Robert Beals, Tetsuro Nishino, and Keisuke Tanaka. On the complexity of negation-limited boolean networks. *SIAM Journal on Computing*, 27(5):1334–1347, 1998.
- [BS95] Janusz A. Brzozowski and Carl-Johan H. Seger. *Asynchronous circuits*. Springer New York, 1995.
- [Cal58] Samuel H. Caldwell. *Switching Circuits and Logical Design*. John Wiley & Sons Inc, 1958.
- [CM78] Ashok K Chandra and George Markowsky. On the number of prime implicants. *Discrete Mathematics*, 24(1):7–11, 1978.
- [CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
- [Eic65] E. B. Eichelberger. Hazard detection in combinational and sequential switching circuits. *IBM J. Res. Dev.*, 9(2):90–99, March 1965.
- [FFL18] S. Friedrichs, M. Függer, and C. Lenzen. Metastability-containing circuits. *IEEE Transactions on Computers*, 67(08):1167–1183, aug 2018.
- [Fis75] Michael J Fischer. The complexity of negation-limited networks—a brief survey. In *Automata Theory and Formal Languages*, pages 71–82. Springer, 1975.
- [FKLP17] Matthias Függer, Attila Kinali, Christoph Lenzen, and Thomas 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.
- [FM05] G. S. Frandsen and P. B. Miltersen. Reviewing bounds on the circuit size of the hardest functions. *Information Processing Letters*, 95(2):354–357, 2005.
- [FMT21] 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, ITCSC 2021*, volume 185 of *LIPICs*, pages 89:1–89:7. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- [Got49] 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.
- [GP14] 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.
- [GS95] Michelangelo Grigni and Michael Sipser. Monotone separation of logarithmic space from logarithmic depth. *Journal of Computer and System Sciences*, 50(3):433–437, 1995.

- [Hås98] J. Håstad. The shrinkage exponent of de morgan formulas is 2. *SIAM Journal on Computing*, 27(1):48–64, 1998.
- [HOI<sup>+</sup>12] 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.
- [HR00] 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.
- [Huf57] David A. Huffman. The design and use of hazard-free switching networks. *J. ACM*, 4(1):47–62, January 1957.
- [IKL<sup>+</sup>19] 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.
- [Juk12] Stasys Jukna. *Boolean function complexity: advances and frontiers*, volume 27. Springer Science & Business Media, 2012.
- [Juk21] S. Jukna. Notes on hazard-free circuits. *SIAM Journal on Discrete Mathematics*, 35(2):770–787, 2021.
- [Khr71] VM Khrapchenko. Complexity of the realization of a linear function in the class of  $\Pi$ -circuits. *Mathematical Notes of the Academy of Sciences of the USSR*, 9(1):21–23, 1971.
- [Kle38] Stephen Cole Kleene. On notation for ordinal numbers. *The Journal of Symbolic Logic*, 3(4):150–155, 1938.
- [Kle52] Stephen Cole Kleene. *Introduction to Metamathematics*. North Holland, 1952.
- [KN96] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1996.
- [Kör66] Stephan Körner. *Experience and theory : an essay in the philosophy of science*. International library of philosophy and scientific method. Routledge & Kegan Paul, London, 1966.
- [KW90] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990.
- [LM16] Christoph Lenzen and Moti Medina. Efficient metastability-containing gray code 2-sort. In *2016 22nd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 49–56, 2016.
- [Lup58] Oleg B Lupanov. On a method of circuit synthesis. *Izvestia VUZ*, 1:120–140, 1958.
- [Lup60] O. B. Lupanov. On the complexity of realization of functions of propositional calculus by formulas. *Problemy Kibernetiki*, 3:61–80, 1960. in Russian.

- [Lut92] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44(2):220–258, 1992.
- [Mal14] Grzegorz Malinowski. Kleene logic and inference. *Bulletin of the Section of Logic*, 43(1/2):42–52, 2014.
- [MS82] Kurt Mehlhorn and Erik 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.
- [MSB12] Michael Mendler, Thomas R. Shiple, and Gérard Berry. Constructive boolean circuits and the exactness of timed ternary simulation. 40, 2012.
- [Muk72] M. Mukaidono. On the b-ternary logical function - a ternary logic considering ambiguity. *Systems, Computers, Controls*, 3(3):27–36, 1972.
- [Muk83a] 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.
- [Muk83b] 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.
- [ND92] 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, Nov 1992.
- [PR17] 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.
- [Raz85a] Alexander A Razborov. Lower bounds for the monotone complexity of some boolean functions. In *Soviet Math. Dokl.*, volume 31, pages 354–357, 1985.
- [Raz85b] Alexander 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.
- [Raz87] Alexander 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.
- [RM99] R. Raz and P. McKenzie. Separation of the monotone NC hierarchy. *Comb.*, 19(3):403–435, 1999.
- [RS42] John Riordan and Claude E Shannon. The number of two-terminal series-parallel networks. *Journal of Mathematics and Physics*, 21(1-4):83–93, 1942.
- [RW92] Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM (JACM)*, 39(3):736–744, 1992.

- [RY20] A. Rao and A. Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.
- [Sha49] Claude E Shannon. The synthesis of two-terminal switching circuits. *The Bell System Technical Journal*, 28(1):59–98, 1949.
- [Tar88] Éva Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.
- [TFL17] Ghaith Tarawneh, Matthias Függer, and Christoph Lenzen. Metastability tolerant computing. In *2017 23rd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 25–32, 2017.
- [TNB96] Keisuke Tanaka, Tetsuro Nishino, and Robert Beals. Negation-limited circuit complexity of symmetric functions. *Information processing letters*, 59(5):273–279, 1996.
- [TWM<sup>+</sup>09] Mohit Tiwari, Hassan M.G. Wassel, Bitu Mazloom, Shashidhar Mysore, Frederic T. Chong, and Timothy Sherwood. Complete information flow tracking from the gates up. *SIGARCH Comput. Archit. News*, 37(1):109–120, March 2009.
- [Weg87] Ingo Wegener. *The complexity of Boolean functions*. John Wiley & Sons, Inc., 1987.
- [YR64] Michael Yoeli and Shlomo Rinon. Application of ternary algebra to the study of static hazards. *J. ACM*, 11(1):84–97, jan 1964.