

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

Abstract. We prove that each OBDD (ordered binary decision diagram) for the middle bit of n -bit integer multiplication with one of the asymptotically best known variable orders, namely the pairwise ascending order $x_0, y_0, \dots, x_{n-1}, y_{n-1}$, requires size $\Omega(2^{(6/5)n})$. This is asymptotically optimal due to an upper bound of the same order by Amano and Maruoka (2007).

1. Introduction

OBDDs (ordered binary decision diagrams) are a graph 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 [6].

Definition 1. Let $X = \{x_1, \dots, x_n\}$ be a set of variables and let π be a permuted list of the variables in X called *variable order*. A π -OBDD (*ordered decision diagram*) on X 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 π . Each node v in the OBDD represents a boolean function $f_v: \{0, 1\}^n \rightarrow \{0, 1\}$ 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 by 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. 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 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 small polynomial size in the input length, at least if an appropriate variable order is chosen, while others require 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 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 $(z_{2n-1}, \dots, z_0) \in \{0, 1\}^{2n}$ denote the binary representation of their product. Then the i th output bit of n -bit multiplication, for $i \in \{0, \dots, 2n - 1\}$, is defined by $MUL_{i,n}(x, y) := z_i$.

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 [9]). 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 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, $MUL_{n-1,n}$, require 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 similar size for $MUL_{i,n}$ 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 [8] managed to improve Bryant's lower bound for $MUL_{n-1,n}$ 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 $x_0, \dots, x_{n-1}, y_0, \dots, y_{n-1}$. 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 $x_0, y_0, \dots, x_{n-1}, y_{n-1}$. 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 [3, 5, 7] or most recently with the OBDD size of the most significant bit [2].

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. (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 asymptotically best known orders today. More precisely, we obtain:

Theorem 3. *The size an OBDD with variable order $x_0, y_0, \dots, x_{n-1}, y_{n-1}$ for $MUL_{n-1,n}$ 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 $1,62 \cdot 10^{21}$. 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 [8] 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 OBDDs sizes of $MUL_{n-1,n}$ for different variable orders determined by experiments.

2. Preliminaries

For a nonnegative integer x represented by the boolean vector (x_{n-1}, \dots, x_0) in binary and $\ell \leq r$, let $[x]_\ell^r$ be the number represented by (x_r, \dots, x_ℓ) in binary and let $[x]_\ell = x_\ell$. For integers x, y with $y \neq 0$, 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 $i \in \{0, \dots, n-1\}$, $(x \bmod 2^n) \operatorname{div} 2^i = [x]_i^{n-1} = (x \operatorname{div} 2^i) \bmod 2^{n-i}$.
- (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 $d' \geq d > 0$ 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 $x \geq 0$ and $d > 0$ be integers. Then $x \operatorname{div} d = 0$ iff $x < d$.

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 [8]).

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

3. Main Lemmas and Proof of Theorem 3

The general plan of the proof of the main theorem is straightforward: For an appropriately chosen $i \in \{1, \dots, n\}$, we count the number subfunctions of $MUL_{n-1,n}$ 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. We decompose the counting of subfunctions into two main lemmas that we state here. We then apply these lemmas for proving the main theorem.

The following fact has been 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,*

$$\begin{aligned} (\text{MUL}_{n-1,n})|_{a,b}(x, y) &= [(x2^i + a) \cdot (y2^i + b)]_{n-1} \\ &= [[a]_0^{n-i-1}y + [b]_0^{n-i-1}x + [ab]_i^{n-1}]_{n-i-1}. \end{aligned}$$

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

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

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

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

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

□

The number of nodes in a quasireduced OBDD on a certain level is exactly the number of subfunctions obtained by setting all variables to constants that occur before this level according to the variable order. Therefore, the above proposition is obviously useful for proving upper bounds on the size of such OBDDs, since it shows that the number of subfunctions that result from fixing the variables $x_0, y_0, \dots, x_{i-1}, y_{i-1}$ is at most the number of triples $([a]_0^{n-i-1}, [b]_0^{n-i-1}, [ab]_i^{n-1})$ for assignments $a, b \in \{0, 1\}^i$, which is trivially upper bounded by $2^{3(n-i)}$. Our aim is to also use this characterization of subfunctions, but here we need a *lower bound* on the number subfunctions in terms of the number of triples in order to apply Lemma 5. This is exactly what we show in the first main lemma.

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

Main Lemma 1. Let $n/2 \leq i \leq n-1$. Let $a, b, c, a', b', c' \in \{0, \dots, 2^{n-i} - 1\}$ be given with a, b, a', b' odd and $(a, b, c) \neq (a', b', c')$. Then $f_{a,b,c} = f_{a',b',c'}$ implies $a' \equiv -a \pmod{2^{n-i}}$, $b' \equiv -b \pmod{2^{n-i}}$ and $c + c' \equiv 2^{n-i-1} - 1 \pmod{2^{n-i}}$.

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

Corollary 7. Let $S \subseteq \{(a, b) \mid a, b \in \{0, \dots, 2^i - 1\}, a, b \text{ odd}\}$. Then the number of different subfunctions $(\text{MUL}_{n-1,n})|_{a,b}$ 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 $x, y \in \{0, \dots, 2^{n-i} - 1\}$. By Proposition 6 and the definitions,

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

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

We prove Main Lemma 1 in the next section. Given this lemma and its corollary, we know that in order to lower bound the number of subfunctions obtained by fixing $x_0, y_0, \dots, x_{i-1}, y_{i-1}$, it suffices to count the number of triples $([a]_0^{n-i-1}, [b]_0^{n-i-1}, [ab]_i^{n-1}) \in \{0, 1\}^{3(n-i)}$. For this, our aim is to lower bound the number of different $[ab]_i^{n-1}$ for fixed $[a]_0^{n-i-1}, [b]_0^{n-i-1} \in \{0, \dots, 2^{n-i} - 1\}$ and varying $[a]_{n-i}^{i-1}, [b]_{n-i}^{i-1} \in \{0, \dots, 2^{2i-n} - 1\}$.

We reformulate this aim using the following definitions. Let

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

For $a, b \in \{0, \dots, 2^{n-i} - 1\}$ and $(x, y) \in U$, define $h_{a,b}: U \rightarrow \{0, \dots, 2^{n-i} - 1\}$ 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 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 $(3/5)n \leq i \leq (2/3)n$. Let $\alpha \in (0, 1]$. Then there is a set $A \subseteq \{(a, b) \mid a, b \in \{0, \dots, 2^{n-i} - 1\}, a, b \text{ odd}\}$ with $|A| \geq (1 - \alpha) \cdot 2^{2(n-i-1)}$ such that for all $(a, b) \in A$, $|h_{a,b}(U)| \geq (\alpha/(1 + \alpha)) \cdot 2^{n-i}$.

Corollary 7 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 $\text{MUL}_{n-1,n}$ obtained by fixing $x_0, y_0, \dots, x_{i-1}, y_{i-1}$. By Lemma 5 and Corollary 7, the number of these subfunctions is at least half the number of different triples $([a]_0^{n-i-1}, [b]_0^{n-i-1}, [ab]_i^{n-1})$ with $a, b \in \{0, \dots, 2^i - 1\}$ 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

$$\frac{3 - 2\sqrt{2}}{4}2^{3\lceil (2/5)n \rceil} - 1.$$

□

The constant in front of the term of 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 last 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 $\tilde{f}_{u,v,w}(x, y) = (uy + vx + w) \bmod 2^{n-i}$. Let $N := 2^{n-i-1}$ and $L := \{0, \dots, N - 1\}$. We observe that

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

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

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

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 $y = 0$, which 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

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

Furthermore, we have $b' \equiv b\tilde{b}$ for a suitable odd $\tilde{b} \in \{0, \dots, 2^{n-i} - 1\}$. Hence, setting $\tilde{c} := c' - c\tilde{b}$, we get

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

and thus

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

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

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

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

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

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

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

Observe that \tilde{b} is an integer with $\tilde{b} \geq 1$. Using (4), it follows that the mapping $j \mapsto (j \cdot \tilde{b} + \tilde{c}) \bmod 2^{n-i}$ 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 $\tilde{b} = 1$ and $0 = \tilde{c} = c' - c\tilde{b} = c' - c$.

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

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

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

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

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

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

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

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

5. Proof of Main Lemma 2

For what follows, let $n/2 \leq i \leq n-1$ and $m := 2i - n$. We first rewrite the type of products which we consider for the proof of the main lemma in a more suitable way.

Proposition 8. For $a, b \in \{0, \dots, 2^{n-i} - 1\}$ and $x, y \in \{0, \dots, 2^m - 1\}$,

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

Proof: The proof is similar to that of Proposition 6. First, we get

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

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

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

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

Recall the following definitions from Section 3. We have

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

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

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

By Proposition 8, we also have

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

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

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 more general variant of Woelfel's technique [8] based on universal hashing. The key observation is that a function $h \in \mathcal{H}$ has large range if it has a small number of *collisions*, i. e., pairs of different arguments from U mapped to the same value. Different from [8], we cannot prove a useful bound on the probability that for *each* pair of different arguments from U , a random function from \mathcal{H} has a small probability of inducing a collision. Instead, we consider the average number of collisions over all functions from \mathcal{H} .

Definition 9. Let \mathcal{H} be a class of functions $U \rightarrow R$, 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 similar to proofs of universality of hash classes, while the additional first part is required here to get a sufficiently good bound).

Lemma 10. Let $n/2 \leq i \leq (2/3)n$. For $k \in \{0, \dots, m-1\}$ let c, d be such that $|c|, |d| \in \{0, \dots, 2^{m-k} - 1\}$ and such that c is odd.

- (1) Let $a, b \in \{0, \dots, 2^{n-i} - 1\}$, a, b odd. Let $x, x' \in \{0, \dots, 2^m - 1\}$ with $x' - x = c2^k$. Then there are at most 2^k values $y \in \{0, \dots, 2^m - 1\}$ such that $y' := y + d2^k \in \{0, \dots, 2^m - 1\}$ and $h_{a,b}(x, y) = h_{a,b}(x', y')$.
- (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 $y \in \{0, \dots, 2^m - 1\}$ be given and suppose that $y' = y + d2^k \in \{0, \dots, 2^m - 1\}$. By the definition of $h_{a,b}$, Proposition 8, and Proposition 4, 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

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

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

$$\begin{aligned} (x'y' - xy) \cdot 2^{n-i} + a(y' - y) + b(x' - x) &= (dx2^k + cy2^k + cd2^{2k})2^{n-i} + ad2^k + bc2^k \\ &= cy2^{n-i+k} + dx2^{n-i+k} + r, \end{aligned}$$

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

Hence, (*) is equivalent to

$$cy2^{n-i+k} + dx2^{n-i+k} \equiv e - r \pmod{2^i}.$$

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 $e < 2^m$ and $m \leq n - i$ by the assumption that $i \leq (2/3)n$, $e - r \equiv 0 \pmod{2^{n-i+k}}$ implies that the value of e is fixed given a, b, c, d and k and thus r . In particular, it does not depend on $x, y, x',$ and y' .

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

$$cy + dx \equiv (e - r)2^{-(n-i+k)} \pmod{2^{m-k}}.$$

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

$$y \equiv c^{-1}((e - r)2^{-(n-i+k)} - dx) \pmod{2^{m-k}}.$$

Since $y \in \{0, \dots, 2^m - 1\}$ and e does not depend on y , this means that there are at most 2^k suitable values for y . Hence, for 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 $k \in \{0, \dots, m-1\}$, $c \in \mathbb{Z}_{2^{m-k}}^*$, and $d \in \mathbb{Z}_{2^{m-k}}$. In the proof of the first part, it has been shown that $h_{a,b}(x, y) = h_{a,b}(x', y')$ only if

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

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

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

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

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

By the assumption $i \leq (2/3)n$, we have $n - i - (m - k) \geq 0$. Since the statement of part (2) is trivially true if $i = (2/3)n$ and $k = 0$, we may also assume that $n - i - (m - k) > 0$. Suppose that $a = s2^{m-k} + t$ 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

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

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

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

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

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 $(x, y) \neq (x', y')$, then $x' - x \neq 0$ or $y' - y \neq 0$. 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 10, part (2), that the innermost sum of α 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 10, 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' :

$$\begin{aligned} \sum_{c \in \mathbb{Z}_{2^{m-k}}^*} \sum_{\substack{x, x' \in \mathbb{Z}_{2^m} \\ |x-x'|=c2^k}} 1 &= \sum_{c \in \mathbb{Z}_{2^{m-k}}^*} 2 \cdot \sum_{0 \leq x \leq 2^m - c2^k - 1} 1 = \sum_{c \in \mathbb{Z}_{2^{m-k}}^*} 2 \cdot (2^m - c2^k) \\ &= 2^{2m-k} - 2^{k+1} \cdot \sum_{j=0}^{2^{m-k-1}-1} (2j+1) = 2^{2m-k} - 2^{k+1} \cdot 2^{2(m-k-1)} = 2^{2m-k-1}. \end{aligned}$$

Putting everything together, we get

$$\begin{aligned} \alpha &\leq \sum_{k=0}^{m-1} 2^{2m-k-1} \cdot 2^{m-k} \cdot 2^k \cdot 2^{-(2n-3i+k)} = 2^{3m+3i-2n-1} \sum_{k=0}^{m-1} 2^{-2k} \\ &= 2^{3m+3i-2n-1} \cdot \frac{4}{3} (1 - 2^{-2m}) = \frac{2}{3} (2^{9i-5n} - 2^{5i-3n}), \end{aligned}$$

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 [8].

Lemma 12 ([8]). Let \mathcal{H} be a class of functions $U \rightarrow R$, 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 $u := |U|$.

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

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

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

which after rearranging gives the claimed lower bound on r ,

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

□

Proof of Main Lemma 2: By Lemma 11, 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

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

By Markov's inequality, $|A| \geq (1 - \alpha)|U|$. Let $(a, b) \in A$. By Lemma 12,

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

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. □

6. Conclusion

We conclude with a comparison of the OBDD sizes of $MUL_{n-1,n}$ 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:

- *Optimal variable order, π_{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 ([4], see also [6], Section 5.5).
- *Pairwise ascending order, π_{pwa}* : $x_0, y_0, \dots, x_{n-1}, y_{n-1}$.
- *Hybrid order, π_{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 $x_m, \dots, x_1, y_m, \dots, y_1, x_{m+1}, y_{m+1}, \dots, x_{n-2}, y_{n-2}, x_0, y_{n-1}, x_{n-1}, y_0$.

Optimal variable orders for $MUL_{n-1,n}$ 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 $\Omega(2^{(6/5)n})$.

n :	π_{opt} :	π_{pwa} :	π_{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	1717	2188	1748
10	4026	5248	4043
11	9654	12373	9682
12	21931	29400	21935
13		68777	52510
14		162768	119801
15		377359	277799
16		879709	646863
17		2046724	1473281
18		4710612	3436311
19		10996431	7879855

Table 1: OBDD sizes of $MUL_{n-1,n}$ for different variable orders.

Acknowledgment

Thanks to Beate Bollig and André Gronemeier for proofreading and helpful comments.

References

- [1] K. Amano and A. Maruoka. Better upper bounds on the QOBDD size of integer multiplication. *Discrete Applied Mathematics*, 155(10):1224–1232, 2007.
- [2] B. Bollig. On the OBDD complexity of the most significant bit of integer multiplication. In *Proc. of 5th TAMC*, 306–317, 2008.
- [3] B. Bollig and 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*, 419–424, 2001.
- [4] S. J. Friedman and K. J. Supowit. Finding the optimal variable ordering for binary decision diagrams. In *Proc. of 24th DAC*, 348–355, June 1987.
- [5] M. Sauerhoff and P. Woelfel. Time-space tradeoff lower bounds for integer multiplication and graphs of arithmetic functions. In *Proc. of 35th STOC*, 186–195, 2003.
- [6] I. Wegener. *Branching Programs and Binary Decision Diagrams—Theory and Applications*. Monographs on Discrete and Applied Mathematics. SIAM, Philadelphia, PA, 2000.
- [7] I. Wegener and P. Woelfel. New results on the complexity of the middle bit of multiplication. *Computational Complexity*, 16(3):298–323, 2007.
- [8] P. Woelfel. Bounds on the OBDD-size of integer multiplication via universal hashing. *Journal of Computer and System Sciences*, 71(4):520–534, 2005.
- [9] B. Yang, Y.-A. Chen, R. E. Bryant, and D. R. O’Hallaron. Space- and time-efficient BDD construction via working set control. In *Proc. of 3rd ASP-DAC*, 423–432, 1998.