Electronic Colloquium on Computational Complexity, Report No. 56 (2008)

On the OBDD complexity of the most significant bit of integer multiplication

Beate Bollig

LS2 Informatik, TU Dortmund,
44221 Dortmund, Germany

Abstract. Integer multiplication as one of the basic arithmetic functions has been in the focus of several complexity theoretical investigations. Ordered binary decision diagrams (OBDDs) are one of the most common dynamic data structures for boolean functions. Among the many areas of application are verification, model checking, computer-aided design, relational algebra, and symbolic graph algorithms. In this paper it is shown that the OBDD complexity of the most significant bit of integer multiplication is exponential answering an open question posed by Wegener (2000).

Keywords: Computational complexity, integer multiplication, lower bounds, ordered binary decision diagrams

1 Introduction and Result

Integer multiplication is certainly one of the most important functions in computer science and a lot of effort has been spent in designing good algorithms and small circuits and in determining its complexity. For one of the latest results see, e.g., [8]. Ordered binary decision diagrams (OBDDs) are the most common dynamic data structure for boolean functions. Although many exponential lower bounds on the OBDD size of boolean functions are known and the lower bound methods are simple, it is often a more difficult task to prove large lower bounds for some predefiend and interesting functions. Despite the well-known lower bounds on the OBDD size of the so-called middle bit of multiplication ([7], [17]), until now the OBDD complexity of the most significant bit of multiplication has been unknown and Wegener [16] has asked whether its OBDD complexity is exponential. In the following we answer his question affirmatively.

1.1 Branching programs or binary decision diagrams

Besides boolean circuits and formulae, branching programs (BPs), sometimes also called binary decision diagrams (BDDs), are one of the standard representations for boolean functions. (For a history of results on branching programs see, e.g., the monograph of Wegener [16]).

ISSN 1433-8092

Definition 1. *A* branching program (BP) *on the variable set $X_n = \{x_1, \ldots, x_n\}$ is a directed acyclic graph with one source and two sinks labeled by the constants 0 and 1. Each non-sink node (or decision node) is labeled by a boolean variable and has two outgoing edges, one labeled by 0 and the other by 1.*

An input $b \in \{0,1\}^n$ activates all edges consistent with b, i.e., the edges labeled by b_i which leave nodes labeled by x_i. A computation path *for an input b in a* BP *G is a path of edges activated by the input b which leads from the source to a sink. A computation path for an input b which leads to the 1-sink is called* accepting path *for b.*

Let B_n denote the set of all boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}$. The BP *$G$ represents a function $f \in B_n$ for which $f(b) = 1$ iff there exists an accepting path for the input b.*

The size *of a branching program G is the number of its nodes and is denoted by $|G|$. The* branching program size *of a boolean function f is the size of the smallest* BP *representing f. The* length *of a branching program is the maximum length of a path.*

It is well known that the logarithm of the branching program size is essentially the same as the space complexity of the nonuniform variant of Turing machines. Hence, it is a fundamental open problem to prove superpolynomial lower bounds on the size of branching programs for explicitly defined boolean functions. In order to develop and strengthen lower bound techniques one considers restricted computation models. There are several possibilities to restrict BPs, among them restrictions on the multiplicity of variable tests or the order in which variables may be tested.

Definition 2. *i) A branching program is called* read-k-times (BPk) *if each variable is tested on each path at most k times.*

ii) A branching program is called s-oblivious *for a sequence of variables $s = (s_1, \ldots, s_l)$, $s_i \in X_n$, or short* oblivious, *if the set of decision nodes can be partitioned into disjoint sets V_i, $1 \leq i \leq l$, such that all nodes from V_i are labeled by s_i and the edges which leave V_i-nodes reach a sink or a V_j-node where $j > i$. The* length *of an s-oblivious branching program is the length of the sequence s.*

Besides the complexity theoretical viewpoint people have used branching programs in applications. Representations of boolean functions that allow efficient algorithms for many operations, in particular synthesis (combine two functions by a binary operation) and equality test (do two representations represent the same function?) are necessary. Bryant [6] introduced ordered binary decision diagrams (OBDDs) which have become the most popular data structure for boolean functions. Among the many areas of application are verification, model checking, computer-aided design, relational algebra, and symbolic graph algorithms.

Definition 3. *An* OBDD *is a branching program with a* variable order *given by a permutation π on the variable set. On each path from the source to the sinks, the variables at the nodes have to appear in the order prescribed by π (where* *some variables may be left out). A* π-OBDD *is an* OBDD *ordered according to* π*. The* π-OBDD *size of* f *denoted by* π-OBDD(f) *is the size of the smallest* π-OBDD *representing* f*. The* OBDD size *of* f*, sometimes also called* OBDD complexity *of* f*, (denoted by* OBDD(f)*) is the minimum of all* π-OBDD(f)*.*

1.2 Integer multiplication and binary decision diagrams

Lower bounds for integer multiplication are motivated by the general interest in the complexity of important arithmetic functions.

Definition 4. *The boolean function* $\mathrm{MUL}_{i,n} \in B_{2n}$ *maps two* n*-bit integers* $x = x_{n-1} \dots x_0$ *and* $y = y_{n-1} \dots y_0$ *to the* i*th bit of their product, i.e.,* $\mathrm{MUL}_{i,n}(x, y) = z_i$*, where* $x \cdot y = z_{2n-1} \dots z_0$*.*

The bit z_{2n-1} is the most significant bit of integer multiplication in the following sense. Let $(z_{2n-1}, \dots, z_0)$ be the binary representation of the integer z, i.e., $z = \sum_{i=0}^{2n-1} z_i \cdot 2^i$. Since the bit z_{2n-1} has the highest value, for the approximation of the value of the product of two n-bit numbers x and y it is the most interesting one. On the other hand for space bounded models of computation the most significant bit of integer multiplication is the easiest one to compute in the sense that if it cannot be computed with size $s(n)$, then any other bit z_i, $2n-1 > i \geq n-1$, cannot be computed with size $s(n/2)$. Moreover, if the bit z_{n-1} cannot be computed with size $s(n/2)$, any other bit z_i, $n-1 > i \geq 0$, cannot be computed in size $s(i/2)$.

The middle bit of integer multiplication (the bit z_{n-1}) is the hardest bit to compute for space bounded models of computation in the sense that if it can be computed with size $s(n)$, then any other bit can be computed with size at most $s(2n)$. More precisely, any branching program for $\mathrm{MUL}_{2n-1,2n}$ can be converted into a branching program representing $\mathrm{MUL}_{i,n}$, $0 \leq i \leq 2n-1$, by relabeling the nodes and by replacing some inputs with the constant 0. Therefore, the first exponential lower bounds have been proved for $\mathrm{MUL}_{n-1,n}$. For OBDDs Bryant [7] has presented an exponential lower bound of $2^{n/8}$ and Gergov has extended the result for so-called nondeterministic linear-length oblivious branching programs [9]. Later Ponzio has shown that the complexity of this function is $2^{\Omega(\sqrt{n})}$ for read-once branching programs [12]. Progress in the analysis of $\mathrm{MUL}_{n-1,n}$ has been achieved by a new approach using universal hashing. Woelfel [17] has improved Bryant's lower bound to $\Omega(2^{n/2})$ and Bollig and Woelfel [3] have presented a lower bound of $\Omega(2^{n/4})$ for read-once branching programs. Exponential lower bounds have also been proved for more general read-once branching program models that allow limited nondeterminism and for models where some but not all variables may be tested multiple times (see, e.g., [2], [5], [18], [4]). Finally, Sauerhoff and Woelfel [13] have presented exponential lower bounds on the size of read-k-times branching programs representing the middle bit of multiplication.

Despite the well-known lower bounds for the middle bit of multiplication, until now the OBDD complexity of the most significant bit of multiplication has been unknown. Since the most significant bit is a monotone function it seems to be easier to compute than the middle bit. The known upper bounds on the OBDD size confirms this intuition. Amano and Maruoka [1] have presented an upper bound of $O(2^n)$ on the OBDD size of the most significant bit of multiplication, whereas the best known upper bound for the middle bit is $O(2^{(6/5)n})$. Furthermore, in the lower bound proofs on the OBDD size for $\text{MUL}_{n-1,n}$ it has been shown that for an arbitrary variable order π there exists an assignment b to one of the input vectors such that the π-OBDD size for the resulting subfunction is exponential. In contrast it is not difficult to see that the π-OBDD size for any subfunction of $\text{MUL}_{2n-1,n}$ where one of the input vectors is a constant is $O(n^2)$.

Computing the set of nodes that are reachable from some source $s \in V$ in a digraph $G = (V, E)$ is an important problem in computer-aided design, hardware verification, and model checking. Proving exponential lower bounds on the space complexity of a common class of OBDD-based algorithms for the reachability problem, Sawitzki [14] has presented the first exponential lower bound on the size of π-OBDDs representing the most significant bit for the variable order π where the variables are tested according to increasing significance, i.e. $\pi = (x_0, y_0, x_1, y_1, \ldots, x_{n-1}, y_{n-1})$. For the lower bounds on the space complexity of the OBDD-based algorithms he has used the assumption that the output OBDDs use the same variable order as the input OBDDs. But in contrast, practical algorithms usually run variable reordering heuristics on intermediate OBDD results in order to minimize their size. Therefore, it is interesting whether the OBDD complexity of the most significant bit of multiplication is exponential.

In this paper we present the following result.

Theorem 1. $\text{OBDD}(\text{MUL}_{2n-1,n}) = \Omega(2^{n/288})$.

As a by-product we improve Sawitzkis lower bound on the π-OBDD size for the variable order $\pi = (x_0, y_0, x_1, y_1, \ldots, x_{n-1}, y_{n-1})$ [14] up to $\Omega(2^{n/4})$ using a much simpler proof.

2 Preliminaries

2.1 Notation

In the rest of the paper we use the following notation.

Let $[x]_r^l$, $n-1 \geq l \geq r \geq 0$, denote the bits $x_l \ldots x_r$ of a binary number $x = (x_{n-1}, \ldots, x_0)$. For the ease of description we use the notation $[x]_r^l = z$ if $(x_l, \ldots, x_r)$ is the binary representation of the integer $z \in \{0, \ldots, 2^{l-r+1}-1\}$. Sometimes, we identify $[x]_r^l$ with z if the meaning is clear from the context.

Let $\ell \in \{0, \ldots, 2^m - 1\}$, then $\overline{\ell}$ denotes the number $(2^m - 1) - \ell$.

2.2 Communication Complexity

In order to obtain lower bounds on the size of OBDDs one-way communication complexity has become a standard technique (see Hromkovič [10] and Kushilevitz and Nisan [11] for the theory of communication complexity and the results mentioned below).

The main subject is the analysis of the following (restricted) communication game. Consider a boolean function $f \in B_n$ which is defined on the variables in $X_n = \{x_1, \ldots, x_n\}$, and let $\mathit{\Pi} = (X_A, X_B)$ be a partition of X_n. Assume that Alice has only access to the input variables in X_A and Bob has only access to the input variables in X_B. In a one-way communication protocol, upon a given input x, Alice is allowed to send a single message (depending on the input variables in X_A) to Bob who must then be able to compute the answer $f(x)$. The *one-way communication complexity* of the function f denoted by $C(f)$ is the worst case number of bits of communication which need to be transmitted by such a protocol that computes f. It is easy to see that an OBDD G with respect to a variable order where the variables in X_A are tested before the variables in X_B can be transformed into a communication protocol and $C(f) \leq \lceil \log |G| \rceil$. Therefore, linear lower bounds on the communication complexity of a function f lead to exponential lower bounds on the OBDD complexity.

One central notion of communication complexity are fooling sets which play an important role for the lower bound proof used later on.

Definition 5. *Let $f : \{0,1\}^{|X_A|} \times \{0,1\}^{|X_B|} \to \{0,1\}$. A set $S \subseteq \{0,1\}^{|X_a|} \times \{0,1\}^{|X_B|}$ is called* fooling set *for f if $f(a,b) = c$ for all $(a,b) \in S$ and some $c \in \{0,1\}$ and if for different pairs $(a_1,b_1),(a_2,b_2) \in S$ at least one of $f(a_1,b_2)$ and $f(a_2,b_1)$ is unequal to c.*

Theorem 2. *If $f : \{0,1\}^{|X_A|} \times \{0,1\}^{|X_B|} \to \{0,1\}$ has a fooling set of size t, the communication complexity of f is bounded below by $\lceil \log t \rceil$.*

Because of our considerations above, the size t of a fooling set for f is a lower bound on the size of OBDDs representing f with respect to a variable order where the variables X_A are tested before the variables X_B. Because of the symmetric definition of fooling sets, t is also a lower bound on the size of OBDDs representing f with respect to a variable order where the variables X_B are tested before the variables X_A. The crucial thing to prove large lower bounds on the OBDD complexity of a function is to obtain for all partitions of the variables large lower bounds on the size of fooling sets for subfunctions of the given function.

Now we take a look at known results about the communication complexity of some functions. Let EQ: $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be defined by $\mathrm{EQ}(a,b) = 1$ iff the vectors $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ are equal. It is well-known and easy to prove that $C(\mathrm{EQ}) = n$. Similar results can be obtained for the functions $\overline{\mathrm{GT}} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, $\overline{\mathrm{GT}^*} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, and $\overline{\mathrm{GT}^{**}} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, where $\overline{\mathrm{GT}}(a,b) = 1$ iff $[a]_1^n \leq [b]_1^n$, $\overline{\mathrm{GT}^*}(a,b) = 1$ iff $\overline{\alpha} \leq [b]_1^n$, where α is the integer with binary representation a, and $\overline{\mathrm{GT}^{**}}(a,b) = 1$ iff $[a]_1^n \leq \overline{\beta}$, where β is the integer with binary representation b. Furthermore, obviously the same results can be obtained if Alice gets exactly one of the variables a_i and b_i, $1 \leq i \leq n$. (The reason is that for $\overline{\mathrm{GT}^*}$ and $\overline{\mathrm{GT}^{**}}$ the variables of the same significance are symmetric variables, i.e., variables that can be exchanged without changing the considered functions. To be more precise two variables z_i and z_j are symmetric variables for a Boolean function f when $f_{|z_i=0,z_j=1} = f_{|z_i=1,z_j=0}$. For $\overline{\mathrm{GT}}$ we choose as fooling set all assignments where the variables of the same significance are equal.)

The addition function $\mathrm{ADD}_{i,n} \in B_{2n}$ maps two n-bit integers $x = x_{n-1} \dots x_0$ and $y = y_{n-1} \dots y_0$ to the ith bit of their sum, i.e., $\mathrm{ADD}_{i,n}(x, y) = s_i$, where $x + y = s_n \dots s_0$. It is easy to see that $\mathrm{ADD}_{n,n}$ has a fooling set of size 2^n if for each i, $0 \le i \le n-1$, Alice gets exactly one of the variables x_i and y_i. The idea of Bryant's lower bound proof on the OBDD size of $\mathrm{MUL}_{n-1,n}$ [7] is the following. For each variable order, there is a subfunction of $\mathrm{MUL}_{n-1,n}$ which essentially equals the computation of the output bit at position m of the addition of two m-bit numbers x and y where $m \ge n/8$. The variable order is bad in the sense that among Alice's m variables is exactly one of the variables x_i and y_i.

3 An exponential lower bound on the OBDD complexity of the most significant bit of integer multiplication

In this section, we prove Theorem 1 and determine a lower bound of $\Omega(2^{n/288})$ on the size of OBDDs for the representation of the most significant bit of multiplication mentioned above. We start to prove a lower bound of $\Omega(2^{n/432})$ and present afterwards ideas how to improve this lower bound up to $\Omega(2^{n/288})$.

Besides Bryant's lower bound proof on the size of OBDDs representing the middle bit of multiplication we use the idea of the following reduction from multiplication to squaring presented by Wegener [15] where squaring computes the square of an n-bit input. For two n-bit numbers u and w the number $z := u \cdot 2^{2(n+1)} + w$ is defined. Then

$$z^2 = u^2 \cdot 2^{4(n+1)} + uw2^{2(n+1)+1} + w^2.$$

Since w^2 and uw are numbers of length $2n$, the binary representation of the product uw can be found in the binary representation of z^2.

In the following for the sake of simplicity we do not apply floor or ceiling functions to numbers even when they need to be integers whenever this is clear from the context and has no bearing on the essence of the proof.

We start with a simplified presentation of our main proof ideas. Our aim is to show for an arbitrary variable order π that a π-OBDD for $\mathrm{MUL}_{2n-1,n}$ contains in a certain way a π-OBDD for the function $\overline{\mathrm{GT}^{**}}(w', w'')$, where the length of the inputs w' and w'' is $\Theta(n)$ and the w'-variables are before the w''-variables in π. Therefore, there exists a large fooling set and as a consequence also the size of the π-OBDD for $\mathrm{MUL}_{2n-1,n}$ has to be large. The vectors w' and w'' are subvectors of one of the inputs x and y for $\mathrm{MUL}_{2n-1,n}$, in the following w.l.o.g. of x. The key observation is the following one.

Claim 1. *For a number $2^{n-1} + \ell 2^{(1/2)n}$, $\ell \le 2^{n/6-1}$, the corresponding smallest number such that the product of the two numbers is at least 2^{2n-1} is $2^n - \ell 2^{(1/2)n+1} + 4\ell^2$. (Figure 2 shows the corresponding x- and y-inputs.)*

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

Proof. Let a be an integer $2^{n-1} + \ell 2^{n/2}$, where $1 \leq \ell \leq 2^{n/6-1}$. Then the smallest number b_a such that $a \cdot b_a \geq 2^{2n-1}$ and therefore $\mathrm{MUL}_{2n-1,n}(a, b_a) = 1$ is

$$\left\lceil \frac{2^{2n-1}}{2^{n-1} + \ell 2^{n/2}} \right\rceil = 2^n - \ell 2^{n/2+1} + 4\ell^2 - \left\lfloor \frac{4\ell^3}{2^{n/2-1} + \ell} \right\rfloor .$$

Since ℓ is at most $2^{n/6-1}$ the last term is 0 and we are done. □

For realizing our proof idea we have to make sure that if x represents a number $2^{n-1} + \ell 2^{n/2}$, $1 \leq \ell \leq 2^{n/6-1}$, the upper half of y represents the number $2^{n/2} - 2\ell$, i.e., $[y]_{n/2}^{n-1} = 2^{n/2} - 2\ell$. We will see that if we cannot guarantee this requirement, the π-OBDD size for $\mathrm{MUL}_{2n-1,n}$ is large.

In order to use Wegener's observation on squaring mentioned above combined with Bryant's lower bound proof we only consider integers ℓ where $\ell = u2^{2(m+1)} + w$, $u, w < 2^m$ and $m = n/18 - 1$. (Later on we show that m can be enlarged up to $n/12 - 7/6$ which leads to a larger lower bound.) For this reason we replace the variables $x_{n/2+m}, \ldots, x_{n/2+2m+1}$ by 0. Afterwards we replace some of the x-variables by constants such that $u \cdot w$ is equal to the sum $w''2^{2d+c} + (w'' + w')2^{d+c} + w'2^c$, where w' and w'' are different parts of x. The length n' of w' and w'' is at least $m/8 = \Theta(n)$ and $d > n'$. Furthermore, as a simplification we can assume that the w'-variables are before the w''-variables in π.

The last step is to replace some of the y-variables such that the lower part of y can be seen as a number of at least $4\ell^2$, where $x = 2^{n-1} + \ell 2^{n/2}$, iff the sum of w' and w'' is at most $2^{n'} - 1$.

Now we make these ideas more precise. We start our proof by the following observation.

Lemma 1. *A pair (x_i, y_{i+1}), $n/2 + 1 \leq i \leq (3/4)n - 2$, is called (x, y)-pair. Let S be the set of the first $|S|$ variables according to a variable order π. A pair (x_i, y_{i+1}) is called separated with respect to S iff $x_i \in S$ and $y_{i+1} \notin S$ or vice versa. If there exist a set S according to π such that there are at least m separated (x, y)-pairs with respect to S, the π-OBDD size of the most significant bit of integer multiplication is at least 2^m.*

Proof. In the following, we prove the existence of a fooling set with at least 2^m elements. For this reason we choose a subfunction of $\mathrm{MUL}_{2n-1,n}$ such that the computation of this subfunction resembles the computation of the function $\overline{\mathrm{GT}^*_m}$.

The key observation is the following one.

Claim 2. *For a number $2^{n-1} + \ell 2^{(1/2)n}$, $\ell < 2^{(1/4)n-1}$, the corresponding smallest number divisible by $2^{(1/2)n}$ such that the product of the two numbers is at least 2^{2n-1} is $2^n - \ell 2^{(1/2)n+1} + 2^{(1/2)n}$.*

For the sake of completeness we include the simple proof.

Proof. From the proof of Claim 1 we know that $b_a = 2^n - \ell 2^{n/2+1} + 4\ell^2 - \lfloor \frac{4\ell^3}{2^{n/2-1}+\ell} \rfloor$ is the smallest number such that $a \cdot b_a \geq 2^{2n-1}$ for $a = 2^{n-1} + \ell 2^{n/2}$. Since $4\ell^2 \geq \lfloor \frac{4\ell^3}{2^{n/2-1}+\ell} \rfloor$ and $4\ell^2 < 2^{(1/2)n}$ we obtain the desired result. □

We assume the existence of a set S according to π such that there are at least m separated (x,y)-pairs.

Now we replace some of the variables in the following way.

- $y_{n-1}, \ldots, y_{(3/4)n}$ are replaced by 1,
- $y_{n/2}, y_{n/2+1}$ are replaced by 1,
- $y_0, \ldots, y_{n/2-1}$ are replaced by 0,
- $x_0, \ldots, x_{n/2-1}$ are replaced by 0,
- $x_{n/2}$ is replaced by 1,
- x_{n-1} is replaced by 1,
- $x_{n-2}, \ldots, x_{(3/4)n-1}$ are replaced by 0,
- x_i is replaced by 1 and y_{i+1} is replaced by 0 if $i \in \{n/2+1, \ldots, (3/4)n-2\}$ and (x_i, y_{i+1}) is not separated with respect to S.

Figure 1 illustrates some of these replacements. The effect is the following one. For each assignment to the separated x-variables the corresponding smallest assignment to the separated y-variables such that the product of x and y is at least 2^{2n-1} has the property that $y_{i+1} = x_i \oplus 1$ (for all i where x_i is a separated variable).

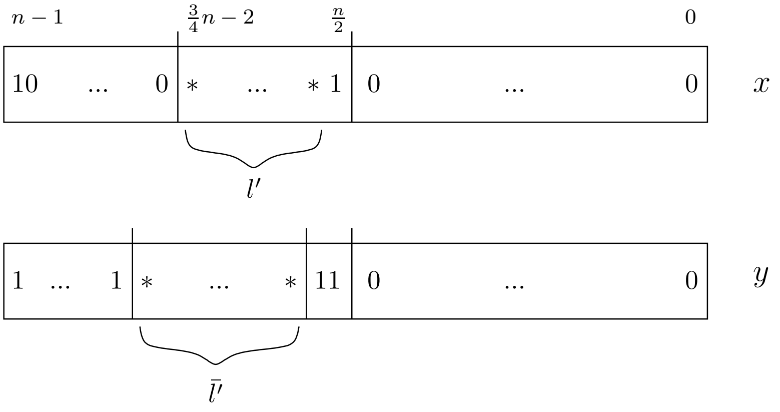

Fig. 1. *The effect of the replacements in the proof of Lemma 1.*

In the rest of the proof we show that all assignments to the separated x- and y-variables in S together with the corresponding assignments to the remaining separated x-and y-variables not in S are a fooling set of size at least 2^m.

Let b_S be an assignment to the separated x- and y-variables in S and b_r the corresponding assignment to the remaining separated x- and y-variables. In the following $b_S(x_i)$ $(b_S(y_j))$ denotes the assignment of b_S to the separated variable x_i (y_j) in S. Together with the first replacements to constants b_S and b_r can be seen as numbers $2^{n-1} + \ell 2^{n/2}$ and $2^n - \ell 2^{n/2+1} + 2^{n/2}$.

$$\begin{array}{ll} & (2^{n-1} + \ell 2^{n/2}) \cdot (2^n - \ell 2^{n/2+1} + 2^{n/2}) \\ = & 2^{2n-1} + 2^{(3/2)n}\ell - 2^{(3/2)n}\ell - 2^{n+1}\ell^2 + 2^{(3/2)n-1} + 2^n\ell \\ = & 2^{2n-1} - 2^{n+1}\ell^2 + 2^{(3/2)n-1} + 2^n\ell \\ > & 2^{2n-1}. \end{array}$$

Therefore, $\mathrm{MUL}_{2n-1,n}(2^{n-1} + \ell 2^{n/2}, 2^n - \ell 2^{n/2+1} + 2^{n/2}) = 1$.

Let b'_S and b''_S be two different assignments to the separated x- and y-variables in S, b'_r and b''_r the two corresponding assignments to the remaining separated x- and y-variables. Let $i_{max} := \max\{i \mid b'_S(x_i) \neq b''_S(x_i)$ or $b'_S(y_{i+1}) \neq b''_S(y_{i+1})\}$. W.l.o.g. let $b'_S(x_{i_{max}}) \neq b''_S(x_{i_{max}})$ and $b'_S(x_{i_{max}}) > b''_S(x_{i_{max}})$. Since $b'_r(y_{i_{max}+1}) < b''_r(y_{i_{max}+1})$, we can conclude that b''_S together with b'_r and the first replacements to constants can be seen as numbers $2^{n-1} + \ell_1 2^{n/2}$ and $2^n - \ell_2 2^{n/2+1} + 2^{n/2}$, where $\ell_2 > \ell_1$.

We get the following result.

$$\begin{array}{ll} & (2^{n-1} + \ell_1 2^{n/2}) \cdot (2^n - \ell_2 2^{n/2+1} + 2^{n/2}) \\ = & 2^{2n-1} + 2^{(3/2)n}\ell_1 - 2^{(3/2)n}\ell_2 - 2^{n+1}\ell_1\ell_2 + 2^{(3/2)n-1} + 2^n\ell_1 \\ < & 2^{2n-1}. \end{array}$$

Therefore, $\mathrm{MUL}_{2n-1,n}(2^{n-1} + \ell_1 2^{n/2}, 2^n - \ell_2 2^{n/2+1} + 2^{n/2}) = 0$. □

Obviously, the same result can be shown if we change the roles of the x- and y-variables.

In the following let π be an arbitrary variable order.

First, we take a closer look at the variables $x_{n/2}, \ldots, x_{n/2+n/6-2}$. For the ease of description we assume that $(n/6-1) \bmod 3 = 2$. We rename $[x]_{n/2}^{n/2+n/18-2}$ by $[w]_0^{m-1}$ and $[x]_{n/2+n/9}^{n/2+n/6-2}$ by $[u]_0^{m-1}$, where $m := (n/6-3)/3$. Figure 2 illustrates the partition of the input x.

Let S be the set of the first $|S|$ variables according to π where there are at least $m/2$ variables from $\{w_0, \ldots, w_{m-1}\}$ for the first time. Let $I_S \subseteq \{0, \ldots, m-1\}$ be the set of indices i for which $w_i \in S$. Using simple counting arguments we can prove that there exists a distance parameter d such that there exists a set of pairs $P = \{(w_i, w_{i+d}) | i \in I_S$ and $(i+d) \notin I_S$ or $i \notin I_S$ and $(i+d) \in I_S$, where $0 \leq i < m/2 \leq i+d \leq m-1\}$ and $|P| \geq m/8$ (see [7] for a similar proof). Let I'' be the set of indices i, $0 \leq i < m/2$, where w_i belongs to a pair in P.

Case 1: There are at least $m/24$ separated $(x_{n/2+i}, y_{n/2+i+1})$-pairs with respect to S, where $i \in I''$ or $i - d \in I''$. Using Lemma 1 we can conclude that the π-OBDD size of the most significant bit of integer multiplication is at least $2^{m/24}$.

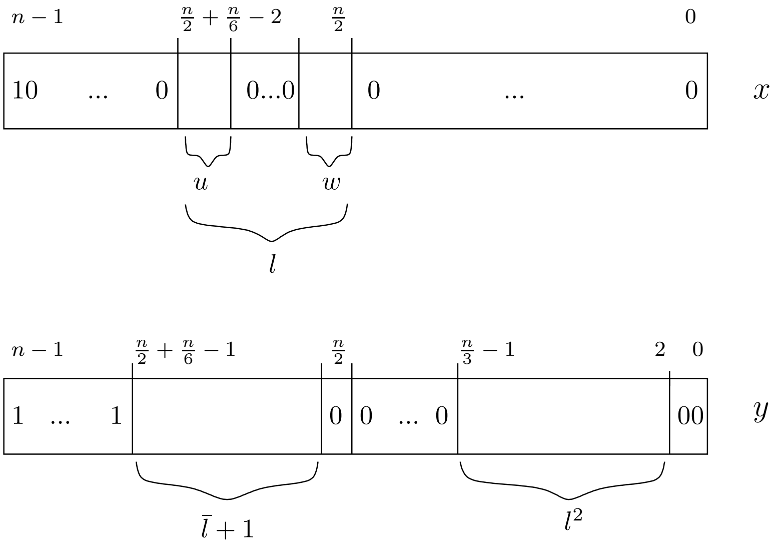

Fig. 2. *The partition of the inputs x and y.*

Case 2: There are less than $m/24$ separated $(x_{n/2+i}, y_{n/2+i+1})$-pairs with respect to S, where $i \in I''$ or $i-d \in I''$. Let $I' \subseteq I''$ be the set of indices such that $(x_{n/2+i}, y_{n/2+i+1})$ and $(x_{n/2+i+d}, y_{n/2+i+d+1})$, $i \in I''$, are not separated with respect to S. Obviously, $|I'| \geq (2/24)m$.

Now we replace some of the variables in the following way.

- $y_{n-1}, \ldots, y_{n/2+n/6}$ are replaced by 1,
- $y_{n/2}, \ldots, y_{n/3}$, y_1, and y_0 are replaced by 0,
- x_{n-1} is replaced by 1,
- $x_{n-2}, \ldots, x_{n/2+n/6-1}$ are replaced by 0,
- $x_{n/2+n/9-1}, \ldots, x_{n/2+n/18-1}$ are replaced by 0,
- $y_{n/2+n/9}, \ldots, y_{n/2+n/18}$ are replaced by 1,
- $x_0, \ldots, x_{n/2-1}$ are replaced by 0.

Figure 2 illustrates these replacements. Furthermore, u_0 and u_d are set to 1, all other u-variables are set to 0. The effect of these replacements is that $[u]_0^{m-1} = 2^d + 1 =: u$. The corresponding y-variables $y_{n/2+n/9+1}$ and $y_{n/2+n/9+d+1}$ are set to 0, all other variables y_j, where $n/2 + n/6 - 1 \leq j \leq n/2 + n/9 + 1$ are replaced by 1. The variables y_{4m+6}, y_{4m+d+7}, and $y_{4m+2d+6}$ are set to 1, the other variables y_j with $4m+6 \leq j \leq 6m+5$ are set to 0. The effect of these replacemenst is that $[y]_{4m+6}^{6m+5} = u^2$ (Figure 4 shows these replacements). The variables $y_{4m+5}, y_{2m+4}, y_{2m+3}$ and y_{2m+2} are set to 0, and the variables $y_{2m+1}, \ldots, y_2$ are set to 1. The effect of the last replacements is that $2^{2m} > [y]_2^{2m+1} > w^2$, where w is defined as the integer with binary representation $[w]_0^{m-1}$. Figure 4 illustrates these replacements. Now we take a closer look at the product $u \cdot w$, where u is equal to $2^d + 1$. Figure 5 illustrates the composition of the product $u \cdot w$ (under certain assumptions on the number w).

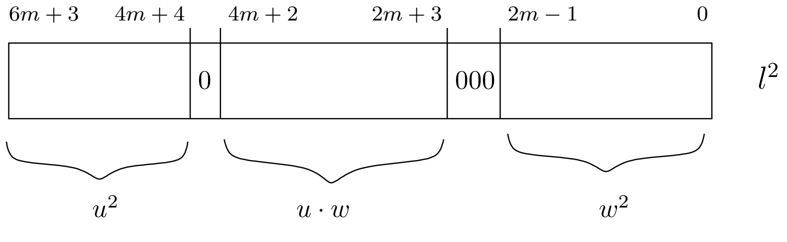

Fig. 3. *The bit composition of the number l^2.*

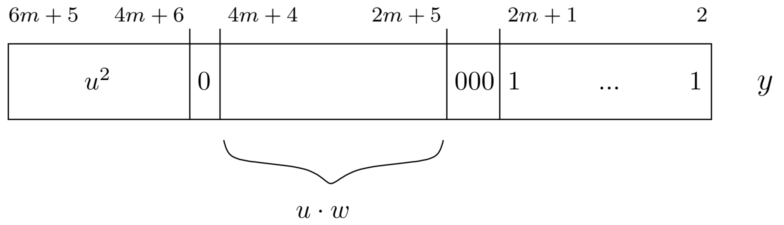

Fig. 4. *The effect of the replacements of some of the y-variables.*

A pair $(w_{i+d}, y_{2m+5+2d+i})$, $i \in I'$, is called (w, y)-pair. A (w, y)-pair is called separated with respect to S iff $w_{i+d} \in S$ and $y_{2m+5+2d+i} \notin S$ or vice versa.

Case 2.1:

In the following we prove the existence of a fooling set with at least $2^{m/24}$ elements. For this reason we choose a subfunction of $\mathrm{MUL}_{2n-1,n}$ such that the computation of this subfunction resembles the computation of the function $\overline{\mathrm{GT}_{m/24}}$.

There are at least $m/24$ separated (w, y)-pairs with respect to S. Similar to the proof of Lemma 1 we can show that there exists a fooling set of size at least $m/24$. The separated w-variables and their corresponding y-variables are called free. Furthermore, a variable $y_{n/2+i+d+1}$ for which the variable w_{i+d} is free is also called free. Remember that the variable $y_{n/2+i+d+1}$ is in S iff w_{i+d} is in S because of the definition of I'. Let $\min I'$ be the minimal and $\max I'$ be the maximal element of I'. In the rest of the proof, we choose for each variable $y_{n/2+i+d+1}$, where w_{i+d} is a free variable and $i \neq \min I'$, an assignment such that $y_{n/2+i+d+1} = w_{i+d} \oplus 1$ without further mentioning it.

- The variables $w_{\min I'+d}$, $y_{2m+5+2d+\min I'}$, and $y_{n/2+\min I'+d+1}$ are set to 1.
- The variables w_j and $y_{n/2+j+1}$, $0 \leq j < \min I' + d$, are set to 0.

- All other variables w_i which are not free are set to 0, their corresponding variables $y_{n/2+i+1}$ are set to 1.
- The other y-variables which are not free are replaced in the following way. The variable $y_{2m+5+\max I'+d+1}$ is set to 1, the remaining y-variables without the free variables to 0.

What is the effect of these replacements? Remember that $[w]_0^{m-1} = [x]_{n/2}^{n/2+m-1}$. We only consider assignments to the variables for which the following holds. If $[x]_{n/2}^{n/2+3m+1} = \ell$ then $[y]_{n/2+1}^{n/2+3m+2} = \overline{\ell} + 1$. Now iff $[y]_2^{6m+3}$ represents a number r, where $r \geq \ell^2$, the product $x \cdot y$ is greater than 2^{2n-1}. We take a closer look at the variables $y_2, \ldots, y_{6m+3}$. Figure 3 shows the composition of the number ℓ^2. One effect of our replacements is that $[y]_{4m+6}^{6m+5} = u^2$ and $[y]_2^{2m+1} > w^2$. Therefore, iff $[y]_{2m+5}^{4m+4}$ represents a number r', where $r' \geq u \cdot w$, $[y]_2^{6m+5}$ represents a number r, where $r \geq \ell^2$. Figure 5 illustrates the composition of the number $u \cdot w$ which is the same as the sum of w and $w \cdot 2^d$. Since we have replaced the variable $y_{2m+5+\max I'+d+1}$ by 1 and because of our other replacements, $[y]_2^{6m+5}$ represents a number r, where $r \geq \ell^2$, iff for each separated (w, y)-pair, the assignment to the variable $y_{2m+5+2d+i}$ is at least as large as the assignment to the variable w_{i+d}. Therefore, the considered subfunction resembles the function $\overline{\mathrm{GT}_{m/24}}$.

In the rest of the proof we show that all possible assignments b_S to the free variables in S together with all possible assignments b_r to the remaining free variables, such that $y_{2m+5+2d+i} = w_{i+d}$ for the free variables, are a fooling set of size at least $m/24$.

Together with the replacements to constants an assignment to the free w-variables and the corresponding assignment to the free y-variables can be seen as numbers $2^{n-1} + \ell 2^{n/2}$ and $2^n - \ell 2^{n/2+1} + c$, where $c > 4\ell^2$. Therefore, the product of the two numbers is larger than 2^{2n-1}.

Let b'_S and b''_S be two different assignments to the free w- and y-variables in S, b'_r and b''_r the two corresponding assignments to the remaining free w- and y-variables. Let $i_{max} := \max\{i \mid b'_S(w_{i+d}) \neq b''_S(w_{i+d})$ or $b'_S(y_{2m+5+2d+i}) \neq b''_S(y_{2m+5+2d+i})\}$. W.l.o.g. let $b'_S(w_{i_{max}+d}) \neq b''_S(w_{i_{max}+d})$ and $b'_S(w_{i_{max}+d}) > b''_S(w_{i_{max}+d})$. Since $b'_r(y_{2m+5+2d+i_{max}}) > b''_r(y_{2m+5+2d+i_{max}})$, we can conclude that b'_S together with b''_r and the first replacements to constants can be seen as numbers $2^{n-1} + \ell 2^{n/2}$ and $2^n - \ell 2^{n/2+1} + c$, where $c' < 4\ell^2$. Since

$$(2^{n-1} + \ell 2^{n/2}) \cdot (2^n - \ell 2^{n/2+1} + c') < 2^{2n-1},$$

we are done.

Case 2.2: In the following we prove the existence of a fooling set with at least $2^{m/24}$ elements. For this reason we choose a subfunction of $\mathrm{MUL}_{2n-1,n}$ such that the computation of this subfunction resembles the computation of the function $\overline{\mathrm{GT}^{**}_{m/24}}$.

There are less than $m/24$ separated (w, y)-pairs with respect to S. Let $I \subseteq I'$ be the set of indices such that $(w_{i+d}, y_{2m+5+2d+i})$, $i \in I'$, are not separated with respect to S. Obviously, $|I| \geq m/24$. Let $\min I$ and $\max I$ be the minimal resp. maximal element of I. For this reason we replace some of the variables in the following way.

- The variables $w_{\min I}$ and $y_{n/2+\min I+1}$ are set to 1, the variable $w_{\min I+d}$ is set to 0, the variable $y_{n/2+\min I+d+1}$ is set to 1,
- the variables w_i, $i < \min I$, are set to 0, the corresponding variables $y_{n/2+i+1}$ are set to 0,
- the variables w_i, $\min I < i < \max I$ and $i \notin I$ are set to 1, the corresponding variables $y_{n/2+i+1}$ are set to 0,
- all other variables w_j, $j \notin I$ and $j - d \notin I$, are set to 0, the corresponding variables $y_{n/2+j+1}$ are set to 1.

Furthermore, the variables $y_{2m+5+2d+i}$, $i \in I' \setminus I$, are replaced by 0. The variables $y_{2m+5+2d+i}$, $y_{n/2+i+1}$, and $y_{n/2+i+d+1}$, $i \in I$, are called free. The y-variables which are not free are replaced in the following way.

- The variables y_j, $2m + 5 + \min I \leq j \leq 2m + 5 + \max I$, are set to 1,
- the variables y_j, $2m + 5 + \min I + d \leq j \leq 2m + 5 + \max I + d$, are set to 1, and
- all other variables y_j with $2m + 5 \leq j \leq 6m + 5$ besides the free y-variables are set to 0.

The free y-variables are not separated from their corresponding w-variables, since we know that $w_i \in S$ and $y_{n/2+i+1} \in S$ or $w_i \notin S$ and $y_{n/2+i+1} \notin S$, where $i \in I$, because of the definition of I. The same holds for $w_{i+d}, y_{n/2+i+d+1}$, and $y_{2m+5+2d+i}$, where $i \in I$. In the rest of the proof we only consider assignments with the property that

- $y_{n/2+i+1} = w_i \oplus 1$,
- $y_{n/2+i+d+1} = w_{i+d} \oplus 1$, and
- $y_{2m+5+2d+i} = w_{i+d}$,

where $i \in I$, without further mentioning it.

In the following we prove that all possible assignments to the variables w_i, $i \in I$, together with the assignments to the variables w_{i+d}, $i \in I$, such that $w_{i+d} = w_i \oplus 1$ are a fooling set of size at least $m/24$. Together with the replacements to constants our assignments to the variables w_i, $i \in I$ or $i - d \in I$, can be seen as a number $2^{n-1} + \ell 2^{n/2}$. The corresponding assignments to the y-variables can be interpreted as number $2^n - \ell 2^{n/2+1} + c$, where $c > 4\ell^2$. Therefore, the product of the two numbers is larger than 2^{2n-1}. To see this we decompose ℓ into $u \cdot 2^{2m+2} + w$, where $u = [u]_0^{m-1}$ and $w = [w]_0^{m-1}$. The number c can be decomposed into

$$[y]_{4m+6}^{6m+5} \cdot 2^{4m+6} + [y]_{2m+5}^{4m+4} \cdot 2^{2m+5} + [y]_2^{2m+1} \cdot 2^2.$$

As mentioned before, $[y]_{4m+6}^{6m+5} = u^2$ and $w^2 < [y]_2^{2m+1} < 2^{2m}$ (see Figure 2). The number $[w]_0^{m-1}$ can be decomposed into $[w]_{\min I+d}^{\max I+d} \cdot 2^{\min I+d} + [w]_{\min I}^{\max I} \cdot 2^{\min I}$. Let $w' := [w]_{\min I}^{\max I}$ and $w'' := [w]_{\min I+d}^{\max I+d}$. Now the number $[y]_{2m+5}^{4m+4}$ can be decomposed into $w'' \cdot 2^{2m+5+2d+\min I} + (2^{\max I - \min I+1} - 1) \cdot 2^{2m+5+d+\min I} + (2^{\max I - \min I+1} - 1) \cdot 2^{2m+5+\min I}$. Iff $w' + w'' \leq 2^{\max I - \min I+1} - 1$, the number

$[y]_{2m+5}^{4m+4}$ is greater than $u \cdot w$ and altogether $[y]_2^{6m+5} > \ell^2$. Therefore, we can conclude $c > 4\ell^2$.

If $w' + w'' > 2^{\max I - \min I + 1} - 1$, the number $[y]_{2m+5}^{4m+4}$ is less than $u \cdot w$. Therefore, x can be seen as number $2^{n-1} + \ell' 2^{n/2}$ and y as $2^n - \ell' 2^{n/2+1} + c$, where $c < 4\ell'^2$. Since

$$(2^{n-1} + \ell' 2^{n/2}) \cdot (2^n - \ell' 2^{n/2+1} + c) < 2^{2n-1}$$

we are done.

Altogether we have shown that for an arbitrary variable order π the π-OBDD size for the most significant bit of multiplication is at least $2^{m/24}$. Considering the fact that $m := (n/6-3)/3 = n/18-1$ we obtain a lower bound of $2^{n/432-1} = \Omega(2^{n/432})$ on the OBDD complexity of $\mathrm{MUL}_{2n-1,n}$.

Now we present the ideas how to improve the lower bound on the OBDD complexity of $\mathrm{MUL}_{2n-1,n}$ up to $\Omega(2^{n/288})$. As we have seen in the proof of Claim 1 for a number $2^{n-1} + \ell 2^{n/2}$, the corresponding smallest number such that the product of the two numbers is at least 2^{2n-1} is $2^n - \ell 2^{n/2+1} + 4\ell^2 - \left\lfloor \frac{4\ell^3}{2^{n/2-1}+\ell} \right\rfloor$. For $\ell \leq 2^{n/4-3/2}$ and $\ell < 2^{3m+2}$, the number $\left\lfloor \frac{4\ell^3}{2^{n/2-1}+\ell} \right\rfloor$ is smaller than ℓ and therefore smaller than 2^{3m+2}.

Our aim is to enlarge m to $n/12 - 7/6$. The idea is to choose u and w such that $u^2 2^{4m+6} + w^{(2m+5)+m-1} + (w''+1)2^{(2m+5)+m-1-d} > 4\ell^2 - \left\lfloor \frac{4\ell^3}{2^{n/2-1}+\ell} \right\rfloor \geq u^2 2^{4m+6} + w^{(2m+5)+m-1} + w'' 2^{(2m+5)+m-1-d}$, where $\ell = u 2^{2(m+1)} + w$. As a result we can adapt our lower bound proof easily.

We choose $u := 2^{m-1} + 2^{m-1-d}$ instead of $u = 2^d + 1$, set the variable w_0 to 1, and adapt the settings to the corresponding variables, e.g., $y_{(2m+5)+m-1}$ and $y_{(2m+5)+m-2}$ are set to 1. The variables u_{m-1} and u_{m-1-d} are set to 1 and $[y]_{4m+6}^{6m+5} = u^2$. The proof of Case 2 has to be adapted to pairs $(w_{i+d}, y_{3m+4+i+d})$, $i \in I'$.

Using techniques from analytical number theory Sawitzki [14] has presented a lower bound of $\Omega(2^{n/6})$ on the size of π-OBDDs representing the most significant bit of integer multiplication for the variable order π where the variables are tested according to increasing significance, i.e. $\pi = (x_0, y_0, x_1, y_1, \ldots, x_{n-1}, y_{n-1})$. A larger lower bound can be proved in an easier way and without analytical number theory using the fact that for a number $2^{n-1} + \ell 2^{(1/2)n}$, $\ell \leq 2^{(1/4)n-1}$, the corresponding smallest number such that the product of the two numbers is at least 2^{2n-1} is $2^n - \ell 2^{n/2+1} + 4\ell^2 - \left\lfloor \frac{4\ell^3}{2^{n/2-1}+\ell} \right\rfloor$. Since

$$4\ell^2 - \left\lfloor \frac{4\ell^3}{2^{n/2-1}+\ell} \right\rfloor > 4(\ell-1)^2$$

for $\ell \leq 2^{(1/4)n-1}$, it is not difficult to construct a fooling set of size $2^{(1/4)n-1}$.

Furthermore, we only want to mention here that similar to Gergov's [9] generalization of Bryant's lower bound on the size of OBDDs for the middle bit of multiplication to arbitrary oblivious programs of linear length the result for the most significant bit of multiplication can be analogously extended.

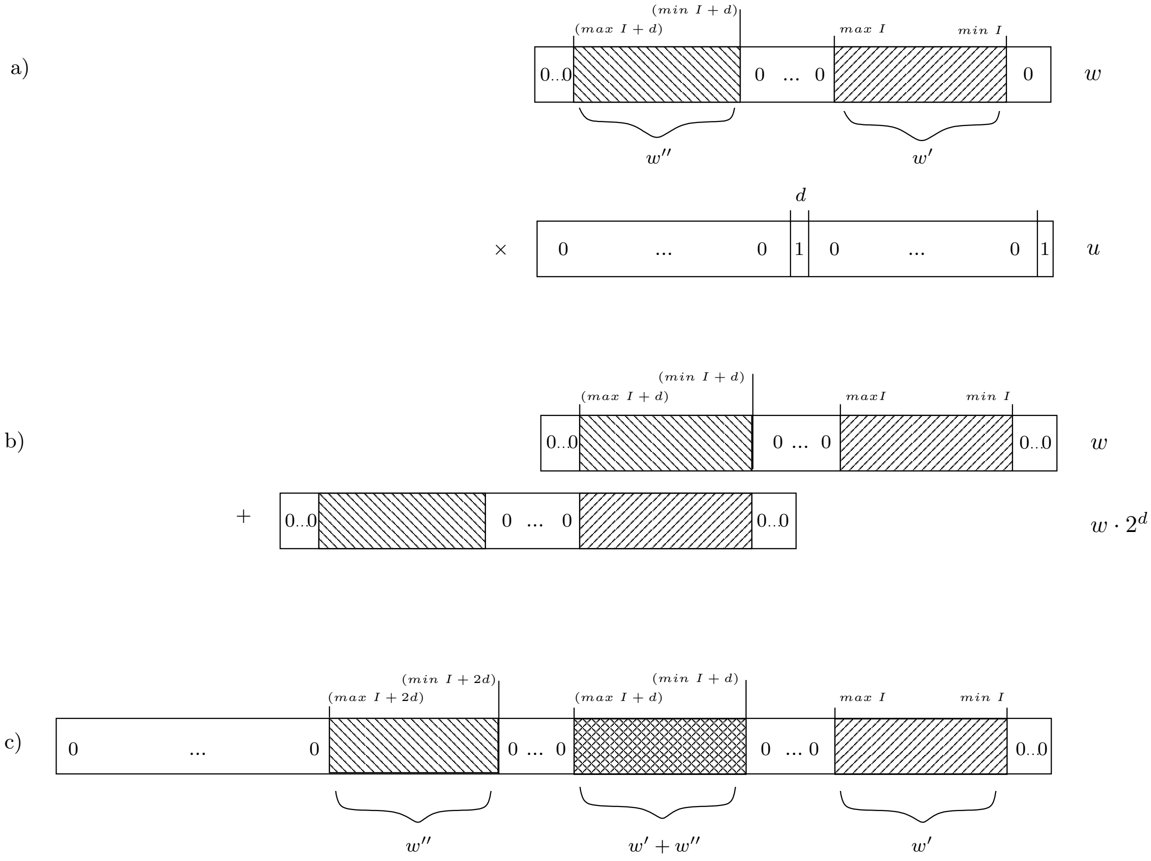

Fig. 5. *The product* $u \cdot w$ (*under the assumption that* $w' + w'' < 2^{\max I - \min I + 1}$).

Acknowledgement

The author would like to thank Martin Sauerhoff for careful reading the first version of the paper and for valuable suggestions which helped to improve the paper.

References

1. Amano, K. and Maruoka, A. (2007). Better upper bounds on the QOBDD size of integer multiplication. Discrete Applied Mathematics 155, 1224–1232.
2. Bollig, B. (2001). Restricted nondeterministic read-once branching programs and an exponential lower bound for integer multiplication. RAIRO Theoretical Informatics and Applications, 35:149–162.
3. Bollig, B. and Woelfel, P. (2001). A read-once branching program lower bound of $\Omega(2^{n/4})$ for integer multiplication using universal hashing. Proc. of 33rd STOC, 419–424.
4. Bollig, B., Waack, St., and Woelfel, P. (2006). Parity graph-driven read-once branching programs and an exponential lower bound for integer multiplication. Theoretical Computer Science 362, 86–99.
5. Bollig, B. and Woelfel, P. (2005). A lower bound technique for nondeterministic graph-driven read-once branching programs and its applications. Theory of Computing Systems 38, 671–685.
6. Bryant, R. E. (1986). Graph-based algorithms for Boolean manipulation. IEEE Trans. on Computers 35, 677–691.
7. Bryant, R. E. (1991). On the complexity of VLSI implementations and graph representations of Boolean functions with application to integer multiplication. IEEE Trans. on Computers 40, 205–213.
8. Führer, M. (2007). Faster integer multiplication. Proc. of 39th STOC, 57–66.
9. Gergov, J. (1994). Time-space trade-offs for integer multiplication on various types of input oblivious sequential machines. Information Processing Letters 51, 265–269.
10. Hromkovič, J. (1997). *Communication Complexity and Parallel Computing.* Springer.
11. Kushilevitz, E. and Nisan, N. (1997). *Communication Complexity.* Cambridge University Press.
12. Ponzio, S. (1998). A lower bound for integer multiplication with read-once branching programs. SIAM Journal on Computing 28, 798–815.
13. Sauerhoff, M. and Woelfel, P. (2003). Time-space trade-off lower bounds for integer multiplication and graphs of arithmetic functions. Proc. of 33rd STOC, 186–195.
14. Sawitzki, D. (2006). Exponential lower bounds on the space complexity of OBDD-based graph algorithms. Proc. of LATIN, LNCS 3831, 471-482.
15. Wegener, I. (1993). Optimal lower bounds on the depth of polynomial-size threshold circuits for some arithmetic functions. Information Processing Letters 46/2, 85–87.
16. Wegener, I. (2000). *Branching Programs and Binary Decision Diagrams - Theory and Applications.* SIAM Monographs on Discrete Mathematics and Applications.
17. Woelfel, P. (2005). New bounds on the OBDD-size of integer multiplication via universal hashing. Journal of Computer and System Science 71/4, 520–534.
18. Woelfel, P. (2002). On the complexity of integer multiplication in branching programs with multiple tests and in read-once branching programs with limited nondeterminism. Proc. of 17th Computational Complexity, 80–89.