Clean quantum and classical communication protocols

Harry Buhrman,[1] Matthias Christandl,[2] Christopher Perry,[2] and Jeroen Zuiddam[1]

[1] *QuSoft, CWI Amsterdam and University of Amsterdam, Science Park 123, 1098 XG Amsterdam, Netherlands*

[2] *QMATH, Department of Mathematical Sciences, University of Copenhagen, Universitetsparken 5, 2100 Copenhagen, Denmark*

By how much must the communication complexity of a function increase if we demand that the parties not only correctly compute the function but also return all registers (other than the one containing the answer) to their initial states at the end of the communication protocol? Protocols that achieve this are referred to as *clean* and the associated cost as the *clean communication complexity*. Here we present clean protocols for calculating the Inner Product of two n-bit strings, showing that (in the absence of pre-shared entanglement) at most $n+3$ qubits or $n+O(\sqrt{n})$ bits of communication are required. The quantum protocol provides inspiration for obtaining the optimal method to implement distributed *CNOT* gates in parallel whilst minimizing the amount of quantum communication. For more general functions, we show that nearly all Boolean functions require close to $2n$ bits of classical communication to compute and close to n qubits if the parties have access to pre-shared entanglement. Both of these values are maximal for their respective paradigms.

Introduction. In a communication task two players, Alice and Bob, receive inputs x and y and wish to calculate the value of some function f. To achieve this, messages will have to be exchanged between them and, depending on the resources available to them, these may consist of classical or quantum communication in the form of bits and qubits respectively. Typically in such scenarios one is interested in minimizing the amount of communication that has to take place to evaluate the function and the number of bits/qubits that must be exchanged to do this is referred to as the classical/quantum *communication complexity* [1, 2].

A protocol for calculating a function will act on three distinct types of registers. Each player will receive an input register, containing x or y, and an ancillary working space, initialized in some standard state such as a string of bits all set to 0, a number of qubits provided in the $|0\rangle$ state or possibly containing entangled states shared between the parties. The final type of register is the answer register which will contain the value of $f(x,y)$ at the end of the protocol. On the completion of a generic protocol for computing f, the input and ancillary registers will no longer be in their starting states and will depend upon both x and y.

However, leaving these registers in such states can be problematic. Firstly, if Alice and Bob wish to keep private the particular protocol that they ran, then discarding these unclean states may leak information regarding this to a third party. Secondly, in the quantum setting, if the players wish to run the protocol over a superposition of input states (perhaps as a subroutine of a larger computation), then allowing the ancillary registers to end up in some unclean, input dependent state and then discarding them can lead to a loss of coherence in the superposition over answers. Finally, the players' computational space may be in short supply and without knowing the registers' final states they cannot easily use them for future calculations.

To avoid such issues we can demand that a protocol (in addition to computing f) returns the input and ancillary registers to their starting state. Following [3], we call such a protocol *clean* and the minimum number of bits/qubits that a clean protocol needs to exchange to compute a given function is the *clean communication complexity*. We shall denote these quantities by $C_{clean}(f)$ and $Q_{clean}(f)$. In the case where the players have access to pre-shared entanglement (which they must restore at the end of the protocol), the associated cost will be written Q^{*}_{clean}. We focus on the scenario where the players must compute the function exactly.

In all three scenarios, an unclean communication protocol can be converted into a clean one at the cost of doubling the communication. To do this, the players run the unclean protocol, copy the output to another location and then run the unclean protocol backwards. At first glance it may appear that clean, classical protocols are even easier to construct: the players keep a copy of their input and then simply erase all ancillary bits once the protocol is complete. However, Landauer's principle [4–6] implies that such irreversible manipulations will generate heat or else cost work. As such, if one is interested in avoiding such costs, it makes sense to consider protocols where all operations must be reversible. In light of these constructions, it is natural to ask: do more efficient clean protocols, without this doubling in communication, exist?

We first focus on the clean communication complexity of computing the Inner Product of two distributed bit strings of length n, showing that (without pre-shared entanglement) this can be done by exchanging $n+3$ qubits. As a clean protocol for this function must exchange at least $n+1$ qubits, this is very close to tight. We also provide a clean, classical protocol that computes Inner Product while exchanging only $n+O(\sqrt{n})$ bits. This provides a saving over the most obvious protocol which, as we shall show, are close to optimal for the clean, classical computation of most functions.

A variation on our quantum protocol can be used to implement n copies of a $CNOT$ gate in parallel by exchanging $n+1$ qubits. In a quantum computing architecture consisting of distributed clusters of highly controllable qubits linked by quantum communication (such as that envisaged in [7]), it is prudent to minimize the number of qubits exchanged. Our implementation is optimal.

Next we turn to the clean communication complexity of random functions on inputs of length n. We show here that in contrast to Inner Product, nearly all functions are such that $C_{clean}(f)$ is close to the maximal $2n$: the simple method of generating clean protocols discussed above is near optimal. On the quantum side, we find that $Q^*_{clean}(f)$ is close to n for most functions. As superdense coding [8] allows all functions to be uncleanly computed while exchanging $\frac{n}{2}$ qubits when the players pre-share entanglement, this is again close to maximal. Whether similarly $Q_{clean}(f)$ is close to $2n$ remains an open question.

Clean Protocols. Clean protocols have a long history in proving bounds in the model of quantum communication complexity with free entanglement assistance [9]. For example, considering clean, quantum protocols for the Inner Product function was used to imply that any entanglement assisted quantum protocol for this function must use at least $\lceil n/2 \rceil$ qubits [3]. Clean protocols have also been used to lower bound the entanglement assisted, quantum communication complexity [10] and that, in this model of communication, most functions have complexity that scales linearly in n [11]. Cleanliness has also been used to analyze privacy amongst honest players [12], bound the amount of quantum communication required to implement distributed quantum computation [13] and for constructing resource inequalities that carefully account for the way protocols can be combined [14, 15].

More formally, a clean, quantum protocol for computing a function $f:\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined as follows [3]. The initial state at the beginning of the protocol is of the form:

$$|x\rangle_A |\vec{0}\rangle_{A_0} |y\rangle_B |\vec{0}\rangle_{B_0} |\Phi\rangle_{A_E B_E} |z\rangle_{B_{\text{ans}}}, \tag{1}$$

where $|x\rangle_A = \bigotimes_{i=1}^n |x_i\rangle_{A_i}$ and $|y\rangle_B = \bigotimes_{i=1}^n |y_i\rangle_{B_i}$ are Alice and Bob's respective inputs stored in n qubits, $|\vec{0}\rangle_{A_0}$ and $|\vec{0}\rangle_{B_0}$ their qubit ancillas, $|\Phi\rangle_{A_E B_E}$ their pre-shared entanglement (if supplied) and $|z\rangle_{B_{\text{ans}}}$ is the initial state of the answer register with $z \in \{0,1\}$. Throughout this paper we will assume that at the beginning and end of a protocol the answer register is held by Bob.

Players then take turns to act on their share of the qubits. In each turn a player will apply a unitary transformation to the qubits in their possession and then send some subset of them to the other player. The protocol computes f cleanly if the final state of the qubits is:

$$|x\rangle_A |\vec{0}\rangle_{A_0} |y\rangle_B |\vec{0}\rangle_{B_0} |\Phi\rangle_{A_E B_E} |z \oplus f(x,y)\rangle_{B_{\text{ans}}}, \tag{2}$$

where the addition in the answer register is modulo 2. Clean classical protocols are defined similarly but with registers and communication given in terms of bits rather than qubits and no entanglement. All transformations must be reversible.

Inner Product. The specific function that we shall focus on in this paper is the Inner Product function, IP_n. This is defined by:

$$\begin{aligned} IP_n &: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}, \\ IP_n(x,y) &= \sum_{i=1}^n x_i \cdot y_i \mod 2. \end{aligned} \tag{3}$$

It is well known that for both players to know the answer, at least n bits of classical communication are needed to (uncleanly) compute IP_n exactly [16, Example 1.29]. For quantum strategies in which the players pre-share entanglement, $\lceil \frac{n}{2} \rceil$ qubits must be sent [3] to achieve the same goal. In [3], it is also shown that clean, quantum protocols for computing IP_n must exchange at least n qubits. The quantum communication required to uncleanly compute IP_n without prior entanglement is unknown (though must lie between $\lceil \frac{n}{2} \rceil$ and n). For quantum protocols that are allowed to err with fixed probability less than 1/2, the complexity is still $\Omega(n)$ [17].

Here we examine the clean communication complexity of IP_n without entanglement assistance. To this end, we first consider the quantum communication complexity of implementing the transformation:

$$|x\rangle_A |y\rangle_B \mapsto (-1)^{x \cdot y} |x\rangle_A |y\rangle_B, \tag{4}$$

i.e. the distributed computation of the inner product of x and y in the phase. Such a transformation corresponds to performing *controlled-Z* gates across n pairs of qubits and by a suitable local basis change this can be converted into an implementation of n-fold *CNOTs*.

In [18] it was shown that 2 qubits of communication together with sharing 4 ebits is exactly equivalent as a resource to the ability to implement 2 $CNOT$ gates and sharing 4 ebits. As such, this provides a protocol for implementing IP_n in the phase using $n+8$ qubits of communication and 8 ancilla qubits (for even n). This can be adapted to give a protocol requiring $n+2$ qubits of communication for even n and $n+3$ qubits when n is odd. In the following lemma, we give an improved, optimal protocol:

Lemma 1. *The clean, quantum communication complexity of exactly implementing IP_n in the phase satisfies:*

$$Q_{clean}\left(IP_n^{phase}\right) = n+1. \tag{5}$$

One ancilla qubit is required.

(Without using ancilla qubits, $n+1$ qubits for odd n and $n+2$ for even n suffice.)

Proof. The $n+1$ qubit protocol for even n is as follows. Alice initially prepares an ancilla qubit in the state $|x_1\rangle$ and sends it to Bob who applies a phase of $(-1)^{x_1 \cdot y_1}$. He

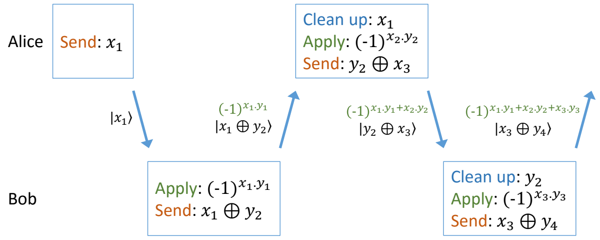

FIG. 1. *Clean, quantum protocol for calculating IP_n in the phase.* Here we illustrate the first 4 rounds of communication. In each round, a player cleans up the message they sent previously, applies the relevant global phase and communicates the next bit of their input string.

then adds y_2 to the communication qubit and sends it back to Alice in the state $|x_1 + y_2\rangle$. Now, Alice cleans up her previous communication by subtracting x_1 from the communication and then uses the value of y_2 to apply the phase $(-1)^{x_2 \cdot y_2}$. She then adds x_3 to the communication qubit to leave it in the state $|y_2 \oplus x_3\rangle$ and sends it back to Bob. A schematic of these first rounds is given in Figure 1.

The players then proceed similarly, with each round of communication being used to convey a new bit to the other party and send a received bit back in order to clean the ancilla qubit. After n rounds, the global phase will be $(-1)^{x \cdot y}$ and Alice will hold the communication qubit in the state $|y_n\rangle$. She sends this back to Bob who cleans it, completing the protocol using $n+1$ qubits of communication and the change in ownership of one ancilla qubit. For odd n, Alice will perform the final cleaning step. The protocol to implement the transformation without an ancilla qubit is given in Appendix B1.

The lower bound is proved in Appendix C3. It uses the concept of information complexity [19] to show that in a clean protocol for implementing Eq. (4) n bits of information must flow in each direction. Without pre-shared entanglement, we show that n qubits of communication cannot achieve this. □

The above lemma provides the optimal method for implementing n CZ gates in parallel while exchanging $n+1$ qubits. Such a protocol would prove useful for quantum computing architectures where quantum communication is used to interface and implement gates between clusters of highly controllable qubits. As an example, in quantum error correction one could imagine using the Steane code [20] to protect 2 logical qubits using 2 spatially separated clusters of 7 physical qubits. To implement a CZ gate between the logical qubits requires 7 CZs to be performed in parallel between the physical qubits.

Our protocol achieves this while exchanging only 8 qubits whereas the naive protocol would send 14. Protocols based solely on shared entanglement and classical communication [21–23] use 7 pairs of ebits, 14 bits of communication and the implementation of 14 measurements while their coherent counterpart [18] requires 1 shared ebit and 8 qubits of communication.

In Appendix B2 we give a clean quantum protocol for computing IP_n:

Theorem 2. *The clean, quantum communication complexity of exactly computing IP_n satisfies:*

$$n+1 \leq Q_{clean}\left(IP_n\right) \leq \begin{cases} n+3 & \text{for } n \text{ odd,} \\ n+2 & \text{for } n \text{ even.} \end{cases} \tag{6}$$

No ancilla qubits are required.

By adapting the protocol from Lemma 1, IP_n can be computed cleanly using 2 qubits and $n+1$ bits. We give this protocol in Appendix B3.

Our novel quantum communication protocols inspire a classical protocol for Inner Product (given in Appendix B4) which is near optimal and for which only the naive $2n$ protocol was known before:

Theorem 3. *The clean, classical communication complexity of exactly computing IP_n satisfies:*

$$n+1 \leq C_{clean}\left(IP_n\right) \leq n + 4\sqrt{n} + \frac{1}{\sqrt{n}-1} + 2. \tag{7}$$

No ancilla bits are required.

Generic functions. In contrast to Theorem 3, we will show that nearly all Boolean functions on n-bit inputs require $2n - O(\log n)$ bits of classical communication to compute cleanly. The proof follows from the following two lemmas. In what follows, X and Y are the random variables for Alice and Bob's inputs and A and B are the random variables received by Alice and Bob respectively over the course of the protocol. By $|a|$ and $|b|$ we denote the number of bits received by Alice and Bob.

Lemma 4. *Consider picking uniformly at random a Boolean function f_n on n-bit inputs. Then with probability $1 - o(1)$, all protocols that compute f_n exactly are such that either:*

1. *Alice must receive:*

$$|a| \geq n - \log\left(n+1\right) - 2, \tag{8}$$

bits and there exists a uniform distribution over at least half the pairs of inputs such that:

$$I\left(Y : AX\right) \geq n - \log\left(n+1\right) - 3. \tag{9}$$

Or:

2. *Bob must receive:*

$$|b| \geq n - \log\left(n+1\right) - 2, \tag{10}$$

bits and there exists a uniform distribution over at least half the pairs of inputs such that:

$$I\left(X : BY\right) \geq n - \log\left(n+1\right) - 3. \tag{11}$$

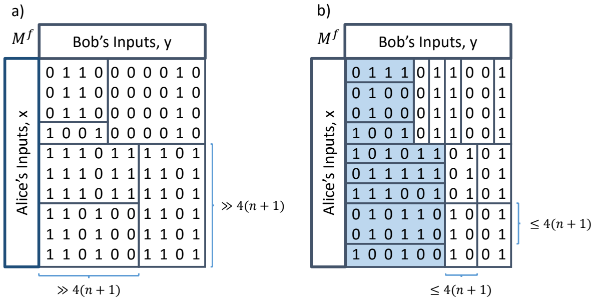

FIG. 2. *Partitions of the communication matrix into rectangles.* Note that knowledge of y, together with knowledge of which rectangle the players' input pair belongs to, allows Bob to correctly deduce the value of $f(x,y)$. a) As there exists a protocol for computing f that partitions M^f into large rectangles, the Kolmogorov complexity of M^f is low. b) For M^f to have high Kolmogorov complexity, all protocols for computing f must partition M^f into either very narrow or very thin rectangles. To produce the bound in Eq. (9), we take a distribution over the shaded rectangles.

Proof. The full proof is given in Appendix C1a. To prove the first two bounds, begin by noting that the communication matrix M^f (defined by $M^f_{xy} = f(x,y)$) of a random Boolean function has large *Kolmogorov complexity* with high probability. However, a classical protocol for computing f partitions the matrix into *rectangles* (see Appendix A2), each of which has low Kolmogorov complexity. If one of these rectangles is large enough (which happens when the amount of communication that takes place in one direction is small), then the Kolmogorov complexity of M^f will also be low. Such an M^f is shown in Figure 2a. Comparing these two statements leads to the bounds on $|a|$ and $|b|$.

These bounds imply that the rectangles induced by any protocol for computing most f_n must either be very short or very thin as shown in Figure 2b. In fact, they cannot be larger than $4(n+1) \times 2^n$ nor $2^n \times 4(n+1)$. Either at least half the inputs will belong to very short rectangles or at least half the inputs will belong to very thin ones. By taking a distribution over the larger set, we induce a direction into the communication that occurs in the protocol to ensure that one of Eqs. (8) and (10) holds and bound the related mutual information. For example, consider the case where more than half the input pairs lie in rectangles of size less than $2^n \times 4(n+1)$ (as shown in the figure) and the distribution over x and y is formed by picking Alice and Bob's inputs uniformly at random from such rectangles. Then, at the end of the protocol, Alice will know that Bob received one of at most $4(n+1)$ inputs and Eq. (8) will hold. Hence:

$$I(Y:AX) = H(Y) - H(Y|AX) \geq n - \log(n+1) - 3,$$

as required. □

The previous lemma indicates that to compute most functions, either Alice or Bob must receive close to the entirety of the other player's input. We now show that a similar amount of information (and hence communication) must flow back in the other direction to make the protocol clean.

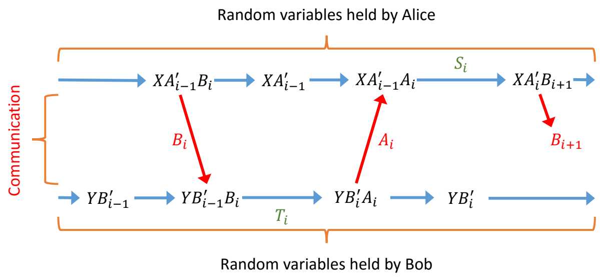

FIG. 3. *Schematic of a classical communication protocol.* Here we show how the random variables held by each player change during round i of a communication protocol. Primed variables denote local memories while non-primed variables are communication. Each player uses a deterministic, reversible function (S_i and T_i) to determine their next message and update their local memory.

Lemma 5. *Let f be a Boolean function and its inputs be chosen according to some distribution. Then, in a clean protocol for exactly computing f:*

$$|b| \geq I(Y:XA) - I(X:Y), \tag{12}$$

and:

$$|a| \geq I(X:YB) - I(X:Y) - 1. \tag{13}$$

Proof. The full proof can be found in Appendix C1b. It revolves around considering a protocol as r rounds in which each player speaks (see Figure 3). The bounds are then constructed by noting that in each round the players' messages are produced by a deterministic, reversible function of their inputs, local memory (denoted by A'_i and B'_i) and the last message received. To obtain (for example) Eq. (13), the chain rule for the conditional mutual information can then be used to write:

$$\begin{aligned} I(X:YB) =& I(X:Y) + I(X:B'_r|Y) \\ &+ \sum_{i=1}^{r} I(X:A_i|YB'_iB_{i+1}\dots B_r) \\ \leq& I(X:Y) + 1 + |a|, \end{aligned}$$

where in the last line we have used the fact that that the protocol is clean and that the conditional mutual information can be upper bounded by the number of bits contained in A_i. □

Combining these two lemmas, together with the fact that $I(X:Y) \leq 1$ for uniform distributions over at least half the possible inputs, we obtain:

Theorem 6. *Consider exactly computing a Boolean function f_n on n-bit inputs that has been picked uniformly at random. Then with probability $1 - o(1)$:*

$$C_{clean}(f_n) \geq 2n - 2\log(n+1) - 7. \tag{14}$$

In the case of quantum protocols, a similar result holds in the entanglement assisted case. Proving this result (Appendix C2) makes use of the fully quantum notion of information complexity introduced in [19]. The proof follows a similar structure to the classical result: arguing that for most functions close to n bits of information has to flow from Alice to Bob and for the protocol to be clean an equivalent amount of information has to be returned.

Theorem 7. *Consider exactly computing a Boolean function f_n on n-bit inputs that has been picked uniformly at random. Then with probability $1 - o(1)$:*

$$Q^*_{clean}(f_n) \geq n - \log n. \tag{15}$$

Conclusion. In this paper we have initiated the study of how big an overhead in communication cleanliness requires. For the Inner Product function (and the task of implementing n CZ gates in parallel) we have exhibited quantum and classical protocols for which the overhead is low. For most Boolean functions however, we have shown that the additional cost incurred by demanding cleanliness is close to maximal for the classical and entanglement assisted complexities. Many questions remain. For example, what are the clean complexities of other notable functions such as Equality and Disjointness?

As Theorems 6 and 7 show that the clean, classical and entanglement assisted communication complexity for most functions is close to maximal, one can ask: does something similar hold for $Q_{clean}(f)$? We leave this as an open question but conjecture it to be close to $2n$ as Inner Product appears somewhat special in its ability to reuse a single ebit efficiently. However, the concept of information cost is blind to sending ebits so the technique used for the entanglement assisted case does not immediately generalize to proving a bound potentially larger than n.

ACKNOWLEDGMENTS

We thank Aram Harrow for helpful discussions and for bringing [18] to our attention. HB was partially funded by the European Commission, through the SIQS project and by the Netherlands Organisation for Scientific Research (NWO) through gravitation grant Networks. MC and CP acknowledge financial support from the European Research Council (ERC Grant Agreement no 337603), the Danish Council for Independent Research (Sapere Aude), VILLUM FONDEN via the QMATH Centre of Excellence (Grant No. 10059) and the Swiss National Science Foundation (project no PP00P2_150734). JZ is supported by NWO through the research programme 617.023.116 and by the European Commission through the SIQS project.

[1] A. C.-C. Yao, in *Proceedings of the eleventh annual ACM symposium on Theory of computing* (ACM, 1979) pp. 209–213.

[2] A. C.-C. Yao, in *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on* (IEEE, 1993) pp. 352–361.

[3] R. Cleve, W. Van Dam, M. Nielsen, and A. Tapp, in *Quantum Computing and Quantum Communications* (Springer, 1999) pp. 61–74.

[4] R. Landauer, IBM journal of research and development **5**, 183 (1961).

[5] C. Bennett, Maxwells Demon. Entropy, Information, Computing , 197 (1973).

[6] C. H. Bennett, Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics **34**, 501 (2003).

[7] D. Kielpinski, C. Monroe, and D. J. Wineland, Nature **417**, 709 (2002).

[8] C. H. Bennett and S. J. Wiesner, Physical Review Letters **69**, 2881 (1992).

[9] R. Cleve and H. Buhrman, Physical Review A **56**, 1201 (1997).

[10] H. Buhrman and R. de Wolf, in *Computational Complexity, 16th Annual IEEE Conference on, 2001.* (IEEE, 2001) pp. 120–130.

[11] A. Montanaro and A. Winter, in *Automata, Languages and Programming* (Springer, 2007) pp. 122–133.

[12] H. Klauck, in *STACS 2002* (Springer, 2002) pp. 335–346.

[13] M. A. Nielsen, C. M. Dawson, J. L. Dodd, A. Gilchrist, D. Mortimer, T. J. Osborne, M. J. Bremner, A. W. Harrow, and A. Hines, Physical Review A **67**, 052301 (2003).

[14] A. W. Harrow and P. W. Shor, Information Theory, IEEE Transactions on **56**, 462 (2010).

[15] A. W. Harrow, “Entanglement spread and clean resource inequalities,” in *XVIth International Congress on Mathematical Physics* (World Scientific, 2012) Chap. 53, pp. 536–540.

[16] E. Kushilevitz and N. Nisan, *Communication complexity* (Cambridge University Press, 1997).

[17] I. Kremer, *Quantum Communication*, Master’s thesis, The Hebrew University of Jerusalem (1995).

[18] A. Harrow, Physical Review Letters **92**, 097902 (2004).

[19] D. Touchette, in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing* (ACM, 2015) pp. 317–326.

[20] A. Steane, in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 452 (The Royal Society, 1996) pp. 2551–2577.

[21] D. Gottesman, arXiv preprint quant-ph/9807006 (1998).

[22] J. Eisert, K. Jacobs, P. Papadopoulos, and M. Plenio, Physical Review A **62**, 052317 (2000).

[23] D. Collins, N. Linden, and S. Popescu, Physical Review A **64**, 032302 (2001).