Discrete Applied Mathematics 158 (2010) 1195–1204

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

# An asymptotically optimal lower bound on the OBDD size of the middle bit of multiplication for the pairwise ascending variable order

Martin Sauerhoff

*Krengelstrasse 9, 44869 Bochum, Germany*

ARTICLE INFO

*Article history:*
Received 7 July 2008
Received in revised form 30 November 2009
Accepted 3 March 2010
Available online 18 April 2010

*Keywords:*
Multiplication
Middle bit
OBDD
Lower bound
Universal hashing

ABSTRACT

We prove that each OBDD (ordered binary decision diagram) for the middle bit of $n$-bit integer multiplication for one of the variable orders which so far achieve the smallest OBDD sizes with respect to asymptotic order of growth, namely the pairwise ascending order $x_0, y_0, \ldots, x_{n-1}, y_{n-1}$, requires a size of $\Omega(2^{(6/5)n})$. This is asymptotically optimal due to a bound of the same order by Amano and Maruoka (2007) [1].



## 1. Introduction

OBDDs (ordered binary decision diagrams) are a graphical representation for boolean functions with strong algorithmic properties that are used in a wide range of applications, most prominently in hardware verification. For a thorough introduction into practical and theoretical aspects of this model, see, e.g., Wegener's monograph [8].

**Definition 1.** Let $X = \{x_1, \ldots, x_n\}$ be a set of variables and let $\pi$ be a permuted list of the variables in $X$ called a *variable order*. An *OBDD (ordered decision diagram) on X with respect to* $\pi$ is a directed graph with the following properties. The graph has a designated start node and sinks labeled by the boolean constants 0 or 1. Each internal node is labeled by a variable from $X$ and has two outgoing edges labeled by 0 and 1, resp. For each path in the graph, the sequence of variables at its nodes is required to be a subsequence of $\pi$. Each node $v$ in the OBDD represents a boolean function $f_v\colon \{0,1\}^n \to \{0,1\}$ in the following way. A given input $a \in \{0,1\}^n$ defines a path from $v$ to one of the sinks where each node labeled by variable $x_i$ is left via the edge labeled by $a_i$. The output $f_v(a)$ equals the label of the reached sink. The *size* of the OBDD is the number of its nodes. The function represented by the OBDD is the function represented by its start node. Finally, the OBDD size of a boolean function is the minimum size of an OBDD representing it.

The size of an OBDD directly corresponds to its storage requirement. Furthermore, the run time for basic operations on boolean functions represented by OBDDs such as applying boolean operations is a polynomial in the size of the involved OBDDs. For applications of OBDDs, size is therefore the decisive parameter. Several practically important functions have OBDDs of a small polynomial size in the input length, at least if an appropriate variable order is chosen, while others require an exponential size regardless of the variable order.

Integer multiplication is obviously a highly practically relevant function while at the same time it is also a notoriously difficult benchmark problem for representations of boolean functions like OBDDs and a well-known bottleneck in the

*E-mail address:* martin.sauerhoff@udo.edu.

0166-218X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2010.03.002

verification of arithmetic circuits. More precisely, we are interested in the binary encoding of integer multiplication defined as follows.

**Definition 2.** For nonnegative integers with binary representations $x, y \in \{0, 1\}^n$, let $(z_{2n-1}, \ldots, z_0) \in \{0, 1\}^{2n}$ denote the binary representation of their product. Then the *ith output bit of n-bit multiplication*, for $i \in \{0, \ldots, 2n-1\}$, is defined by $\mathrm{MUL}_{i,n}(x, y) := z_i$.

At the time of writing, even representing all the output bits of 16-bit multiplication in a single OBDD is still a challenging task for standard PC hardware (it requires more than 3 GB of storage and a clever, non-standard implementation of the algorithms [11]). It is known from experiments that one of the problems is that the different output bits of this function have very different optimal variable orders. One could therefore hope that we can at least represent the output bits by separate OBDDs of a moderate size if the variable order is chosen appropriately, which would be sufficient for many applications.

Bryant took the first step in destroying this hope by proving that OBDDs for the middle bit of multiplication, $\mathrm{MUL}_{n-1,n}$, require an exponential size $\Omega(2^{n/8})$ for any variable order. He also motivated looking at the middle bit by the fact that, by reductions, lower bounds for it also imply lower bounds of a similar size for $\mathrm{MUL}_{i,n}$ with $i$ close to $n-1$. A more profane motivation is that due to symmetry properties, one can hope that lower bounds are easier to obtain than for other bits. Experiments indicate that the most difficult bit of integer multiplication (with respect to OBDD size) is in fact *not* the middle bit, but some yet unknown bit with a higher index.

Introducing a new technique based on universal hashing, Woelfel [10] managed to improve Bryant's lower bound for $\mathrm{MUL}_{n-1,n}$ considerably to $2^{\lfloor n/2 \rfloor}/61 - 4$. Furthermore, he also showed the first nontrivial upper bound of size $7/3 \cdot 2^{(4/3)n}$, choosing the variable order $x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}$. Amano and Maruoka [1] improved the upper bound to $2.8 \cdot 2^{(6/5)n}$ even for quasi-reduced OBDDs, i.e. OBDDs where on each path from the start node to a sink, all variables have to appear. For this, they used the variable order $x_0, y_0, \ldots, x_{n-1}, y_{n-1}$. Based on a comparison with the optimal sizes of quasireduced OBDDs for input lengths up to $n = 12$, they conjectured that their result is in fact asymptotically optimal. Further papers have dealt with the complexity of the middle bit of multiplication for more general models than OBDDs [5,7,9] or most recently with the OBDD size of the most significant bit [2–4].

Despite the considerable amount of research dealing with the complexity of the middle bit function, the gap between lower and upper bounds for its OBDD size is still large. Applying the best known lower bound due to Woelfel to the most important input lengths today, $n = 32$ and $n = 64$, yields that 1071 nodes and 70.4 million nodes, resp., are required. These numbers are both too small to explain why we still cannot construct the respective OBDDs using current standard PC hardware. (E.g., the usual representation of an OBDD node consists of three pointers, a reference count, and some boolean flags, which on a 32-bit operating system all fit into 16 bytes. Assuming 2 GB of memory leads to manageable OBDD sizes in the order of $10^8$.)

The contribution of this paper is to show that the upper bound of Amano and Maruoka is in fact asymptotically optimal for the order chosen by them, which is also one of the orders which so far achieve the smallest OBDD sizes with respect to an asymptotic order of growth. More precisely, we obtain:

**Theorem 3.** *The size an OBDD with variable order $x_0, y_0, \ldots, x_{n-1}, y_{n-1}$ for $\mathrm{MUL}_{n-1,n}$ is at least $(3 - 2\sqrt{2})/4 \cdot 2^{3\lfloor (2/5)n \rfloor} - 1$.*

Thus, for $n = 32$ already more than 2.9 billion nodes are required. For $n = 64$, the bound is larger than $1.62 \times 10^{21}$. These numbers surely explain why we cannot construct the corresponding OBDDs for this variable order, and we are only left with the possibility that there are considerably better variable orders.

We remark that the lower bound in Theorem 3 does not follow in an obvious way by just "fine-tuning" the known results. This is ruled out by the observation of Woelfel [10] that any lower bound that is obtained by setting all variables of one of the factors of multiplication to constants, which is true for all previous proofs, can only be of order $O(2^{n/2})$.

The rest of the paper is organized as follows. We first introduce some notation and general lemmas in the next section. We then state two main lemmas and apply them to prove Theorem 3 (Section 3). Finally, in Sections 4 and 5, we prove these main lemmas. We conclude with OBDD sizes of $\mathrm{MUL}_{n-1,n}$ for different variable orders determined by experiments.

## 2. Preliminaries

For a nonnegative integer $x$ represented by the boolean vector $(x_{n-1}, \ldots, x_0)$ in binary and $\ell \le h$, let $[x]_\ell^h$ be the number represented by $(x_h, \ldots x_\ell)$ in binary and let $[x]_\ell = x_\ell$. For integers $x, y$ with $y \ne 0$, let $x \operatorname{div} y := \lfloor x/y \rfloor$.

We use the following number theoretic facts which are easy to verify.

**Proposition 4.** (1) *For integers x and i, n with* $i \in \{0, \ldots, n-1\}$, $(x \bmod 2^n) \operatorname{div} 2^i = [x]_i^{n-1} = (x \operatorname{div} 2^i) \bmod 2^{n-i}$.
(2) *Let $x$ and $m, y > 0$ be integers such that $y$ divides $m$. Then* $(xy) \bmod m = y \cdot (x \bmod (m/y))$.
(3) *Let $x$, $y$ and $d' \ge d > 0$ be integers such that $d$ divides $d'$. Then* $(dx + y) \operatorname{div} d' = (x + y \operatorname{div} d) \operatorname{div} (d'/d)$.
(4) *Let $x$, $y$ and $d > 0$ be integers. Then $x \operatorname{div} d = y \operatorname{div} d$ implies that $|x - y| < d$.*
(5) *Let $x \ge 0$ and $d > 0$ be integers. Then $x \operatorname{div} d = 0$ iff $x < d$.*

For any integer $m \geq 2$, let $\mathbb{Z}_m$ be the ring of integers modulo $m$ and let $\mathbb{Z}_m^*$ denote the set of their invertible elements with respect to multiplication. It is a well-known fact that $\mathbb{Z}_m^* = \{x \mid x \in \mathbb{Z}_m, \gcd(x, m) = 1\}$.

For any boolean function $f: \{0,1\}^n \to \{0,1\}$ defined on variables from $X = \{x_1, \dots, x_n\}$ and some partial assignment $a$ to a subset $X' \subseteq X$ of the variables, we use $f|_a$ to denote the subfunction of $f$ that is obtained by fixing the variables in $X'$ according to $a$.

Finally, we apply the well-known method for proving lower bounds on the size of OBDDs in the following form (a proof is given in [10]).

**Lemma 5.** *Let $G$ be a $\pi$-OBDD representing the function $f: \{0,1\}^n \to \{0,1\}$. Let $s$ be the number of subfunctions of $f$ obtained by setting a fixed number of variables according to $\pi$ to constants. Then $|G| \geq 2s - 1$.*

## 3. Main lemmas and proof of Theorem 3

We first state two main lemmas and then combine them for proving the main theorem of the paper. Before we give the details, we present an outline of the whole proof.

Our general plan is straightforward: For an appropriately chosen $i \in \{1, \dots, n\}$, we count the number subfunctions of $\mathrm{MUL}_{n-1,n}$ resulting from setting the variables $x_0, y_0, \dots, x_{i-1}, y_{i-1}$ to constants in all possible ways and then apply Lemma 5. The counting of subfunctions is done with the aid of the two main lemmas.

We start with the observation of Amano and Maruoka [1] that, due to the structure of the multiplication problem, an *upper bound* on the number of subfunctions $(\mathrm{MUL}_{n-1,n})|_{a,b}$ for assignments $a, b \in \{0,1\}^i$ to the variables $x_0, \dots, x_{i-1}$ and $y_0, \dots, y_{i-1}$, resp., is given by the number the triples $([a]_0^{n-i-1}, [b]_0^{n-i-1}, [ab]_i^{n-1})$. As our first main lemma, we prove that this characterization is in fact almost one-to-one, namely the number of subfunctions is also *lower bounded* by half of the number of these triples. The proof is by elementary number theoretic arguments and is given in Section 4.

It thus suffices to prove a strong enough lower bound on the number of triples of this form. This can be done by fixing the first two components in an arbitrary way and then lower bounding the number of third components that are still possible. Our second main lemma says that, for $i \geq (3/5)n$, there is a constant fraction of all $2^{2(n-i)}$ assignments to the first components such that a constant fraction of all $2^{n-i}$ possible third components appear. Thus, we get at least $c \cdot 2^{3(n-i)}$ triples altogether, $c > 0$ a constant, giving the desired lower bound of order $2^{\Omega((6/5)n)}$ for $i = (3/5)n$.

For the proof of the second main lemma, we use arguments from the theory of hashing, following the approach of Woelfel [10]. For assignments $\widetilde{a} := [a]_0^{n-i-1}$ and $\widetilde{b} := [b]_0^{n-i-1}$ to the first two components of the considered triples, the third component computes the hash function $h_{\widetilde{a},\widetilde{b}}$ mapping the remaining bits of $a$ and $b$, $u := [a]_{n-i}^{i-1}$ and $v := [b]_{n-i}^{i-1}$, to $h_{\widetilde{a},\widetilde{b}}(u, v) := [(u2^{n-i} + \widetilde{a})(v2^{n-i} + \widetilde{b})]_i^{n-1}$. By showing that an average hash function of this kind has only few collisions, where the average is over the choices for $\widetilde{a}$ and $\widetilde{b}$, we then get that there must be a large portion of such functions with a large range. The details of this are given in Section 5.

We now prepare the statement of the first main lemma by presenting the fact used by Amano and Maruoka [1] as the basis for their upper bound on the size of OBDDs for the middle bit of multiplication.

**Proposition 6** ([1]). *Let $n/2 \leq i \leq n-1$. Let $a, b \in \{0, \dots, 2^i - 1\}$, which we regard as assignments to the $i$ least significant bits of each of the two factors of $n$-bit multiplication. Then, for any $x, y \in \{0, \dots, 2^{n-i} - 1\}$, regarded as assignments to the remaining bits of the factors,*

$$(\mathrm{MUL}_{n-1,n})|_{a,b}(x, y) = [(x2^i + a) \cdot (y2^i + b)]_{n-1} = \left[[a]_0^{n-i-1} y + [b]_0^{n-i-1} x + [ab]_i^{n-1}\right]_{n-i-1}.$$

For the sake of completeness, we include the simple proof.

**Proof.** First, using that $i \geq n/2$ and Proposition 4, we get:

$$\begin{aligned}
[(x2^i + a) \cdot (y2^i + b)]_i^{n-1} &= \left(\left(xy2^{2i} + (ay + bx)2^i + ab\right) \bmod 2^n\right) \operatorname{div} 2^i \\
&\overset{i \geq n/2}{=} \left(\left((ay + bx)2^i + ab\right) \bmod 2^n\right) \operatorname{div} 2^i \\
&\overset{\text{Proposition 4},(1)+(3)}{=} \left(ay + bx + (ab) \operatorname{div} 2^i\right) \bmod 2^{n-i} \\
&\overset{\text{Proposition 4},(1)}{=} \left([a]_0^{n-i-1} y + [b]_0^{n-i-1} x + [ab]_i^{n-1}\right) \bmod 2^{n-i}.
\end{aligned}$$

Hence, again by Proposition 4, part (1),

$$\begin{aligned}
[(x2^i + a) \cdot (y2^i + b)]_{n-1} &= \left(\left([a]_0^{n-i-1} y + [b]_0^{n-i-1} x + [ab]_i^{n-1}\right) \bmod 2^{n-i}\right) \operatorname{div} 2^{n-i-1} \\
&= \left[[a]_0^{n-i-1} y + [b]_0^{n-i-1} x + [ab]_i^{n-1}\right]_{n-i-1}. \quad \square
\end{aligned}$$

Due to this proposition, the number of subfunctions of $\mathrm{MUL}_{n-1,n}$ obtained by fixing the variables $x_0, y_0, \ldots, x_{i-1}, y_{i-1}$ in an arbitrary way is upper bounded by the number of triples $([a]_0^{n-i-1}, [b]_0^{n-i-1}, [ab]_i^{n-1})$ for assignments $a, b \in \{0, 1\}^i$. This clearly gives an upper bound on the size of the respective level of nodes in an OBDD and even quasireduced OBDD. Following the outline at the beginning, here we want a *lower bound* on the number subfunctions in terms of the number of triples. This is exactly what our first main lemma provides.

For what follows, let $n/2 \le i \le n-1$. For $a, b, c \in \{0, \ldots, 2^{n-i}-1\}$ and $x, y \in \{0, \ldots, 2^{n-i}-1\}$ define $f_{a,b,c}(x, y) := [ay + bx + c]_{n-i-1}$.

**Main Lemma 1.** *Let $n/2 \le i \le n-1$. Let $a, b, c, a', b', c' \in \{0, \ldots, 2^{n-i}-1\}$ be given with $a, b, a', b'$ odd and $(a, b, c) \ne (a', b', c')$. Then $f_{a,b,c} = f_{a',b',c'}$ implies $a' \equiv -a \bmod 2^{n-i}$, $b' \equiv -b \bmod 2^{n-i}$ and $c + c' \equiv 2^{n-i-1} - 1 \bmod 2^{n-i}$.*

For $a, b \in \{0, 1\}^i$ define $g(a, b) := \left([a]_0^{n-i-1}, [b]_0^{n-i-1}, [ab]_i^{n-1}\right) \in \{0, 1\}^{3(n-i)}$.

**Corollary 1.** *Let $S \subseteq \{(a, b) \mid a, b \in \{0, \ldots, 2^i - 1\}, a, b \text{ odd}\}$. Then the number of different subfunctions $(\mathrm{MUL}_{n-1,n})|_{a,b}$ with $(a, b) \in S$, where $a, b$ are regarded as assignments to the $i$ least significant bits of the two factors of $n$-bit multiplication, is at least $|g(S)|/2$.*

**Proof.** Let $(a, b) \in S$ and $x, y \in \{0, \ldots, 2^{n-i}-1\}$. By Proposition 6 and the definitions,

$$(\mathrm{MUL}_{n-1,n})|_{a,b}(x, y) = \left[[a]_0^{n-i-1} y + [b]_0^{n-i-1} x + [ab]_i^{n-1}\right]_{n-i-1} = f_{\widetilde{a},\widetilde{b},\widetilde{c}}(x, y),$$

with $\widetilde{a} := [a]_0^{n-i-1}$, $\widetilde{b} := [b]_0^{n-i-1}$, and $\widetilde{c} := [ab]_i^{n-1}$. By Main Lemma 1, there are at most two different triples $(\widetilde{a}, \widetilde{b}, \widetilde{c})$ that yield the same function $f_{\widetilde{a},\widetilde{b},\widetilde{c}}$. Hence, the number of considered subfunctions of $\mathrm{MUL}_{n-1,n}$ is at least half the number of different triples $(\widetilde{a}, \widetilde{b}, \widetilde{c})$ belonging to $(a, b) \in S$, and the latter is exactly $|g(S)|$. $\square$

Given this lemma and its corollary, we know that in order to lower bound the number of subfunctions obtained by fixing $x_0, y_0, \ldots, x_{i-1}, y_{i-1}$, it suffices to count the number of triples $\left([a]_0^{n-i-1}, [b]_0^{n-i-1}, [ab]_i^{n-1}\right) \in \{0, 1\}^{3(n-i)}$. For this, we look at the multiplications in the third components as hash functions, as indicated in the outline at the beginning. Let

$$U := \{(x, y) \mid x, y \in \{0, \ldots, 2^{2i-n} - 1\}\}.$$

For $a, b \in \{0, \ldots, 2^{n-i}-1\}$ and $(x, y) \in U$, define $h_{a,b}\colon U \to \{0, \ldots, 2^{n-i}-1\}$ by

$$h_{a,b}(x, y) := [(x2^{n-i} + a) \cdot (y2^{n-i} + b)]_i^{n-1}.$$

We then show that there is a constant fraction of all possible $(a, b)$ (the settings to $[a]_0^{n-i-1}$, $[b]_0^{n-i-1}$ in our original problem) for which the size of the range of the hash function $h_{a,b}$ is a constant fraction of all possible values (the number of different $[ab]_i^{n-1}$ in our original problem).

**Main Lemma 2.** *Let $(3/5)n \le i \le (2/3)n$. Let $\alpha \in (0, 1]$. Then there is a set $A \subseteq \{(a, b) \mid a, b \in \{0, \ldots, 2^{n-i}-1\}, a, b \text{ odd}\}$ with $|A| \ge (1-\alpha) \cdot 2^{2(n-i-1)}$ such that for all $(a, b) \in A$, $|h_{a,b}(U)| \ge (\alpha/(1+\alpha)) \cdot 2^{n-i}$.*

Corollary 1 and Main Lemma 2 together yield our main result.

**Proof of Theorem 3.** We choose $i := \lceil (3/5)n \rceil$ and consider the subfunctions of $\mathrm{MUL}_{n-1,n}$ obtained by fixing $x_0, y_0, \ldots, x_{i-1}, y_{i-1}$. By Lemma 5 and Corollary 1, the number of these subfunctions is at least half the number of different triples $\left([a]_0^{n-i-1}, [b]_0^{n-i-1}, [ab]_i^{n-1}\right)$ with $a, b \in \{0, \ldots, 2^i - 1\}$ odd. Using Main Lemma 2, the number of subfunctions can thus be lower bounded by

$$\frac{1}{2}(1-\alpha)2^{2(n-i-1)} \cdot \frac{\alpha}{1+\alpha} 2^{n-i} = (1-\alpha)\frac{\alpha}{1+\alpha}\frac{1}{8}2^{3(n-i)}$$

for any $\alpha \in (0, 1]$. We maximize the function $f(\alpha) := (1-\alpha) \cdot \alpha/(1+\alpha)$ for $\alpha \in (0, 1]$ by choosing $\alpha := \sqrt{2} - 1$, which gives $f(\alpha) = 3 - 2\sqrt{2}$ and thus a lower bound on the number of subfunctions of

$$\frac{3 - 2\sqrt{2}}{8} 2^{3(n-i)}.$$

Applying Lemma 5 and substituting $i = \lceil (3/5)n \rceil$, we get that the size of the OBDD is at least

$$\frac{3 - 2\sqrt{2}}{4} 2^{3\lfloor (2/5)n \rfloor} - 1. \quad \square$$

The constant in front of the term of the largest order in the lower bound can be improved by summing the sizes of individual levels in the OBDD consisting of nodes labeled by the same variable. Given that this only yields small improvements of the bound, we refrain from carrying out the details.

## 4. Proof of Main Lemma 1

In this and the next section, we prove the two main lemmas that we have stated and already applied in the previous section.

**Proof of Main Lemma 1.** Throughout the proof, we simply write "$\equiv$" for equivalence modulo $2^{n-i}$. Let $a, b, c, a', b', c' \in \{0, \dots, 2^{n-i} - 1\}$ with $a, b, a', b'$ odd be given such that $f_{a,b,c} = f_{a',b',c'}$. We show that then either $a \equiv a'$, $b \equiv b'$, and $c \equiv c'$ (implying that even $a = a'$, $b = b'$ and $c = c'$) or $a' \equiv -a$, $b' \equiv -b$, and $c + c' \equiv 2^{n-i-1} - 1$.

For any $u, v, w \in \{0, \dots, 2^{n-i} - 1\}$, let $\widetilde{f}_{u,v,w}(x, y) = (uy + vx + w) \bmod 2^{n-i}$. Let $N := 2^{n-i-1}$ and $L := \{0, \dots, N - 1\}$. We observe that

$$f_{u,v,w}(x, y) = [(uy + vx + w) \bmod 2^{n-i}]_{n-i-1} = 0 \Leftrightarrow \widetilde{f}_{u,v,w}(x, y) \in L.$$

Due to the assumption that $f_{a,b,c} = f_{a',b',c'}$, for any $x, y \in \{0, \dots, 2^{n-i} - 1\}$,

$$\widetilde{f}_{a,b,c}(x, y) \in L \Leftrightarrow \widetilde{f}_{a',b',c'}(x, y) \in L. \tag{1}$$

In what follows, we apply this fact for $y = 0$ and show that either $b \equiv b'$ and $c \equiv c'$ or $b \equiv -b'$ and $c + c' \equiv 2^{n-i-1} - 1$. We get an analogous conclusion for $a, a'$ instead of $b, b'$ by working with $x = 0$ instead of $y = 0$. Since $c + c' \equiv 2^{n-i-1} - 1$ implies that $c \not\equiv c'$, this altogether proves the lemma.

Since $b$ is odd and thus $\gcd(b, 2^{n-i}) = 1$, its multiplicative inverse $b^{-1}$ in $\mathbb{Z}_{2^{n-i}}$ exists. For $j = 0, \dots, N - 1$, we can therefore define $x_j := (b^{-1}(j - c)) \bmod 2^{n-i}$. Then

$$\widetilde{f}_{a,b,c}(x_j, 0) = (bx_j + c) \bmod 2^{n-i} \equiv j \in L, \quad \text{for } j = 0, \dots, N - 1. \tag{2}$$

Furthermore, we have $b' \equiv b\widetilde{b}$ for $\widetilde{b} := (b^{-1}b') \bmod 2^{n-i}$, which is again an odd number. Hence, setting $\widetilde{c} := c' - c\widetilde{b}$, we get

$$b'x_j + c' \equiv b\widetilde{b}x_j + \widetilde{c} + c\widetilde{b} = (bx_j + c)\widetilde{b} + \widetilde{c} = j \cdot \widetilde{b} + \widetilde{c}$$

and thus

$$\widetilde{f}_{a',b',c'}(x_j, 0) = (b'x_j + c') \bmod 2^{n-i} \equiv j \cdot \widetilde{b} + \widetilde{c}, \quad \text{for } j = 0, \dots, N - 1.$$

Furthermore, due to (1) and (2),

$$(j \cdot \widetilde{b} + \widetilde{c}) \bmod 2^{n-i} \in L, \quad \text{for } j = 0, \dots, N - 1. \tag{3}$$

In particular, $(0 \cdot \widetilde{b} + \widetilde{c}) \bmod 2^{n-i} = \widetilde{c} \bmod 2^{n-i} \in L$. Hence, there is a unique $k \in \mathbb{Z}$ such that $\widetilde{c} - k2^{n-i} \in L$. We now distinguish the following two cases.

*Case* 1, $\widetilde{b} \le 2^{n-i-1}$: We first prove by induction on $j$ that

$$(j \cdot \widetilde{b} + \widetilde{c}) \bmod 2^{n-i} = j \cdot \widetilde{b} + \widetilde{c} - k2^{n-i}, \quad \text{for } j = 0, \dots, N - 1. \tag{4}$$

By the preceding remarks, we have $(0 \cdot \widetilde{b} + \widetilde{c}) \bmod 2^{n-i} = \widetilde{c} - k2^{n-i}$. Now suppose that, for some $j \in \{0, \dots, N - 1\}$, $(j \cdot \widetilde{b} + \widetilde{c}) \bmod 2^{n-i} = j \cdot \widetilde{b} + \widetilde{c} - k2^{n-i}$. We additionally know from (3) that $j \cdot \widetilde{b} + \widetilde{c} - k2^{n-i} \in L$. Using the assumption of Case 1, it follows that $(j + 1)\widetilde{b} + \widetilde{c} - k2^{n-i} \in \{0, \dots, 2^{n-i} - 1\}$ and thus $((j + 1)\widetilde{b} + \widetilde{c}) \bmod 2^{n-i} = (j + 1)\widetilde{b} + \widetilde{c} - k2^{n-i}$, completing the proof of (4).

Observe that $\widetilde{b}$ is an integer with $\widetilde{b} \ge 1$. Using (4), it follows that the mapping $j \mapsto (j \cdot \widetilde{b} + \widetilde{c}) \bmod 2^{n-i}$ is strictly increasing for $j = 0, \dots, N - 1$. Additionally, its image on the domain $L = \{0, \dots, N - 1\}$ is contained in the set $L$. This is only possible for $\widetilde{b} = 1$ and $0 = \widetilde{c} = c' - c\widetilde{b} = c' - c$.

*Case* 2, $\widetilde{b} > 2^{n-i-1}$: We have $\widetilde{b} \equiv -\overline{b} \bmod 2^{n-i}$ for $\overline{b} = 2^{n-i} - \widetilde{b}$. We observe that $0 \le \overline{b} < 2^{n-i-1}$ is an odd integer. Analogously to the first case, we prove by induction that

$$(j \cdot (-\overline{b}) + \widetilde{c}) \bmod 2^{n-i} \equiv j \cdot (-\overline{b}) + \widetilde{c} - k2^{n-i}, \quad \text{for } j = 0, \dots, N - 1. \tag{5}$$

As in the first case, this is satisfied for $j = 0$, since $(0 \cdot (-\overline{b}) + \widetilde{c}) \bmod 2^{n-i} = \widetilde{c} - k2^{n-i}$. Now suppose that the above holds for some $j \in \{0, \dots, N - 1\}$. Since $j \cdot (-\overline{b}) + \widetilde{c} - k2^{n-i} \in L$, it follows that

$$(j + 1)(-\overline{b}) + \widetilde{c} - k2^{n-i} = j \cdot (-\overline{b}) + \widetilde{c} - k2^{n-i} - \overline{b} \in \{-2^{n-i-1} + 1, \dots, 2^{n-i-1} - 1\}.$$

Hence, either $((j + 1)(-\overline{b}) + \widetilde{c}) \bmod 2^{n-i} \notin L$ or

$$((j + 1)(-\overline{b}) + \widetilde{c}) \bmod 2^{n-i} = (j + 1)(-\overline{b}) + \widetilde{c} - k2^{n-i}.$$

Since $(j(-\overline{b}) + \widetilde{c}) \bmod 2^{n-i} \in L$ for all $j = 0, \dots, N - 1$, the former case can only occur for $j = N - 1$. Hence, we have proved (5).

Since $\overline{b} \geq 1$, it follows from (5) that the mapping $j \mapsto (j(-\overline{b}) + \widetilde{c}) \bmod 2^{n-i}$ is strictly decreasing for $j \in L$, with its image on the domain $L$ contained in $L$. Hence, $\widetilde{b} \equiv -\overline{b} = -1$ and $\widetilde{c} \equiv 2^{n-i-1} - 1 \equiv c' - c\widetilde{b} \equiv c' + c$. $\square$

## 5. Proof of Main Lemma 2

We complete the proof of the main result by proving the second main lemma. Recall the following definitions from Section 3. We let

$$U := \{(x, y) \mid x, y \in \{0, \ldots, 2^m - 1\}\}$$

and for $a, b \in \{0, \ldots, 2^{n-i} - 1\}$, $h_{a,b}: U \to \{0, \ldots, 2^{n-i} - 1\}$ is defined for $(x, y) \in U$ by

$$h_{a,b}(x, y) \ := \ [(x2^{n-i} + a) \cdot (y2^{n-i} + b)]_i^{n-1}.$$

We restate the second main lemma for the convenience of the reader.

**Main Lemma 2** (*Restatement*). *Let $(3/5)n \leq i \leq (2/3)n$. Let $\alpha \in (0, 1]$. Then there is a set $A \subseteq \{(a, b) \mid a, b \in \{0, \ldots, 2^{n-i} - 1\}, a, b \text{ odd}\}$ with $|A| \geq (1 - \alpha) \cdot 2^{2(n-i-1)}$ such that for all $(a, b) \in A$, $|h_{a,b}(U)| \geq (\alpha/(1 + \alpha)) \cdot 2^{n-i}$.*

For what follows, it first suffices to consider values of $i$ with $n/2 \leq i \leq n - 1$. Furthermore, we define $m := 2i - n$. As the first step of the proof of the lemma, we rewrite the product occurring in the definition of $h_{a,b}(x, y)$ in a more suitable way.

**Proposition 7.** *For $a, b \in \{0, \ldots, 2^{n-i} - 1\}$ and $x, y \in \{0, \ldots, 2^m - 1\}$,*

$$[(x2^{n-i} + a) \cdot (y2^{n-i} + b)]_i^{n-1} = \left(\left(xy2^{n-i} + ay + bx + (ab) \operatorname{div} 2^{n-i}\right) \bmod 2^i\right) \operatorname{div} 2^m.$$

**Proof.** The proof is similar to that of Proposition 6. To get a high-level idea why the proposition is correct, we first expand the product:

$$[(x2^{n-i} + a) \cdot (y2^{n-i} + b)]_i^{n-1} = \left[xy2^{2(n-i)} + (ay + bx)2^{n-i} + ab\right]_i^{n-1}.$$

Now observe that the least significant $n - i$ bits in the first two summands within the brackets are zero and thus no carry can be generated together with the corresponding bits of the last summand. Furthermore, the brackets remove the least significant $i$ bits of the result and $i \geq n - i$ (since by assumption $i \geq n/2$). Hence, the least significant $n - i$ bits of all three summands are irrelevant for the overall result and each of these summands can be "shifted right" by $n - i$ places using a "div". It remains to adjust the range of bits selected by the brackets to reflect this shift and finally to replace the brackets by appropriate "div" and "mod" operations.

More formally,

$$\begin{aligned}
[(x2^{n-i} + a) \cdot (y2^{n-i} + b)]_i^{n-1} &= \left(\left(xy2^{2(n-i)} + (ay + bx)2^{n-i} + ab\right) \bmod 2^n\right) \operatorname{div} 2^i \\
&\overset{\text{Proposition 4, (2)}}{=} \left(\left(2^{n-i}\left((xy2^{n-i} + ay + bx) \bmod 2^i\right) + ab\right) \bmod 2^n\right) \operatorname{div} 2^i \\
&\overset{\text{Proposition 4, (1)}}{=} \left(\left(2^{n-i}\left((xy2^{n-i} + ay + bx) \bmod 2^i\right) + ab\right) \operatorname{div} 2^i\right) \bmod 2^{n-i} \\
&\overset{\text{Proposition 4, (3)}}{=} \left(\left((xy2^{n-i} + ay + bx) \bmod 2^i + (ab) \operatorname{div} 2^{n-i}\right) \operatorname{div} 2^m\right) \bmod 2^{n-i}.
\end{aligned}$$

For the last line, we have used that $i \geq n - i$. Due to the fact that $(ab) \operatorname{div} 2^{n-i} < 2^{n-i} \leq 2^i$, we can rewrite the last line as

$$\left[\left(\left(xy2^{n-i} + ay + bx + (ab) \operatorname{div} 2^{n-i}\right) \bmod 2^i\right) \operatorname{div} 2^m\right] \bmod 2^{n-i}.$$

Since the term in the brackets is smaller than $2^{i-m} = 2^{n-i}$, the outermost "mod" operation can be removed, giving the desired result. $\square$

By Proposition 7, we thus have

$$h_{a,b}(x, y) = \left((xy \cdot 2^{n-i} + ay + bx + (ab) \operatorname{div} 2^{n-i}) \bmod 2^i\right) \operatorname{div} 2^m.$$

Let $\mathcal{H} := \{h_{a,b} \mid a, b \in \{0, \ldots, 2^{n-i} - 1\}, a, b \text{ odd}\}$.

Our aim is to show that a constant fraction of the functions in $\mathcal{H}$ have a range whose size is a constant fraction of the number of all possible values. We do this indirectly by using a variant of Woelfel's technique [10] based on universal hashing. The key observation is that a function $h \in \mathcal{H}$ has a large range if it has a small number of *collisions*, i.e., pairs of different arguments from $U$ mapped to the same value.

In the construction of [10], output bits of multiplication with index equal to or below the middle bit are used. The respective class of hash functions considered there is *universal*, meaning that for each pair of different arguments from the universe, a random function of this class induces a collision on the chosen pair only with probability $1/r$, $r$ the size of the range of the hash functions.

Our hash functions in $\mathcal{H}$ compute the output bits of the $i$-bit multiplication of $(a, x)$ and $(b, y)$, $a, b \in \{0, \ldots, 2^{n-i} - 1\}$ and $x, y \in \{0, \ldots, 2^m - 1\}$, with indices $i, \ldots, n-1$, i.e., *above* the middle bit. Different from the hash class of [10], this class is not universal. The probability of a random hash function inducing a collision cannot even be bounded by $c/r$, $c \geq 1$ a constant and $r = 2^{n-i}$ the size of the range of the hash functions (consider, e.g., the input pairs $(x, y) := (0, 0)$ and $(x', y') := (0, 1)$, for which $h_{a,b}$ with any $a < 2^{m-1}$ and an arbitrary $b$ induces a collision). It is not obvious however whether choosing a smaller universe could not repair this.

We do not try to enforce universality of the considered class of hash functions. Instead, we observe that it is in fact sufficient for our purposes that the average number of collisions over all functions from $\mathcal{H}$ is small.

**Definition 8.** Let $\mathcal{H}$ be a class of functions $U \to R$, $U$ and $R$ finite. For $h \in \mathcal{H}$, define its *collision number* $c(h)$ as the number of pairs of different values from $U$ that are mapped to the same value by $h$. Let $c(\mathcal{H})$, the *average number of collisions of* $\mathcal{H}$, be defined by $c(\mathcal{H}) := (1/|\mathcal{H}|) \sum_{h \in \mathcal{H}} c(h)$.

Obviously, we always have the trivial bound $c(\mathcal{H}) \leq \binom{|U|}{2} = \Theta(|U|^2)$. It turns out that in order to prove our second main lemma, we need a much better upper bound, namely of order $|U|$. The key ingredients for proving such a bound are summarized in the following lemma, in which we investigate the conditions under which different elements from $U$ can collide under a given function from $\mathcal{H}$. The second part of this is a weaker form of the universality property of hash classes, while the additional first part is required here to get a sufficiently good bound.

**Lemma 9.** *Let $n/2 \leq i \leq (2/3)n$. For $k \in \{0, \ldots, m-1\}$ let $c, d$ be such that $|c|, |d| \in \{0, \ldots, 2^{m-k} - 1\}$ and such that $c$ is odd.*

(1) *Let $a, b \in \{0, \ldots, 2^{n-i} - 1\}$, $a, b$ odd. Let $x, x' \in \{0, \ldots, 2^m - 1\}$ with $x' - x = c2^k$. Then there are at most $2^k$ values $y \in \{0, \ldots, 2^m - 1\}$ such that for $y' := y + d2^k$ with $y' \in \{0, \ldots, 2^m - 1\}$, $h_{a,b}(x, y) = h_{a,b}(x', y')$.*
(2) *Let $(x, y), (x', y') \in U$ with $x' - x = c2^k$ and $y' - y = d2^k$. Then at most a fraction of $2^{-(2n-3i+k)}$ of the functions in $\mathcal{H}$ map $(x, y)$ and $(x', y')$ to the same value.*

*The same holds if the roles of $x, x'$ and $y, y'$ are exchanged.*

**Proof.** The statement at the end of the lemma obviously follows from the symmetry of the definition of the functions in $\mathcal{H}$.

*Part* (1): Let a $y \in \{0, \ldots, 2^m - 1\}$ be given and let $y' := y + d2^k \in \{0, \ldots, 2^m - 1\}$. By the definition of $h_{a,b}$, Propositions 4 and 7, part (4), $h_{a,b}(x, y) = h_{a,b}(x', y')$ implies that there is an integer $e$ with $|e| < 2^m$ such that

$$(x'y' - xy) \cdot 2^{n-i} + a(y' - y) + b(x' - x) \equiv e \bmod 2^i. \qquad (*)$$

W.l.o.g. (by swapping $x, y$ with $x', y'$), we may even assume that $e \geq 0$. Using that $x' = x + c2^k$ and $y' = y + d2^k$, we get

$$\begin{aligned}(x'y' - xy) \cdot 2^{n-i} + a(y' - y) + b(x' - x) &= (\mathrm{d}x2^k + cy2^k + cd2^{2k})\, 2^{n-i} + ad2^k + bc2^k \\ &= cy2^{n-i+k} + \mathrm{d}x2^{n-i+k} + r,\end{aligned}$$

with $r := ad2^k + bc2^k + cd2^{n-i+2k}$.

Hence, $(*)$ is equivalent to

$$cy2^{n-i+k} + \mathrm{d}x2^{n-i+k} \equiv e - r \bmod 2^i.$$

We observe that the left hand side of this congruence is divisible by $2^{n-i+k}$ and that, since $k < m = 2i - n$, we have $n - i + k < n - i + m = i$. Therefore, the congruence can only hold if also $e - r$ is divisible by $2^{n-i+k}$. Since $0 \leq e < 2^m$ and $m \leq n - i$ by the assumption that $i \leq (2/3)n$, $e - r \equiv 0 \bmod 2^{n-i+k}$ implies that the value of $e$ is fixed given $a, b, c, d$ and $k$ and thus $r$.

Assuming that $e - r \equiv 0 \bmod 2^{n-i+k}$, the above congruence is equivalent to

$$cy + \mathrm{d}x \equiv (e - r)2^{-(n-i+k)} \bmod 2^{m-k}.$$

Since $c$ is odd and thus $\gcd(c, 2^{m-k}) = 1$, the multiplicative inverse $c^{-1}$ of $c$ in $\mathbb{Z}_{2^{m-k}}$ exists and we can solve the above for $y$, getting

$$y \equiv c^{-1}((e - r)2^{-(n-i+k)} - \mathrm{d}x) \bmod 2^{m-k}.$$

Since $y \in \{0, \ldots, 2^m - 1\}$ and $e$ are already fixed by choosing $a, b, c, d$ and $k$, this means that there are at most $2^k$ suitable values for $y$. Hence, for a fixed $a, b$ and given the distances $x' - x = c2^k$ and $y' - y = d2^k$, there are at most $2^k$ choices for $y$ such that $(x, y)$ and $(x', y')$ collide under $h_{a,b}$.

*Part* (2): By the assumptions for this part, $x' - x = c2^k$ and $y' - y = d2^k$ for $k \in \{0, \ldots, m-1\}$, $c \in \mathbb{Z}^*_{2^{m-k}}$, and $d \in \mathbb{Z}_{2^{m-k}}$. In the proof of the first part, is has been shown that $h_{a,b}(x, y) = h_{a,b}(x', y')$ only if

$$r = ad2^k + bc2^k + cd2^{n-i+2k} \equiv e \bmod 2^{n-i+k}$$

for some $e$ with $0 \le e < 2^m$. Since

$$\left(ad2^k + bc2^k + cd2^{n-i+2k}\right) \bmod 2^{n-i+k} = \left((ad+bc) \bmod 2^{n-i}\right) \cdot 2^k,$$

applying part (5) of Proposition 4 yields that this is equivalent to

$$\left(\left((ad+bc) \bmod 2^{n-i}\right) \cdot 2^k\right) \operatorname{div} 2^m = \left((ad+bc) \operatorname{div} 2^{m-k}\right) \bmod 2^{n-i-(m-k)} = 0.$$

By the assumption $i \le (2/3)n$, we have $n-i-(m-k) \ge 0$. Since the statement of part (2) is trivially true if $i = (2/3)n$ and $k = 0$, we may also assume that $n-i-(m-k) > 0$. Let $a = s2^{m-k}+t$ and $b = u2^{m-k}+v$ with $s, u \in \{0, \dots, 2^{n-i-(m-k)}-1\}$, $t, v \in \{0, \dots, 2^{m-k}-1\}$, $t, v$ odd. Then the above is equivalent to

$$ds + cu + r' \equiv 0 \bmod 2^{n-i-(m-k)},$$

with $r' := (dt + cv) \operatorname{div} 2^{m-k}$. Since $\gcd(c, 2^{n-i-(m-k)}) = 1$, $c^{-1} \in \mathbb{Z}^*_{2^{n-i-(m-k)}}$ exists and we can solve the congruence for $u$, giving

$$u \equiv c^{-1}(-ds - r') \bmod 2^{n-i-(m-k)}.$$

Thus, given $c, d, s, t, v$ and $k$, $u$ is completely fixed. The fraction of functions in $\mathcal{H}$ inducing a collision for two different keys can thus be upper bounded by $2^{-(n-i-(m-k))} = 2^{-(2n-3i+k)}$. $\square$

We now apply the previous lemma to bound the average number of collisions of the class $\mathcal{H}$.

**Lemma 10.** *Let $n/2 \le i \le (2/3)n$. Then $c(\mathcal{H}) \le 2^{9i-5n} - 2^{5i-3n}$.*

**Proof.** We obtain the average number of collisions of $\mathcal{H}$ by summing over all pairs of different arguments $(x, y), (x', y') \in U$ the fraction of functions that map these keys to the same value. We first take a closer look at the pairs of arguments over which this sum extends. If $(x, y) \ne (x', y')$, then $x' - x \ne 0$ or $y' - y \ne 0$. It follows that there is a $k \in \{0, \dots, m-1\}$ such that $2^k = \gcd(x'-x, y'-y, 2^m)$. Furthermore, there are $c, d \in \mathbb{Z}_{2^{m-k}}$, where at least one of the numbers $c, d$ is odd, such that $x' - x = c2^k$ and $y' - y = d2^k$. Now let

$$\alpha := \sum_{k=0}^{m-1} \sum_{c \in \mathbb{Z}^*_{2^{m-k}}} \sum_{\substack{x,x' \in \mathbb{Z}_{2^m},\\ |x-x'|=c2^k}} \sum_{d \in \mathbb{Z}_{2^{m-k}}} \sum_{\substack{y,y' \in \mathbb{Z}_{2^m},\\ |y-y'|=d2^k}} \frac{1}{|\mathcal{H}|} \sum_{a,b \in \mathbb{Z}^*_{2^{n-i}}} [h_{a,b}(x,y) = h_{a,b}(x',y')],$$

$$\beta := \sum_{k=0}^{m-1} \sum_{\substack{c \in \mathbb{Z}_{2^{m-k}}\\ \text{even}}} \sum_{\substack{x,x' \in \mathbb{Z}_{2^m},\\ |x-x'|=c2^k}} \sum_{d \in \mathbb{Z}^*_{2^{m-k}}} \sum_{\substack{y,y' \in \mathbb{Z}_{2^m},\\ |y-y'|=d2^k}} \frac{1}{|\mathcal{H}|} \sum_{a,b \in \mathbb{Z}^*_{2^{n-i}}} [h_{a,b}(x,y) = h_{a,b}(x',y')].$$

Then $c(\mathcal{H}) = \alpha + \beta$.

First, we know from Lemma 9, part (2), that the innermost sum of $\alpha$ including the factor $1/|\mathcal{H}|$ can be upper bounded by $2^{-(2n-3i+k)}$. Thus, it remains to count the number of different pairs of keys $(x, y)$ and $(x', y')$ for which this innermost sum is nonzero. By part (1) of Lemma 9, the number of summands over which the second to last sum needs to be extended is at most $2^k$. The number of summands of the third to last sum is obviously $2^{m-k}$. Finally, we count the number of suitable $c$, $x$, and $x'$:

$$\sum_{c \in \mathbb{Z}^*_{2^{m-k}}} \sum_{\substack{x,x' \in \mathbb{Z}_{2^m},\\ |x-x'|=c2^k}} 1 = \sum_{c \in \mathbb{Z}^*_{2^{m-k}}} 2 \cdot \sum_{0 \le x \le 2^m - c2^k - 1} 1 = \sum_{c \in \mathbb{Z}^*_{2^{m-k}}} 2 \cdot (2^m - c2^k)$$

$$= 2^{2m-k} - 2^{k+1} \cdot \sum_{j=0}^{2^{m-k-1}-1} (2j+1) = 2^{2m-k} - 2^{k+1} \cdot 2^{2(m-k-1)} = 2^{2m-k-1}.$$

Putting everything together, we get

$$\alpha \le \sum_{k=0}^{m-1} 2^{2m-k-1} \cdot 2^{m-k} \cdot 2^k \cdot 2^{-(2n-3i+k)} = 2^{3m+3i-2n-1} \sum_{k=0}^{m-1} 2^{-2k}$$

$$= 2^{3m+3i-2n-1} \cdot \frac{4}{3}\left(1 - 2^{-2m}\right) = \frac{2}{3}\left(2^{9i-5n} - 2^{5i-3n}\right).$$

By symmetry, it follows that $\beta = \alpha/2$ and thus $c(\mathcal{H}) = (3/2)\alpha$, which altogether gives the desired upper bound on $c(\mathcal{H})$. $\square$

Finally, we use the following fact implicit in the proof of Lemma 8 in [10].

**Lemma 11** (*[10]*). *Let $\mathcal{H}$ be a class of functions $U \to R$, $U$ and $R$ finite, and let $h \in \mathcal{H}$ be given with collision number $c = c(h)$. Then $|h(U)| \geq |U|^2/(c + |U|)$.*

For the sake of completeness, we include the easy proof.

**Proof.** We count the number of collisions by summing over all values in the range of $h$. Let $r := |h(U)|$ and $u := |U|$.

$$c := c(h) = \sum_{y \in h(U)} \sum_{\substack{x,x' \in U,\\ x \neq x'}} [h(x) = h(x') = y] = \sum_{y \in h(U)} |h^{-1}(y)|(|h^{-1}(y)| - 1)$$
$$= \sum_{y \in h(U)} \left(|h^{-1}(y)|^2 - |h^{-1}(y)|\right) = -u + \sum_{y \in h(U)} |h^{-1}(y)|^2.$$

We lower bound the sum by minimizing the function $f : \mathbb{R}^r \to \mathbb{R}$ with $f(x_1, \ldots, x_r) := \sum_{j=1}^{r} x_i^2$ under the constraint $x_1 + \cdots + x_r = u$. Due to symmetry of the summands and the fact that each is monotonously increasing, it follows that the minimum is attained for $x_1 = \cdots = x_r = u/r$. Thus, we have

$$c \geq -u + r(u/r)^2 = u^2/r - u,$$

which after rearranging gives the claimed lower bound on $r$,

$$r \geq \frac{u^2}{c + u}. \quad \square$$

**Proof of Main Lemma 2.** By Lemma 10, we have

$$c(\mathcal{H}) = E_h(c(h)) \leq 2^{9i-5n} =: c,$$

where the expectation is with respect to uniformly random $h \in \mathcal{H}$. For $\alpha \in (0, 1]$ let

$$A := \{(a, b) \mid a, b \in \{0, \ldots, 2^{n-i} - 1\}, a, b \text{ odd}, c(h_{a,b}) \leq \alpha^{-1} c\}.$$

By Markov's inequality, $|A| \geq (1 - \alpha)|U|$. Let $(a, b) \in A$. By Lemma 11,

$$|h_{a,b}(U)| \geq \frac{2^{4m}}{\alpha^{-1} 2^{9i-5n} + 2^{2m}}.$$

Since $i \geq (3/5)n$ by assumption, it follows that $9i - 5n \geq 2m = 4i - 2n$. Hence, we can lower bound the right hand side above by

$$\frac{2^{4m}}{(\alpha^{-1} + 1)2^{9i-5n}} = \frac{\alpha}{1 + \alpha} 2^{n-i}.$$

This proves the lemma. $\square$

## 6. Conclusion

We conclude with a comparison of the OBDD sizes of $\mathrm{MUL}_{n-1,n}$ for the optimal order with the so far best known structured variable orders (i.e., variables orders with a simple closed description for all $n$). For each sufficiently small $n$ we consider the following orders (see Table 1):

- *Optimal variable order*, $\pi_{\mathrm{opt}}$: By this, we mean one specific optimal variable order that has been the first found by our implementation of the dynamic programming approach of Friedman and Supowit ([6], see also [8], Section 5.5).
- *Pairwise ascending order*, $\pi_{\mathrm{pwa}}$: $x_0, y_0, \ldots, x_{n-1}, y_{n-1}$.
- *Hybrid order*, $\pi_{\mathrm{hyb}}$: It has already been observed by Amano and Maruoka [1] that all the optimal orders found by dynamic programming for small input lengths end with the group of variables $x_0, y_{n-1}, x_{n-1}, y_0$ (this is true for both quasireduced and fully reduced OBDDs). Taking this a step further and trying to mimic other structures observed in these orders, we arrive at the following hybrid order: For $m := \min(n - 2, \lceil (n + 1)/2 \rceil)$, this is the order $x_m, \ldots, x_1, y_m, \ldots, y_1, x_{m+1}, y_{m+1}, \ldots, x_{n-2}, y_{n-2}, x_0, y_{n-1}, x_{n-1}, y_0$.

Optimal variable orders for $\mathrm{MUL}_{n-1,n}$ and $n \leq 12$ can be found under http://ls2-www.cs.uni-dortmund.de/~sauerhof/midbit_orders.

Not surprisingly, the fine-tuned hybrid order even beats the pairwise ascending variable order for most $n$. Nevertheless, this order resembles the pairwise ascending order from around index $n/2$ upward, apart from a constant number of variables at the end, which by the arguments in the main part of the paper implies that also this order leads to OBDDs of size $\Omega(2^{(6/5)n})$.

**Table 1**
OBDD sizes of $\mathrm{MUL}_{n-1,n}$ for different variable orders.

| $n$: | $\pi_{\mathrm{opt}}$: | $\pi_{\mathrm{pwa}}$: | $\pi_{\mathrm{hyb}}$: |
|---|---|---|---|
| 2 | 8 | 9 | 8 |
| 3 | 14 | 16 | 14 |
| 4 | 31 | 36 | 31 |
| 5 | 63 | 73 | 64 |
| 6 | 136 | 169 | 175 |
| 7 | 315 | 381 | 322 |
| 8 | 756 | 928 | 779 |
| 9 | 1 717 | 2 188 | 1 748 |
| 10 | 4 026 | 5 248 | 4 043 |
| 11 | 9 654 | 12 373 | 9 682 |
| 12 | 21 931 | 29 400 | 21 935 |
| 13 | | 68 777 | 52 510 |
| 14 | | 162 768 | 119 801 |
| 15 | | 377 359 | 277 799 |
| 16 | | 879 709 | 646 863 |
| 17 | | 2046 724 | 1473 281 |
| 18 | | 4710 612 | 3436 311 |
| 19 | | 10 996 431 | 7879 855 |

## Acknowledgement

Thanks to Beate Bollig and André Gronemeier for proofreading and helpful comments. Furthermore, the valuable suggestions of the anonymous reviewers are gratefully acknowledged.

## References

[1] K. Amano, A. Maruoka, Better upper bounds on the QOBDD size of integer multiplication, Discrete Applied Mathematics 155 (10) (2007) 1224–1232.
[2] B. Bollig, On the OBDD complexity of the most significant bit of integer multiplication, in: Proc. of 5th TAMC, in: LNCS, vol. 4978, Springer, 2008, pp. 306–317.
[3] B. Bollig, A larger lower bound on the OBDD complexity of the most significant bit of multiplication, in: Proc. of LATIN 2010, in: LNCS, vol. 6034, Springer, 2010, pp. 255–266.
[4] B. Bollig, J. Klump, New results on the most significant bit of integer multiplication, Theory of Computing Systems, in press (doi:10.1007/s00224-009-9238-y).
[5] B. Bollig, P. Woelfel, A read-once branching program lower bound of $\Omega(2^{n/4})$ for integer multiplication using universal hashing, in: Proc. of 33rd STOC, 2001, pp. 419–424.
[6] S.J. Friedman, K.J. Supowit, Finding the optimal variable ordering for binary decision diagrams, in: Proc. of 24th ACM/IEEE Design Automation Conference, DAC, 1987, pp. 348–355.
[7] M. Sauerhoff, P. Woelfel, Time-space tradeoff lower bounds for integer multiplication and graphs of arithmetic functions, in: Proc. of 35th STOC, 2003, pp. 186–195.
[8] I. Wegener, Branching Programs and Binary Decision Diagrams—Theory and Applications, in: Monographs on Discrete and Applied Mathematics, SIAM, Philadelphia, PA, 2000.
[9] I. Wegener, P. Woelfel, New results on the complexity of the middle bit of multiplication, Computational Complexity 16 (3) (2007) 298–323.
[10] P. Woelfel, Bounds on the OBDD-size of integer multiplication via universal hashing, Journal of Computer and System Sciences 71 (4) (2005) 520–534.
[11] B. Yang, Y.-A. Chen, R.E. Bryant, D.R. O'Hallaron, Space- and time-efficient BDD construction via working set control, in: Proc. of Asia and South Pacific Design Automation Conference, ASPDAC, 1998, pp. 423–432.