

# Быстрые алгоритмы умножения и деления натуральных чисел с помощью клеточных автоматов с локаторами

Э. Э. Гасанов<sup>1</sup>, Б. Ф. Хайбуллин<sup>2</sup>

Для умножения и деления  $n$ -значных натуральных чисел известны алгоритмы со сложностью порядка  $n^{\log_2 3}$  и даже порядка  $n^{\log n}$ . В данной работе предложен алгоритм умножения  $n$ -значных натуральных чисел за  $2n + 2$  такта. Здесь под значностью числа  $a$  понимается число  $\lceil \log_2 a \rceil$ . Для деления натуральных чисел с остатком предложен алгоритм с временем работы  $3n + 8$  тактов, где  $n$  — значность частного. Предложенные алгоритмы в качестве вычислителей используются двумерные клеточные автоматы с локаторами.

**Ключевые слова:** умножение натуральных чисел, деление натуральных чисел, клеточные автоматы с локаторами.

## Введение

Пусть  $a$  и  $b$  два натуральных числа, двоичная запись которых содержит по порядку  $n$  разрядов. Наиболее известный и быстрый алгоритм умножения таких чисел был предложен А.А.Карацубой [1], и он имеет сложность  $O(n^{\log_2 3})$ . Более быстрым по порядку алгоритмом умножения, является алгоритм Шёнхаге-Штрассена [2]. Его сложность  $O(n \cdot \log n \cdot \log \log n)$ . Но на практике алгоритм Шёнхаге-Штрассена быстрее алгоритма Карацубы, только если значность числа более 10 тысяч. Еще более быстрым по порядку является алгоритм Фюрера [3], но его преимущество может проявиться при значности чисел более  $10^{13}$ . Относительно недавно появился алгоритм Харвея-Хоевена [4] со сложностью  $O(n \log n)$ .

Для деления натуральных чисел с остатком известен алгоритм Бурникеля-Циглера [5]. Он использует внутри себя алгоритм умножения. Если в качестве алгоритма умножения взять алгоритм Карацубы, то вычислительная сложность алгоритма Бурникеля-Циглера будет  $O(n^{\log_2 3})$ , а если использовать алгоритм Шёнхаге-Штрассена, то сложность алгоритма Бурникеля-Циглера будет  $O(n \cdot \log^2 n \cdot \log \log n)$ .

---

<sup>1</sup>Гасанов Эльяр Эльдарович — зав. каф. математической теории интеллектуальных систем мех.-мат. ф-та МГУ, e-mail: el\_gasanov@mail.ru.

*Gasanov Elyar Eldarovich* — Head of Chair Mathematical Theory of Intellegent Systems, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics.

<sup>2</sup>Хайбуллин Бакир Фаридович — ведущий программист в ООО "Elius", г. Ташкент, Узбекистан, e-mail: bakir\_k@mail.ru.

*Khaybullin Bakir Faridovich* — lead programmer at Elius LLC, Tashkent, Uzbekistan.

В данной работе предлагаются алгоритмы решения задач умножения и деления с остатком  $n$ -значных натуральных чисел с помощью клеточных автоматов с локаторами.

Приведем неформальное описание двумерного клеточного автомата с локаторами.

Расположим в каждой клетке плоской решетки  $\mathbb{Z}^2$  один и тот же автомат с локаторами. Понятие локатора определим чуть позже, сейчас важно, что каждый локатор в каждый момент принимает некоторое значение. Автомат имеет функцию перехода, которая по состоянию соседей автомата и по значениям локаторов в текущий момент определяет состояние автомата в следующий момент. Кроме того, у автомата есть функция вещания, которая по состояниям соседей автомата и по значениям локаторов вычисляет сигнал вещания, который передается в эфир. Сигналы вещания образуют конечную аддитивную коммутативную полугруппу, а эфир представляет собой потенциально бесконечный сумматор сигналов элементарных автоматов, где в качестве оператора суммы выступает определяющая операция данной полугруппы. Каждый локатор представляет собой некоторый телесный угол с вершиной в позиции автомата, а значением локатора в текущий момент является сумма сигналов вещания всех автоматов, попадающих в этот телесный угол. Отметим, что в область суммирования локатора не входит вершина телесного угла. т.е. мы сигнал вещания, посылаемый данным автоматом, не включаем в сумму.

В наших алгоритмах будут использоваться один полный локатор, который представляет собой двумерную плоскость с выколотым началом координат, и 8 локаторов, представляющих собой лучи, направленные на север, северо-восток, восток, юго-восток, юг, юго-запад, запад и северо-запад.

В работе показано, что с помощью таких двумерных клеточных автоматов с локаторами можно решить задачу умножения и деления  $n$ -разрядных чисел за время порядка  $n$ .

Ранее похожий алгоритм умножения был доложен на конференции [6].

## 1. Постановка задачи и формулировка результатов

Понятие клеточного автомата с локаторами введено в работе Э. Э. Гасанова [7]. В работе Г. В. Калачева [8] были выявлены некоторые неточности, приведенного в [7] определения. Точное формальное определение клеточного автомата с локаторами можно найти в работах Д. Э. Ибрагимовой [9] и Э. Э. Гасанова [10]. Здесь мы не будем приводить это определение, а дадим определение двумерного клеточного автомата с

9 локаторами, с помощью которого задачу умножения и деления чисел можно решить за линейное время.

В общем случае локатор — это телесный угол, границы которого являются частями гиперплоскостей, задаваемых линейными уравнениями с целыми коэффициентами. В нашем случае мы будем рассматривать множество из 9 телесных углов

$$L = \{\Omega, \mathcal{N}, \mathcal{NE}, \mathcal{E}, \mathcal{SE}, \mathcal{S}, \mathcal{SW}, \mathcal{W}, \mathcal{NW}\}, \quad (1)$$

где  $\Omega = \mathbb{Z}^2 \setminus \{(0, 0)\}$  — называется полным,  $\mathbb{Z}^2$  — множество двумерных векторов с целыми координатами,  $\mathcal{N} = \{(x, y) : x = 0, y > 0\}$  — назовем “север”,  $\mathcal{NE} = \{(x, y) : y = x, x > 0\}$  — назовем “северо-восток”,  $\mathcal{E} = \{(x, y) : y = 0, x > 0\}$  — назовем “восток”,  $\mathcal{SE} = \{(x, y) : y = -x, x > 0\}$  — назовем “юго-восток”,  $\mathcal{S} = \{(x, y) : x = 0, y < 0\}$  — назовем “юг”,  $\mathcal{SW} = \{(x, y) : y = x, x < 0\}$  — назовем “юго-запад”,  $\mathcal{W} = \{(x, y) : y = 0, x < 0\}$  — назовем “запад”,  $\mathcal{NW} = \{(x, y) : y = -x, x < 0\}$  — назовем “северо-запад”.

Двумерным клеточным автоматом с 9 локаторами называется восьмерка  $\sigma = (\mathbb{Z}^2, Q, V, G, +, L, \varphi, \psi)$ , где  $Q$  — некоторое конечное множество, называемое *множеством состояний*; в множестве  $Q$  выделено одно состояние  $q_0$ , называемое *состоянием покоя*;  $V = (\alpha_1, \dots, \alpha_{h-1})$  — упорядоченный набор попарно различных векторов из  $\mathbb{Z}^2$ ;  $G$  — некоторое конечное множество, “+” — операция на  $G$  такая, что  $(G, +)$  — коммутативная полугруппа с нейтральным элементом  $e \in G$ ;  $L$  — упорядоченный набор телесных углов, задаваемых выражением (1);  $\varphi$  — функция, зависящая от переменных  $x_0, x_1, \dots, x_{h-1}, z_0, z_1, \dots, z_8$ ;  $\varphi : Q^h \times G^9 \rightarrow Q$ ,  $\varphi(q_0, e) = q_0$ ;  $q_0 = (q_0, \dots, q_0) \in Q^h$ ,  $e = (e, \dots, e) \in G^9$ ;  $\psi$  — функция зависящая от переменных  $x_0, x_1, \dots, x_{h-1}, z_0, z_1, \dots, z_8$ ;  $\psi : Q^h \times G^9 \rightarrow G$ ;  $\psi(q_0, e) = e$ . Элементы множества  $\mathbb{Z}^2$  называются *ячейками* клеточного автомата  $\sigma$ ; элементы множества  $Q$  называются *состояниями ячейки* клеточного автомата  $\sigma$ ; набор  $V$  называется *шаблоном соседства* клеточного автомата  $\sigma$ ; элементы множества  $G$  называются *сигналами вещания*; набор  $L$  называется *шаблоном локаторов* клеточного автомата  $\sigma$ ; функция  $\varphi$  называется *локальной функцией переходов* автомата  $\sigma$ ; функция  $\psi$  называется *функцией вещания* автомата  $\sigma$ ; переменные  $x_0, x_1, \dots, x_{h-1}$  принимают значения из  $Q$ , переменные  $z_0, z_1, \dots, z_8$  принимают значения из  $G$ . Состояние  $q_0$  интерпретируется как *состояние покоя*, а условие  $\varphi(q_0, e) = q_0$  — как *условие сохранения состояния покоя*. Ячейки, находящиеся в состоянии отличном от  $q_0$ , будем называть *активными*. Условие  $\psi(q_0, e) = e$  означает, что ячейка в состоянии покоя, не имеющая активных соседей и не получающая сигналов из эфира посылает в эфир нейтральный элемент, что можно интерпретировать как то, что она не посылает сигналы в эфир.

Здесь нам нужно было вводить упорядочение шаблона соседства  $V$  и шаблона локаторов  $L$  для того, чтобы установить взаимно однозначное соответствие между векторами из  $V$  и телесными углами из  $L$  и переменными локальной функции переходов  $\varphi$  и функции вещания  $\psi$  соответственно  $x_0, x_1, \dots, x_{h-1}$  и  $z_0, z_1, \dots, z_8$ . Это соответствие можно сделать более явным, если индексировать переменные функций  $\varphi$  и  $\psi$  самими векторами и телесными углами, т.е. считать, что локальная функция переходов  $\varphi$  и функции вещания  $\psi$  зависят от переменных  $x_0, x_{\alpha_1}, \dots, x_{\alpha_{h-1}}, z_\Omega, z_{\mathcal{N}} \dots, z_{\mathcal{NW}}$ , здесь индекс первой переменной есть нулевой вектор  $0 = (0, 0) \in \mathbb{Z}^2$ . Если договориться так индексировать переменные локальной функции переходов и функции вещания, то их можно записывать в любом порядке, и тогда можно воспринимать шаблон соседства и шаблон локаторов как просто множества, а не упорядоченный набор. В дальнейшем мы будем индексировать переменные локальной функции переходов и функции вещания векторами из шаблона соседства и телесными углами из шаблона локаторов.

При этом мы часто будем опускать в индексах внешние круглые скобки у векторов. Например, если  $h = 2$ ,  $q = 2$  и  $V = \{(-1, 0), (1, 0)\}$ , то пример локальной функции переходов может выглядеть так:  $\varphi = x_{-1,0} \& z_\Omega \vee x_{1,0} \& z_{\mathcal{N}}$ .

Если  $\alpha \in \mathbb{Z}^2$  и  $\nu$  — телесный угол из  $L$ , то через  $\nu(\alpha)$  обозначим телесный угол, полученный параллельным переносом телесного угла  $\nu$  на вектор  $\alpha$ , т.е. вершиной телесного угла  $\nu(\alpha)$  является точка  $\alpha$ .

Если  $\alpha \in \mathbb{Z}^2$  — ячейка клеточного автомата  $\sigma$ , то множество  $V(\alpha) = \{\alpha, \alpha + \alpha_1, \dots, \alpha + \alpha_{h-1}\}$  называется *окрестностью ячейки*  $\alpha$ , а множество  $L(\alpha) = \{\Omega(\alpha), \mathcal{N}, \dots, \mathcal{NW}(\alpha_m)\}$  называется *локаторами ячейки*  $\alpha$ .

*Состоянием клеточного автомата с локаторами*  $\sigma$  назовем пару  $(g, f)$ , где  $g$  — произвольная функция, определенная на множестве  $\mathbb{Z}^2$ , принимающая значения из  $G$ , называемая *состоянием эффира*,  $f$  — произвольная функция, определенная на множестве  $\mathbb{Z}^2$ , принимающая значения из  $Q$  и называемая *распределением состояний клеточного автомата с локаторами*  $\sigma$ . Такую пару функций можно интерпретировать как некую мозаику, получающуюся в двумерном пространстве приписыванием каждой точке с целочисленными координатами некоторого сигнала из  $G$  и некоторого состояния из  $Q$ . Множество всевозможных состояний клеточного автомата с локаторами обозначим  $\Sigma$ .

Если  $\alpha \in \mathbb{Z}^2$ ,  $(g, f)$  — состояние клеточного автомата с локаторами  $\sigma$ , то значение  $g(\alpha)$  назовем *сигналом ячейки*  $\alpha$ , определяемым состоянием  $(g, f)$ , а значение  $f(\alpha)$  — *состоянием ячейки*  $\alpha$ , определяемым состоянием  $(g, f)$ .

Для каждого  $\nu \in L$

$$s_\nu(\alpha) = \sum_{\beta \in \nu(\alpha) \cap \mathbb{Z}^2} g(\beta) \quad (2)$$

назовем *значением локатора*  $\nu$ , определяемым состоянием  $(g, f)$ . Здесь суммирование сигналов осуществляется с помощью определяющей операции  $+$  полугруппы  $G$ . Отметим, что в формулах (2) используются формально бесконечные суммы, и, чтобы они были определены, мы либо будем считать, что только конечное число слагаемых в суммах отлично от нейтрального элемента, либо предположим, что полугруппа  $(G, +)$  является идемпотентным моноидом, т.е. для любого  $h \in G$  выполнено  $h + h = h$ .

На множестве  $\Sigma$  определим *глобальную функцию переходов*  $\Phi_\sigma$  клеточного автомата с локаторами  $\sigma$ , полагая  $\Phi_\sigma(g, f) = (g', f')$ , где  $(g, f), (g', f') \in \Sigma$  и для любой ячейки  $\alpha \in \mathbb{Z}^k$  выполняются тождества

$$f'(\alpha) = \varphi(f(\alpha), f(\alpha + \alpha_1), \dots, f(\alpha + \alpha_{h-1}), s_\Omega(\alpha), s_{\mathcal{N}} \dots, s_{\mathcal{N}\mathcal{W}}(\alpha)), \quad (3)$$

$$g'(\alpha) = \psi(f(\alpha), f(\alpha + \alpha_1), \dots, f(\alpha + \alpha_{h-1}), s_\Omega(\alpha), s_{\mathcal{N}} \dots, s_{\mathcal{N}\mathcal{W}}(\alpha)). \quad (4)$$

Содержательная интерпретация отображения  $\Phi_\sigma$  такова, что сигнал каждой ячейки и состояние каждой ячейки "после перехода" определяется по состоянию упорядоченной окрестности ячейки и по значениям локаторов "до перехода" с помощью законов  $\varphi$  и  $\psi$  одинаково для всех ячеек.

*Поведениями клеточного автомата с локаторами*  $\sigma$  назовем такие последовательности  $(g_0, f_0), (g_1, f_1), (g_2, f_2), \dots$  его состояний, для которых выполняется  $(g_{i+1}, f_{i+1}) = \Phi_\sigma(g_i, f_i)$  для всех  $i = 0, 1, 2, \dots$ , причем  $(g_i, f_i)$  называется *состоянием клеточного автомата с локаторами*  $\sigma$  в момент  $i$ , а  $(g_0, f_0)$  называется *начальным состоянием клеточного автомата с локаторами*  $\sigma$ .

Состояние клеточного автомата, у которого лишь конечное число ячеек находится в отличном от состояния покоя  $g_0$ , и сигналы лишь конечного числа ячеек не равны нейтральному элементу  $e$ , назовем *конфигурацией*. Множество конфигураций будем обозначать через  $\Sigma'$ .

Определим задачу умножения чисел  $a$  и  $b$  для клеточного автомата с локаторами. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами  $(0, 0)$  находится в состоянии, которое можно назвать "начало координат", ячейка с координатами  $(a, 0)$  находится в состоянии, которое можно назвать "первый сомножитель", а ячейка с координатами  $(0, b)$  находится в состоянии, которое можно назвать "второй сомножитель". Клеточный автомат решает задачу умножения чисел, если в финальной

конфигурации ячейка с координатами  $(a \cdot b, 0)$  перейдет в состояние “результат умножения”, а все остальные ячейки, кроме  $(0, 0)$ , перейдут в состояние покоя.

Справедлива следующая теорема, доказанная Э. Э. Гасановым.

**Теорема 1.** *Существует двумерный клеточный автомат с 9 локаторами, который решает задачу умножения чисел  $a$  и  $b$  за время  $2 \lceil \log_2 a \rceil + 2$ .*

Здесь если  $x$  — вещественное число, то  $\lceil x \rceil$  — это наименьшее целое неменьшее чем  $x$ .

Определим задачу деления чисел  $a$  и  $b$  с остатком для клеточного автомата с локаторами. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами  $(0, 0)$  находится в состоянии, которое можно назвать “начало координат”, ячейка с координатами  $(a, 0)$  находится в состоянии, которое можно назвать “делимое”, а ячейка с координатами  $(0, b)$  находится в состоянии, которое можно назвать “делитель”. Пусть  $c = \lfloor a/b \rfloor$  — целая часть от деления  $a$  на  $b$ ,  $d = a \bmod b$  — остаток от деления  $a$  на  $b$ . Клеточный автомат решает задачу деления чисел, если в финальной конфигурации ячейка с координатами  $(c, 0)$  перейдет в состояние “частное”, ячейка с координатами  $(0, d)$  перейдет в состояние “остаток”, а все остальные ячейки, кроме  $(0, 0)$ , перейдут в состояние покоя.

Справедлива следующая теорема, доказанная Б. Ф. Хайбуллиным.

**Теорема 2.** *Существует двумерный клеточный автомат с 9 локаторами, который решает задачу деления чисел  $a$  и  $b$  с остатком за время  $3 \lceil \log_2(a/b) \rceil + 8$ .*

## 2. Вспомогательные задачи

### 2.1. Удвоение числа

Задача удвоения числа  $a$  и  $b$  состоит в следующем. В начальной конфигурации только 2 ячейки находятся не в состоянии покоя, а именно ячейка с координатами  $(0, 0)$  находится в состоянии “начало координат”, а ячейка с координатами  $(a, 0)$  находится в состоянии “аргумент”. Надо, чтобы в финальной конфигурации ячейка с координатами  $(2a, 0)$  перешла в состояние “результат”, а все остальные ячейки, кроме  $(0, 0)$ , оказались в состоянии покоя.

Решить эту задачу можно следующим образом. В начальный момент “начало координат” и “аргумент” подают в эфир сигнал “такт 1”. Ячейка, которая услышит этот сигнал в локаторы юг и юго-восток (а это ячейка

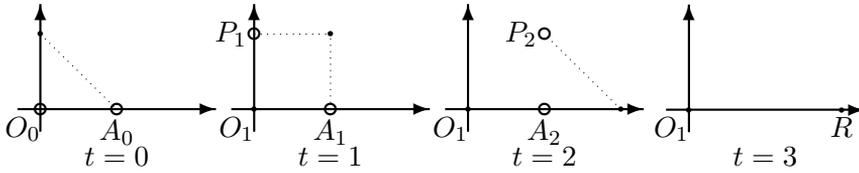


Рисунок 1. Первый алгоритм удвоения чисел

с координатами  $(0, a)$  возбуждается и переходит в состояние “проекция 1”. Во второй момент “проекция 1” и “аргумент” подают в эфир сигнал “такт 2”. Ячейка, которая услышит этот сигнал в локаторы запад и юг (а это ячейка с координатами  $(a, a)$ ), возбуждается и переходит в состояние “проекция 2”. В третий момент ячейки “проекция 2” и “аргумент” подают в эфир сигнал “такт 3”. Ячейка, которая услышит этот сигнал в локаторы запад и северо-запад (а это ячейка с координатами  $(2a, 0)$ ) возбуждается и переходит в состояние “результат”. Т.е. задача решается за 3 такта.

Схематически этот алгоритм отражен на рисунке 1. Здесь ячейки, подающие сигнал в эфир, изображены полыми кружками, “начало координат” обозначается символами  $O$  с индексами, “аргумент” — символами  $A$  с индексами, “проекции” — символами  $P$  с индексами, “результат” — символом  $R$ .

Когда на каждой итерации надо удваивать число, то задачу удвоения числа удобнее сформулировать следующим образом. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами  $(0, 0)$  находится в состоянии “начало координат”, а две ячейки с координатами  $(a, 0)$  и  $(0, a)$  находятся в состоянии “аргумент”. Надо, чтобы в финальной конфигурации ячейки с координатами  $(2a, 0)$  и  $(0, 2a)$  перешли в состояние “результат”, а все остальные ячейки, кроме  $(0, 0)$ , оказались в состоянии покоя.

Решить эту задачу можно следующим образом. В начальный момент ячейки “аргумент” подают в эфир сигнал “такт 1”. Ячейка, которая услышит этот сигнал в локаторы запад и юг (а это ячейка с координатами  $(a, a)$ ), возбуждается и переходит в состояние “проекция”. Во второй момент ячейки “проекция” и “аргумент” подают в эфир сигнал “такт 2”. Ячейка, которая услышит этот сигнал в локаторы запад и северо-запад (а это ячейка с координатами  $(2, 0)$ ) и ячейка, которая услышит этот сигнал в локаторы юг и юго-восток (а это ячейка с координатами  $(0, 2a)$ ) возбуждаются и переходят в состояние “результат”. Тем самым в такой постановке задача решается за 2 такта.

Схематически этот алгоритм отражен на рисунке 2.

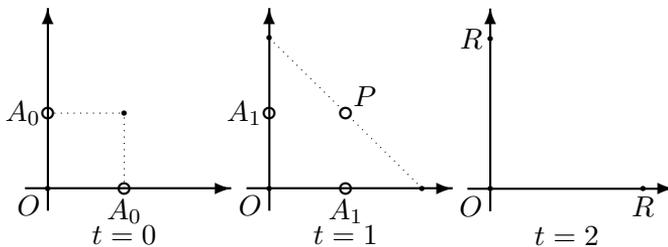


Рисунок 2. Второй алгоритм удвоения чисел

## 2.2. Сложение чисел

Задача сложения состоит в следующем. Дано два целых числа  $a$  и  $b$ , причем  $a \geq 0$ , а  $b > 0$ . В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами  $(0, 0)$  находится в состоянии “начало координат”, а две ячейки с координатами  $(a, 0)$  и  $(0, b)$  находятся в состоянии “слагаемое 1” и “слагаемое 2”. В случае, когда  $a = 0$ , ячейка  $(0, 0)$  будет одновременно находиться в состояниях “начало координат” и “слагаемое 1”. Чтобы это сделать можно ввести еще одно состояние, или можно считать, что “начало координат” отмечается в отдельной компоненте состояния. Надо, чтобы в финальной в состояниях конфигурации ячейка с координатами  $(a + b, 0)$  перешла в состояние “результат”, ячейка  $(0, 0)$  осталась в состоянии “начало координат”, а все остальные ячейки оказались в состоянии покоя.

Решить эту задачу можно следующим образом. В начальный момент ячейки “слагаемое 1” и “слагаемое 2” подают в эфир сигнал “такт 1”. Ячейка, которая услышит этот сигнал в локаторы запад и юг (а это ячейка с координатами  $(a, b)$ ), возбуждается и переходит в состояние “проекция”. Также в состояние “проекция” переходит ячейка “слагаемое 2”, если услышит сигнал “такт 1” в локатор “юг” (это нужно для случая, когда  $a = 0$ ). Ячейка “слагаемое 2”, которая не слышит сигнал “такт 1” в локатор “юг”, переходит в состояние покоя. Ячейка “слагаемое 1”, если слышит сигнал “такт 1” в локаторы “северо-запад” или “север”, переходит в состояние “слагаемое 3”, а иначе остается в прежнем состоянии. Во второй момент ячейки “проекция” и “слагаемое 3” подают в эфир сигнал “такт 2”. Ячейка, которая услышит этот сигнал в локаторы запад и северо-запад (а это ячейка с координатами  $(a + b, 0)$ ) возбуждается и переходит в состояние “результат”. При этом “слагаемое 3” переходит в состояние покоя. Ячейка “начало координат” оба такта не меняет своего состояния.

Тем самым задача сложения чисел решается за 2 такта.

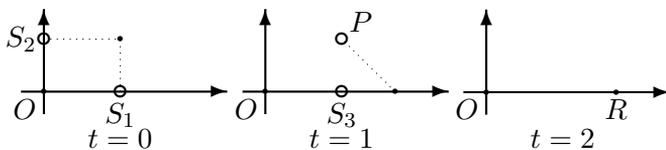


Рисунок 3. Сложение чисел

Схематически этот алгоритм для случая, когда  $a > 0$ , отражен на рисунке 3. На этом рисунке состояние “начало координат” обозначено символом  $O$ , “слагаемое 1” — символом  $S_1$ , “слагаемое 2” —  $S_2$ , “слагаемое 3” —  $S_3$ , “проекция” —  $P$ .

Отметим, что если в начальной конфигурации есть ячейка в состоянии “слагаемое 1”, но нет ячейки в состоянии “слагаемое 2”, то ячейка “слагаемое 1” не будет менять своего состояния.

### 2.3. Перевод числа из унарного представления в двоичное

Пусть  $a$  — натуральное число,  $(a_n, a_{n-1}, \dots, a_1)$  — двоичное представление числа  $a$ . Задача перевода числа из унарного представления в двоичное формулируется следующим образом. В начальной конфигурации в активном состоянии находится  $n + 1$  ячейка: ячейка  $(0, 0)$  — в состоянии “начало координат” и ячейки  $(1, 0), (2, 0), \dots, (n, 0)$  — в состоянии “один”. Задача состоит в том, чтобы в такты с первого по  $n$ -ый выдавать в эфир сигнал “ноль”, если  $a_i = 0$ , и сигнал “единица”, если  $a_i = 1$ ,  $i = 1, 2, \dots, n$ . При этом в финальной конфигурации активной остается только “начало координат”. Здесь первым тактом считается такт следующий после начального.

Точное решение этой задачи приведено в работах [10, 11]. Здесь, как и ранее, мы опишем алгоритм решения на идейном уровне.

Алфавит вещания будет иметь вид  $G = \{0, 1\} \times \{0, 1\} \times \{0, 1, 2\}$ .

Полугрупповой операцией по первой компоненте будет сложение по модулю 2, а по второй и третьей — максимум. Первая компонента будет использоваться для вычисления чисел  $a_i$ ,  $i = 1, 2, \dots, n$ . Вторая компонента — для выявления момента окончания вычислений, а третья компонента — для передачи ответа. Состояние покоя будем обозначать как состояние “ноль”. Алгоритм решения задачи будет следующий.

- 1) В каждый такт все ячейки, которые находятся в состоянии “один” передают в эфир сигнал  $(1, 1, 0)$ .

- 2) Ячейка в состоянии “один”, которая в локатор “восток” получает сигнал  $(0, *, *)$ , переходит в состояние “ноль”. Здесь  $*$  означает любой символ. Во всех остальных случаях ячейка не меняет состояние.
- 3) Если ячейка “начало координат” в локатор “восток” получает сигнал  $(0, 1, 0)$ , то она посылает в эфир сигнал  $(0, 0, 0)$ , что соответствует сигналу “ноль”.
- 4) Если ячейка “начало координат” в локатор “восток” получает сигнал  $(1, 1, 0)$ , то она посылает в эфир сигнал  $(0, 0, 1)$ , что соответствует сигналу “единица”.
- 5) Если ячейка “начало координат” в локатор “восток” получает сигнал  $(0, 0, 0)$ , то она посылает в эфир сигнал  $(0, 0, 2)$ , что соответствует окончанию передачи двоичного представления числа.

Поскольку согласно пункту 1 каждая ячейка в состоянии “один” передает в эфир по первой компоненте значение 1, то “начало координат” получит в локатор “восток” по первой компоненте сумму по модулю 2 количества ячеек в состоянии “один”, а это в первый момент равно  $a_1$ . Второй пункт гарантирует, что каждый такт число ячеек в состоянии “один” будет сокращаться вдвое, поэтому во второй такт “начало координат” получит в локатор “восток” по первой компоненте значение  $a_2$  и т.д. Если “начало координат” получит в локатор “восток” по второй компоненте значение 0, то это означает, что ячеек в состоянии “один” больше не осталось, и можно завершать работу.

Отметим, что суммарное время работы алгоритма равно  $n + 2$ .

В таблице 1 приведено поведение описанного выше клеточного автомата с локаторами для случая, когда число  $a = 5$ . Здесь символом  $Q$  обозначается строка состояний ячеек, причем  $O$  соответствует состоянию “начало координат”,  $0$  — состоянию “ноль”,  $1$  — состоянию “один”; символом  $S$  обозначается строка посылаемых в эфир сигналов; символом  $E$  — строка значений локатора “восток”.

Координаты ячеек		(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)
$t = 0$	$Q$	$O$	1	1	1	1	1
	$S$	(0,0,2)	(1,1,0)	(1,1,0)	(1,1,0)	(1,1,0)	(1,1,0)
	$E$	(1,1,0)	(0,1,0)	(1,1,0)	(0,1,0)	(1,1,0)	(0,0,0)
$t = 1$	$Q$	$O$	0	1	0	1	0
	$S$	(0,0,1)	(0,0,0)	(1,1,0)	(0,0,0)	(1,1,0)	(0,0,0)
	$E$	(0,1,0)	(0,1,0)	(1,1,0)	(1,1,0)	(0,0,0)	(0,0,0)
$t = 2$	$Q$	$O$	0	1	0	0	0
	$S$	(0,0,0)	(0,0,0)	(1,1,0)	(0,0,0)	(0,0,0)	(0,0,0)
	$E$	(1,1,0)	(1,1,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
$t = 3$	$Q$	$O$	0	0	0	0	0
	$S$	(0,0,1)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
	$E$	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
$t = 4$	$Q$	$O$	0	0	0	0	0
	$S$	(0,0,2)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
	$E$	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)

Таблица 1. Перевод числа из унарного представления в двоичное

В такты 1, 2, 3 в третьей компоненте сигнала вещания мы можем наблюдать двоичное представление числа 5 — (1,0,1).

Легко видеть, что приведенный клеточный автомат будет работать и в случае, когда ячейки в состоянии “один” будут стоять не подряд, а в любых положительных позициях оси абсцисс, и тогда автомат выдаст в эфир двоичное представление количества ячеек в состоянии “один”, находящихся правее “начала координат”.

Можем также заметить, что легко модифицировать этот автомат, чтобы он выдавал компоненты двоичного представления не каждый такт, а, например, через такт.

И наконец заметим, что аналогичным образом мы можем подсчитать число ячеек в состоянии “один”, находящихся на любом из лучей из множества  $L$ .

## 2.4. Перевод числа из двоичного представления в унарное

Задача перевода числа из двоичного представления в унарное обратна к предыдущей задаче.

Сформулирована она может быть следующим образом. Пусть  $a$  — натуральное число,  $(a_n, a_{n-1}, \dots, a_1)$  — двоичное представление числа  $a$ . В начальной конфигурации в активном состоянии находится только ячейка (0,0) в состоянии “начало координат 0”. В такты с первого по  $n$ -ый ячейка (0,0) выдает в эфир сигнал “ноль”, если  $a_i = 0$ , и сигнал “единица”, если  $a_i = 1$ ,  $i = 1, 2, \dots, n$ . В финальной конфигурации активными

должны остаться только ячейки  $(0, 0)$  (“начало координат 2”) и  $(a, 0)$  (“результат”).

Точное решение этой задачи можно найти в работах [9, 10]. Приведем идею этого решения.

Алфавит вещания будет иметь вид  $G = \{0, 1\} \times \{0, 1, 2, 3, 4\}$ . Полугрупповой операцией по первой компоненте будет сложение по модулю 2, а по второй — максимум. Первая компонента будет использоваться для вычисления числа  $a$ , вторая компонента — для передачи команд. Состояние покоя будем обозначать как состояние “ноль”. Алгоритм решения задачи будет следующий.

- 1) В начальный (нулевой) такт ячейка  $(0, 0)$  (“начало координат 0”) подает в эфир сигнал  $(1, 2)$ , который можно интерпретировать как “начинаем”, и переходит в состояние “начало координат 1”. По команде “начинаем” все ячейки положительной полуоси оси абсцисс (т.е. те, кто услышат этот сигнал в локатор “запад”) переходят в состояние “один”.
- 2) В следующие  $n$  тактов ячейка  $(0, 0)$  подает в эфир по второй компоненте сигнал  $a_i$ ,  $i = 1, 2, \dots, n$  ( $a_i = 0$  интерпретируется как сигнал “ноль”, а  $a_i = 1$  — как “единица”). При этом на следующий такт после того, как в первый раз  $a_i$  окажется равным 1, ячейка  $(0, 0)$  перейдет в состояние “начало координат 2”, а до этого будет оставаться в состоянии “начало координат 1”. Поскольку число  $a > 0$ , к финальному состоянию ячейка  $(0, 0)$  обязательно окажется в состоянии “начало координат 2”. При этом “начало координат 1” по первой компоненте передает в эфир сигнал 1, а “начало координат 2” — сигнал 0.
- 3) Ячейки в состоянии “один”, если слышат в локатор “запад” по второй компоненте 0 или 1, то передают в эфир сигнал  $(1, 0)$ , а ячейки в состоянии “ноль” — сигнал  $(0, 0)$ .
- 4) Ячейка в состоянии “один” переходит в состояние “ноль”, если она получает в локатор “запад” сигнал, значение второй компоненты которого равно либо 0, либо 1, и оно не совпадает со значением первой компоненты сигнала.
- 5) В  $(n + 1)$ -ый такт ячейка  $(0, 0)$  передает в эфир сигнал  $(0, 3)$ , что означает окончание двоичной записи. В результате все ячейки в состоянии “один” переходят в состояние “два”.
- 6) В  $(n + 2)$ -ый такт ячейка  $(0, 0)$  передает в эфир сигнал  $(0, 3)$ , а все ячейки в состоянии “два” подают в эфир сигнал  $(0, 4)$ . При этом

все ячейки в состоянии “два”, которые в локатор “запад” услышат сигнал (0, 4) перейдут на следующий такт в состояние “ноль”, а самая левая ячейка в состоянии “два” услышит в локатор “запад” сигнал (0, 3) и перейдет в состояние “результат”.

В таблице 2 приведено поведение описанного выше клеточного автомата с локаторами для случая, когда число  $a = 5$ . Здесь символом  $Q$  обозначается строка состояний ячеек, причем  $O_0, O_1, O_2$  соответствует состоянию “начало координат 0”, “начало координат 1” и “начало координат 2”, 0 — состоянию “ноль”, 1 — состоянию “один”, 2 — состоянию “два”,  $R$  — состоянию “результат”; символом  $S$  обозначается строка посылаемых в эфир сигналов; символом  $W$  — строка значений локатора “запад”.

$t$		(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)	(6,0)	(7,0)	(8,0)	(9,0)	(10,0)	(11,0)	(12,0)	(13,0)
0	$Q$	$O_0$	0	0	0	0	0	0	0	0	0	0	0	0	0
	$S$	(0,2)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
	$W$	(0,0)	(0,2)	(0,2)	(0,2)	(0,2)	(0,2)	(0,2)	(0,2)	(0,2)	(0,2)	(0,2)	(0,2)	(0,2)	(0,2)
1	$Q$	$O_1$	1	1	1	1	1	1	1	1	1	1	1	1	1
	$S$	(1,1)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)	(1,0)
	$W$	(0,0)	(1,1)	(0,1)	(1,1)	(0,1)	(1,1)	(0,1)	(1,1)	(0,1)	(1,1)	(0,1)	(1,1)	(0,1)	(1,1)
2	$Q$	$O_2$	1	0	0	1	0	1	0	1	0	1	0	1	0
	$S$	(0,0)	(1,0)	(0,0)	(1,0)	(0,0)	(1,0)	(0,0)	(1,0)	(0,0)	(1,0)	(0,0)	(1,0)	(0,0)	(1,0)
	$W$	(0,0)	(0,0)	(1,0)	(1,0)	(0,0)	(0,0)	(1,0)	(1,0)	(0,0)	(0,0)	(1,0)	(1,0)	(0,0)	(0,0)
3	$Q$	$O_2$	1	0	0	0	1	0	0	0	1	0	0	0	1
	$S$	(0,1)	(1,0)	(0,0)	(0,0)	(0,0)	(1,0)	(0,0)	(0,0)	(0,0)	(1,0)	(0,0)	(0,0)	(0,0)	(1,0)
	$W$	(0,0)	(0,1)	(1,1)	(1,1)	(1,1)	(1,1)	(0,1)	(0,1)	(0,1)	(0,1)	(1,1)	(1,1)	(1,1)	(1,1)
4	$Q$	$O_2$	0	0	0	0	1	0	0	0	0	0	0	0	1
	$S$	(0,3)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)
	$W$	(0,0)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(1,3)	(1,3)	(1,3)	(1,3)	(1,3)	(1,3)	(1,3)	(1,3)
5	$Q$	$O_2$	0	0	0	0	2	0	0	0	0	0	0	0	2
	$S$	(0,3)	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,4)
	$W$	(0,0)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(0,4)	(0,4)	(0,4)	(0,4)	(0,4)	(0,4)	(0,4)	(0,4)
6	$Q$	$O_2$	0	0	0	0	$R$	0	0	0	0	0	0	0	0
	$S$	(0,3)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
	$W$	(0,0)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)	(0,3)

Таблица 2. Перевод числа из двоичного представления в унарное

Будем условно считать, что состояние “начало координат 1” соответствует состоянию “один” для ячейки (0, 0), а состояние “начало координат 2” соответствует состоянию “ноль” для ячейки (0, 0). Тогда докажем по индукции, что если ячейка (0, 0) с первого по  $n$ -й такт по второй компоненте будет посылать в эфир последовательность  $a_1, a_2, \dots, a_n$ , то к  $(n + 1)$ -му такту в состоянии “один” будут ячейки с координатами  $(\sum_{i=1}^n 2^i a_i + 2^n k, 0)$ ,  $k = 0, 1, 2, \dots$

**Базис индукции.**  $n = 1$ . На первом такте все ячейки неотрицательной полуоси оси абсцисс, включая ячейку (0, 0), посылают в эфир сигнал 1 по первой компоненте. Поэтому все ячейки с координатами  $(2k, 0)$ ,  $k = 0, 1, 2, \dots$  (четные ячейки) получают в локатор “запад” по первой компоненте значение 0, а все ячейки с координатами  $(2k + 1, 0)$ ,

$k = 0, 1, 2, \dots$  (нечетные ячейки) — значение 1. Следовательно, если  $a_1 = 0$ , то ко второму такту в состоянии “один” останутся четные ячейки, а если  $a_1 = 1$ , то — нечетные. Базис индукции доказан.

**Индуктивный переход.** Пусть ячейка  $(0, 0)$  с первого по  $(n - 1)$ -й такт по второй компоненте посылала в эфир последовательность  $a_1, a_2, \dots, a_{n-1}$ , к  $n$ -му такту в состоянии “один” остались ячейки с координатами  $(\sum_{i=1}^{n-1} 2^i a_i + 2^{n-1} k, 0)$ ,  $k = 0, 1, 2, \dots$ . Обозначим  $b = \sum_{i=1}^{n-1} 2^i a_i$ . Тогда все ячейки с координатами  $(k, 0, k = 0, 1, \dots, b)$  получают на локатор “запад” по первой компоненте значение 0. Все ячейки с координатами  $(k, 0, k = b+1, b+2, \dots, b+2^{n-1})$  получают на локатор “запад” по первой компоненте значение 1. Опять все ячейки с координатами  $(k, 0, k = b + 2^{n-1} + 1, b + 2^{n-1} + 2, \dots, b + 2^n)$  получают на локатор “запад” по первой компоненте значение 0 и т.д.

Пусть на  $n$ -м такте ячейка  $(0, 0)$  посылает в эфир по второй компоненте значение  $a_n$ . Тогда если  $a_n = 0$ , то на  $(n + 1)$ -м такте в состоянии “один” останутся ячейки с координатами  $(b + 2^n k, 0)$ ,  $k = 0, 1, 2, \dots$ . Если  $a_n = 1$ , то на  $(n + 1)$ -м такте в состоянии “один” останутся ячейки с координатами  $(b + 2^{n-1} + 2^n k, 0)$ ,  $k = 0, 1, 2, \dots$

Индуктивный переход доказан.

Таким образом, если  $a = \sum_{i=1}^n 2^i a_i$ , то к  $(n + 1)$ -му такту самая левая ячейка в состоянии “один” будет иметь координаты  $(a, 0)$ , а значит на  $(n + 3)$ -м такте в состоянии “результат” перейдет ячейка  $(a, 0)$ , что мы и хотели получить.

Отметим, что суммарное время работы алгоритма равно  $n + 3$ .

Отметим, что аналогичным образом можно отложить значение  $a$  на ось ординат, т.е. чтобы в финальной конфигурации в состоянии “результат” оказалась ячейка  $(0, a)$ .

## 2.5. Задача подсчета числа единиц

Пусть в начальной конфигурации ячейка  $(0, 0)$  находится в состоянии “начало координат” и на положительной полуоси оси абсцисс разбросано некоторое конечное число ячеек в состоянии “один”, а остальные ячейки находятся в состоянии покоя (состоянии “ноль”). Хочется подсчитать сколько ячеек находится в состоянии “один”, и если их число равно  $a$ , то хочется в финальной конфигурации, чтобы ячейка  $(0, a)$  перешла в состояние “результат”.

Пусть двоичное представление числа  $a$  имеет вид  $(a_n, a_{n-1}, \dots, a_1)$ . Мы можем запустить алгоритм из раздела 2.3 и тогда, начиная с первого такта, ячейка  $(0, 0)$  будет посылать в эфир последовательность  $a_1, a_2, \dots, a_n$ . Теперь, параллельно используя алгоритм из раздела 2.4, мы можем отложить число  $a$  на оси ординат.

Отметим, что суммарное время работы алгоритма равно  $n + 3$ , поскольку оба алгоритма работают такое время, работают параллельно и синхронно.

Решение этой задачи мы в дальнейшем используем при алгоритме деления.

### 3. Умножение чисел

Напомним задачу умножения чисел  $a$  и  $b$  для клеточного автомата с локаторами. В начальной конфигурации только 3 ячейки находятся не в состоянии покоя, а именно ячейка с координатами  $(0, 0)$  находится в состоянии, которое можно назвать “начало координат”, ячейка с координатами  $(a, 0)$  находится в состоянии, которое можно назвать “первый сомножитель”, а ячейка с координатами  $(0, b)$  находится в состоянии, которое можно назвать “второй сомножитель”. Клеточный автомат решает задачу умножения чисел, если в финальной конфигурации ячейка с координатами  $(a \cdot b, 0)$  перейдет в состояние “результат умножения”, а все остальные ячейки, кроме  $(0, 0)$ , перейдут в состояние покоя.

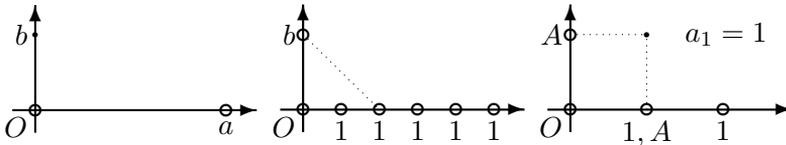
Крупными мазками опишем алгоритма решения задачи умножения чисел  $a$  и  $b$ . Пусть  $(a_n, a_{n-1}, \dots, a_1)$  — двоичное представление числа  $a$ , т.е.  $a = \sum_{i=1}^n 2^{i-1} a_i$ . Тогда  $ab = \sum_{i=1}^n 2^{i-1} b a_i$ . Клеточный автомат, который будет решать задачу умножения чисел  $a$  и  $b$ , будет одновременно выполнять 3 задачи. Чтобы выполнять эти три задачи параллельно в нашем автомате будут работать одновременно 3 разных автомата, каждый над своим множеством компонент состояний и своим множеством компонент сигналов вещания.

С помощью первого автомата будет решаться задача вычисления чисел  $2^{i-1}b$ ,  $i = 1, 2, \dots, n$ . Делать это будем с помощью второго алгоритма удвоения чисел, описанного в разделе 2.1, т.е. сначала сложим два числа  $b$  за 2 такта, затем сложим два числа  $2b$  за 2 такта и т.д., т.е. каждые 2 такта мы сможем получать числа  $2^{i-1}b$ ,  $i = 1, 2, \dots, n$ .

С помощью второго автомата будем находить двоичное представление числа  $a$ , т.е. будем получать числа  $a_1, a_2, \dots, a_n$ . Делать это будем как описано в разделе 2.3. Но будем притормаживать получение чисел  $a_2, a_3, \dots, a_n$  так, чтобы они появлялись одновременно с числами  $2^1b, 2^2b, \dots, 2^{n-1}b$ .

С помощью третьего автомата будем накапливать суммы

$$S_i = \sum_{j=1}^i 2^{j-1} b a_j, i = 1, 2, \dots, n, S_0 = 0.$$



Риснок 4. Умножение чисел,  $t = 0$ ,  $t = 1$ ,  $t = 2$ .

Т.е. если вычисленное число  $a_i$  равно единице, то к числу  $S_{i-1}$  добавим вычисленное число  $2^{i-1}b$ . Это тоже можно сделать за 2 такта, как было описано в разделе 2.2.

Тем самым описанный клеточный автомат с локаторами может приблизительно за  $2n$  тактов вычислить число  $ab$ .

На самом деле алгоритм несколько сложнее и нам понадобится еще четвертый автомат, который будет управляющим и будет координировать действия описанных выше трех автоматов. Этот автомат будет иметь свое множество компонент состояний и свое множество компонент сигналов вещания.

Чтобы запускать удвоение чисел первым автоматом и суммирование чисел третьим автоматом итеративно каждый второй такт, отождествим состояния “аргумент” и “результат” в первом автомате, и состояния “слагаемое 1” и результат в третьем автомате.

Опишем более детально наш алгоритм и получим точную оценку времени работы этого алгоритма.

- 1) В начальный (нулевой) такт ячейка  $(0, 0)$  (“начало координат”) и ячейка  $(a, 0)$  (“первый сомножитель”) посылают в эфир сигнал, который можно назвать “строим унарное представление”. Этот сигнал посылается по компоненте четвертого автомата. Все ячейки, которые услышат этот сигнал в локаторы “восток” и “запад” поймут, что они между “началом координат” и “первым сомножителем” и перейдут в состояние “один” второго автомата, “первый сомножитель” тоже переходит в состояние “один” второго автомата.
- 2) В результате на первом такте ячейки с координатами  $(1, 0), (2, 0), \dots, (a, 0)$  окажутся в состоянии “один”, и мы на них запустим второй автомат для вычисления чисел  $a_1, a_2, \dots, a_n$ . Ячейка  $(0, b)$  (“второй сомножитель”) проецируется на ось абсцисс. Для этого “начало координат” и “второй сомножитель” посылают в эфир по компоненте четвертого автомата сигнал, который можно назвать “проецируем на ось абсцисс”. Ячейка, которая услышит этот сигнал в локаторы “запад” и “северо-запад” (а это будет ячейка  $(b, 0)$ ), перейдет в

состояние “аргумент” первого автомата, “второй сомножитель” тоже перейдет в состояние “аргумент” первого автомата.

- 3) На втором такте в эфире от второго автомата появится число  $a_1$ . Если  $a_1 = 1$ , то ячейка  $(b, 0)$  переводится в состояние “слагаемое 1” третьего автомата, а если  $a_1 = 0$ , то ячейка  $(0, 0)$  переводится в состояние “слагаемое 1” третьего автомата. Также на втором такте ячейки  $(b, 0)$  и  $(0, b)$  окажутся в состоянии “аргумент” первого автомата, поэтому запускается первый автомат для получения чисел  $2^{i-1}b$ ,  $i = 2, 3, \dots, n$ .
- 4) На третьем такте в эфире от второго автомата появится число  $a_2$ . Если  $a_2 = 1$ , то первый автомат, у которого на следующий такт появится ячейка на оси ординат в состоянии “результат” (это будет ячейка  $(0, 2b)$ , переводит добавочно эту ячейку в состояние “слагаемое 2” третьего автомата. Если  $a_2 = 0$ , то у третьего автомата ячейки в состоянии “слагаемое 2” не появится. Также с этого момента второй автомат переключается в режим выдачи ответа через такт.
- 5) На тактах  $2i$ ,  $i = 2, 3, \dots$ , будут появляться результаты удвоения, которые одновременно могут превращаться в “слагаемое 2” третьего автомата (если на предыдущем такте  $a_i = 1$ ), и третий автомат начнет складывать числа. Также на этих тактах может появиться результат сложения третьего автомата, если за 2 такта до этого процесс сложения был запущен. Результат сложения сразу же превратится в “слагаемое 1”, поскольку мы отождествили состояния “слагаемое 1” и “результат” у третьего автомата.
- 6) На тактах  $2i + 1$ ,  $i = 2, 3, \dots, n - 1$ , будут появляться числа  $a_{i+1}$  от второго автомата. Если  $a_{i+1} = 1$ , то первый автомат, у которого на следующий такт появится ячейка на оси ординат в состоянии “результат” (а это будет ячейка  $(0, 2^i b)$ ) переводит добавочно эту ячейку в состояние “слагаемое 2” третьего автомата, что приводит к запуску процесса сложения третьим автоматом. Если  $a_{i+1} = 0$ , у третьего автомата не появляется ячейка в состоянии “слагаемое 2” и процесс сложения чисел не запускается.
- 7) На такте  $2n + 1$  второй автомат поймет, что обработка числа  $a$  закончилась и прокричит в эфир сигнал “окончание работы”. В результате все ячейки, кроме “начала координат” и результата сложения третьего автомата, перейдут в состояние покоя, а результат сложения третьего автомата перейдет в состояние “результат умножения”.

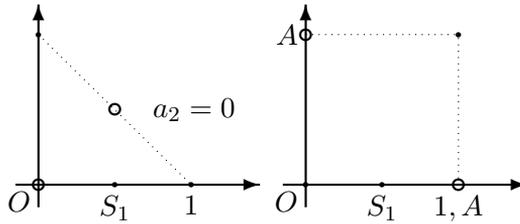


Рисунок 5. Умножение чисел,  $t = 3$ ,  $t = 4$ .

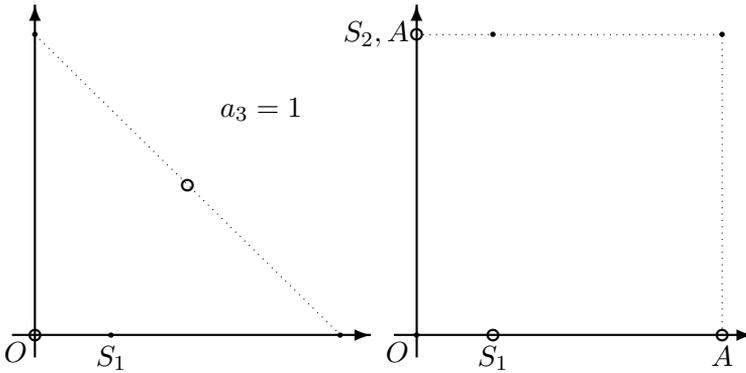


Рисунок 6. Умножение чисел,  $t = 5$ ,  $t = 6$ .

- 8) На такте  $2n + 2$  автомат завершает работу в требуемой финальной конфигурации.

Таким образом описанный клеточный автомат с локаторами решает задачу умножения чисел  $a$  и  $b$  за время  $2\lceil \log_2 a \rceil + 2$ . Теорема 1 доказана.

На рисунках 4 – 7 изображен процесс умножения чисел  $a = 5$  и  $b = 2$ . На этих рисунках полыми кружками изображены ячейки, подающие сигнал в эфир. Символом  $O$  обозначено “начало координат”. Символом  $A$  обозначено состояние “аргумент” первого автомата. Символом  $1$  обозначено состояние “один” второго автомата. Символами  $S_1$  и  $S_2$  обозначены состояния “слагаемое 1” и “слагаемое 2” третьего автомата. Символом  $R$  обозначено состояние “результат умножения”. Когда ячейка одновременно находится в нескольких описанных выше состояниях разных автоматов, то они перечислены через запятую.

#### 4. Деление чисел

Идея алгоритма решения состоит в последовательном проецировании отрезков делителя на абсциссу, на которой также отложено делимое.

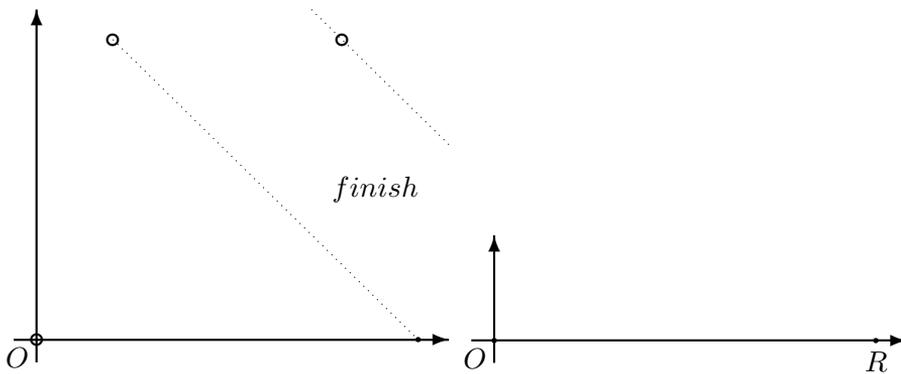


Рисунок 7. Умножение чисел,  $t = 7$ ,  $t = 8$ .

Как только сумма отрезков делителя становится больше или равна делимому, значит частное и остаток от деления найдены. Далее следует откладывание частного и остатка от деления на абсциссе и ординате соответственно, в унарном формате. Таким образом операцию деления можно разбить на два этапа:

- откладывание отрезков делителя на оси абсцисс;
- откладывание частного и остатка от деления на осях абсцисс и ординат.

#### 4.1. Первый этап

Рассмотрим следующий двумерный клеточный автомат с локаторами  $\sigma = (\mathbb{Z}^2, Q, \emptyset, E_8, \max, L, \varphi, \psi)$ .  $Q$  — это множество состояний ячеек автомата,  $Q = \{*, O, O_1, O_2, O_3, O_4, O_5, A, A_1, A_2, A_3, A_4, A_5, A_6, B, B_1, B_2, B_3, B_4, B_5, X, Y_2, Y_3, R\}$ , где  $*$  — состояние покоя. Шаблон соседства пустой. Шаблон локаторов  $L$  задается соотношением (1).  $E_8 = \{0, 1, \dots, 7\}$  — алфавит сигналов вещания, полугрупповая операция — максимум. Ячейки передают сигналы о своем местоположении и состоянии.

Пусть  $b$  — делимое,  $a$  — делитель, и  $b/a \leq 2^n$ . Тогда клеточный автомат с локаторами  $\sigma$  будет решать задачу первого этапа, т.е. откладывания отрезков делителя на оси абсцисс за время  $T_1 = 2n + 5$ .

В качестве примера рассмотрим случай, когда делимое делится на делитель не полностью. Делитель  $a = 3$ , делимое  $b = 17$ . Начало координат обозначено буквой  $O$ . На рисунках ячейки, которые посылают сигналы в эфир, обозначены белыми кругами, а рядом указано значение сигнала вещания, посылаемого в эфир. Ячейки без кругов посылают в

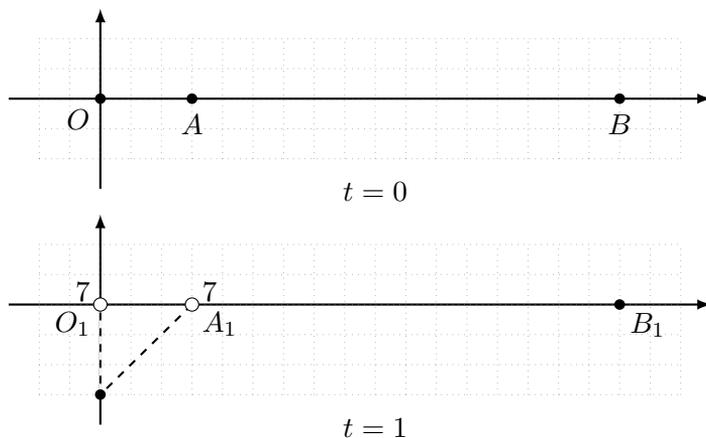


Рисунок 8. Деление чисел, такты 0, 1

эфир нейтральный сигнал 0. Состояние покоя на рисунках изображено в виде пустой клетки.

Первые два такта клеточный автомат готовится откладывать отрезки делителя на абсциссу.

На нулевом такте ( $t = 0$ ) ячейка  $(0, 0)$  в состоянии  $O$  (начало координат) и ячейка  $(a, 0)$  в состоянии  $A$  (делитель) переходят соответственно в состояния  $O_1$  и  $A_1$ , принимают решение о посылке в эфир сигнала 7. Ячейка  $(b, 0)$  в состоянии  $B$  (делимое) переходит в состояние  $B_1$ .

В первом такте ( $t = 1$ ) ячейки в состояниях  $O_1$  и  $A_1$  посылают в эфир сигналы 7, переходят соответственно в состояния  $O_2$  и  $A_2$ . Ячейка в состоянии покоя, которая получает сигналы 7 на локаторы “север” и “северо-восток”, переходит в состояние  $Y_2$ . Вспомогательная ячейка в состоянии  $Y$  необходима для создания временных ячеек в состоянии  $X$ , используемых для проецирования отрезков делителя на ось абсцисс. Ячейка в состоянии  $B_1$  переходит в состояние  $B_2$ .

Далее следует цикл из двух тактов, который работает пока не закончится процедура откладывания отрезков делителя на ось абсцисс. Во время работы цикла ячейки в состоянии  $A_3$  не переходят в другое состояние.

На первом такте цикла (в нашем примере - такты 2,4,6,8) ячейка в состоянии  $O_2$  посылает в эфир сигнал 2, и если она получает сигнал 5 на локатор “восток”, то переходит в состояние  $O_4$ , иначе – в состояние  $O_3$ . Ячейка в состоянии  $A_2$  посылает в эфир сигнал 2, и если она получает сигнал 1 на локатор “восток”, то переходит в состояние  $A_5$ , иначе – в состояние  $A_3$ . Ячейка в состоянии  $B_2$ , которая получает сигнал 1 на

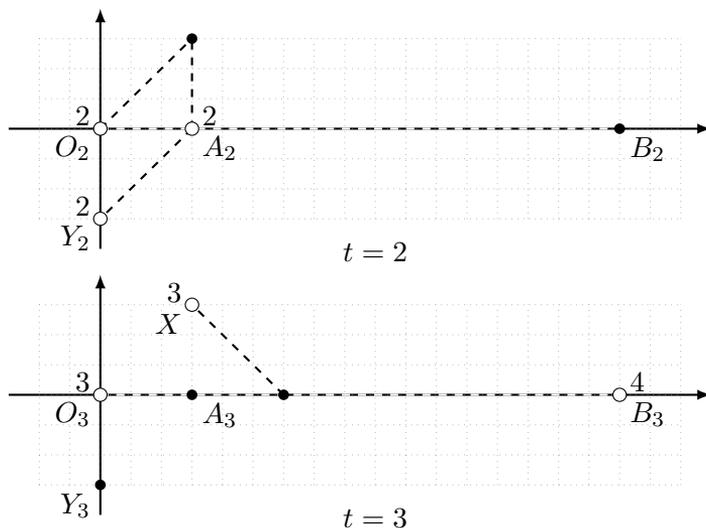


Рисунок 9. Деление чисел, такты 2, 3

локатор “восток”, переходит в состояние  $B_5$ , иначе – в состояние  $B_3$ . Ячейка в состоянии  $Y_2$  посылает в эфир сигнал 2 и переходит в состояние  $Y_3$ . Ячейка в состоянии покоя, которая получает сигнал 2 в локаторы “юго-запад” и “юг”, переходит в состояние  $X$ .

Если в первом такте присутствует ячейка в состоянии  $A_4$ , то она посылает в эфир сигнал 1 и автоматически переходит в состояние покоя. Если присутствует ячейка в состоянии  $B_4$ , то она посылает в эфир сигнал 5 и переходит в состояние  $A_3$ .

На втором такте цикла (в нашем примере - такты 3,5,7,9) ячейка в состоянии  $O_3$  посылает в эфир сигнал 3. Ячейка переходит в состояние  $O_4$ , если получает сигнал 5 на локатор “восток”, иначе возвращается в состояние  $O_2$ . Ячейки в состоянии  $X$  посылают в эфир сигнал 3 и переходят в состояние покоя. Ячейка в состоянии  $B_3$  посылает в эфир сигнал 4. При этом если ячейка получает сигнал 3 на локаторы “северо-запад” и “запад”, то переходит в состояние  $B_4$ , иначе возвращается в состояние  $B_2$ . Ячейка в состоянии  $Y_3$  переходит снова в состояние  $Y_2$ . Ячейки в состоянии покоя, которые получают сигналы 3 на локаторы “северо-запад” и “запад”, переходят в состояние  $A_2$ . Ячейки в состоянии покоя, которые получают сигнал 3 на локатор “северо-запад” и сигнал 4 на локатор “запад”, переходят в состояние  $A_4$ .

Если на втором такте присутствует ячейка в состоянии  $O_4$ , то она посылает в эфир сигнал 6 и переходит в состояние  $O_5$ . Ячейка в состоянии  $A_5$  посылает в эфир сигнал 5 и переходит в состояние  $A_3$ . Ячейка в состоянии  $B_5$  также посылает в эфир сигнал 5 и переходит в состояние

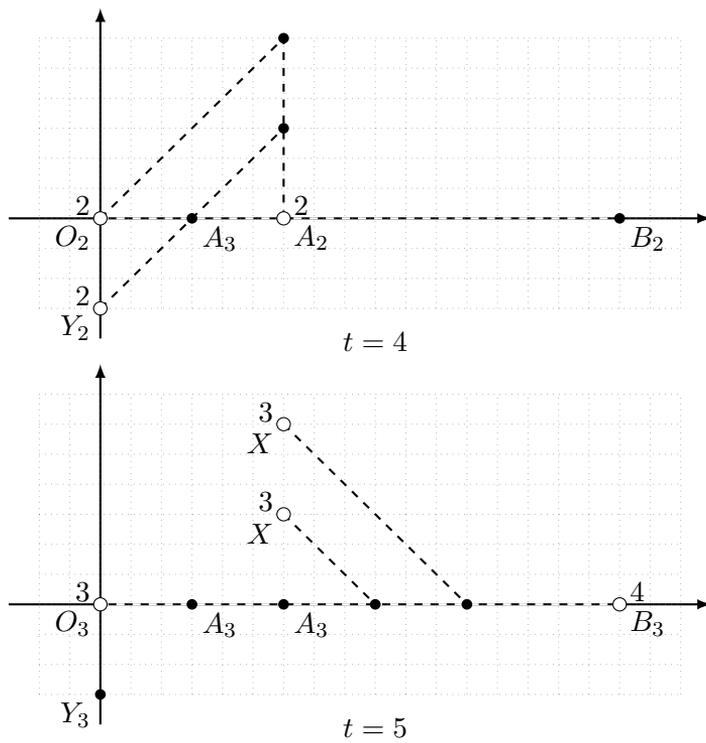


Рисунок 10. Деление чисел, такты 4, 5

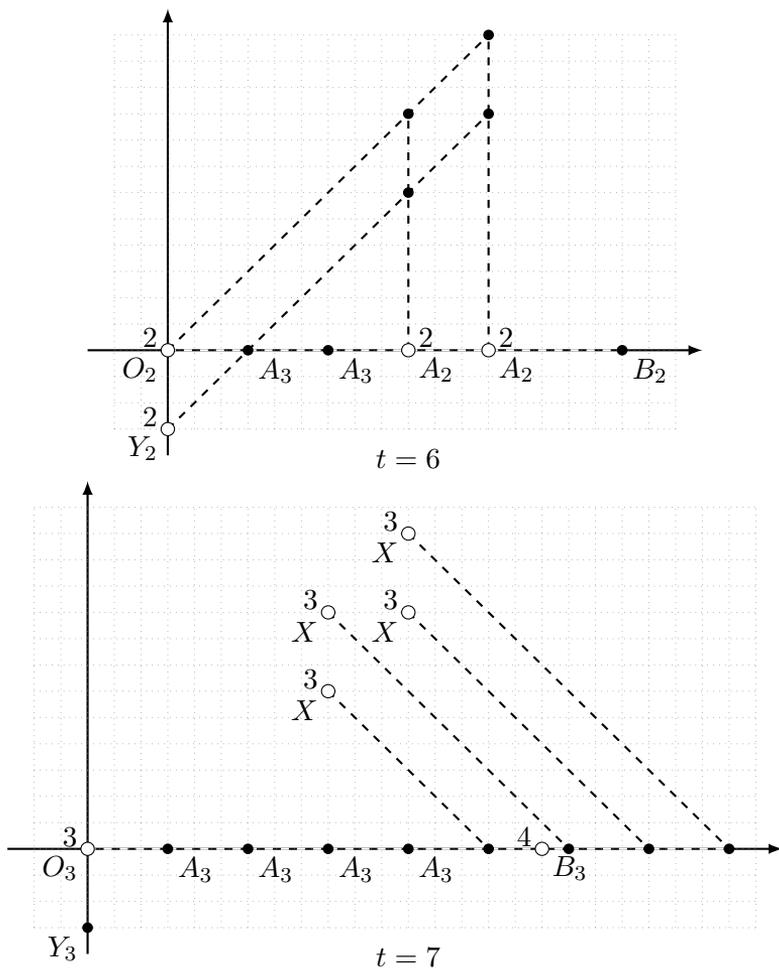


Рисунок 11. Деление чисел, такты 6, 7

покоя. Ячейка в состоянии покоя, которая получает сигнал 5 на локаторы “юг” и “юго-восток”, переходит в состояние  $R$ .

Появление ячейки в состоянии  $B_4$  или  $B_5$  говорит о завершении откладывания отрезков делителя на ось абсцисс. Далее следуют два такта, в которых определяется остаток от деления и удаляется вспомогательная ячейка в состоянии  $Y_2$  или  $Y_3$ .

Ячейка в состоянии  $O_4$  посылает в эфир сигнал 6 и переходит в состояние  $O_5$ . Ячейки в состоянии  $A_3$ , которые получают сигнал 6 на локатор “запад”, переходят в состояние  $A_6$ . Ячейка в состоянии  $Y_2$  или  $Y_3$ , которая получает сигнал 6 на локатор “север”, переходит в состояние покоя.

В нашем примере первый этап завершается на такте 11. В результате определено количество ячеек в состоянии  $A_6$  (частное), и расстояние между самой правой ячейкой в состоянии  $A_6$  и ячейкой в состоянии  $R$  (остаток).

Можно заметить, что за каждую двухтактную итерацию количество ячеек в состоянии  $A$  с индексами удваивается. Это происходит из-за того, что ячейки в состоянии  $X$  получаются на пересечении вертикальных прямых, проходящих через ячейки в состоянии  $A_2$  и двух наклонных прямых, проходящих через ячейки в состоянии  $O_2$  и  $Y_2$ . Поскольку на первой итерации появляется 1 точка в состоянии  $A$  с индексами, а последней итерации таких точек будет  $2^n$ , то всего итераций будет  $n + 1$ . Добавляя к этому 2 такта в начале работы алгоритма и 1 такт в конце (для формирования остатка), получим, что время работы первого этапа будет равно  $T_1 = 2(n + 1) + 3 = 2n + 5$ .

## 4.2. Второй этап

Второй этап — это представление результатов первого этапа в нужном формате.

После первого этапа частное представлено в виде ячеек в состоянии  $A_6$ . Если к описанному выше автомату добавить автомат для подсчета числа единиц, описанный в разделе 2.5, и если отождествить состояние  $A_6$  с состоянием “один” автомата для подсчета числа единиц, то в момент появления ячеек в состоянии  $A_6$  автоматически запустится автомат для подсчета числа единиц. В результате работы этого автомата частное будет отображено на ось абсцисс в унарном формате за время  $T_2 = n + 3$ .

Для отображения остатка на ось ординат достаточно одного такта. Для этого ячейка  $R$  и начало координат, должны послать в эфир некий сигнал, а ячейка, которая услышит этот сигнал в локаторы “восток” и “юг” и будет ячейкой, представляющей остаток. Причем это можно сделать в любое время на фоне вычисления частного.

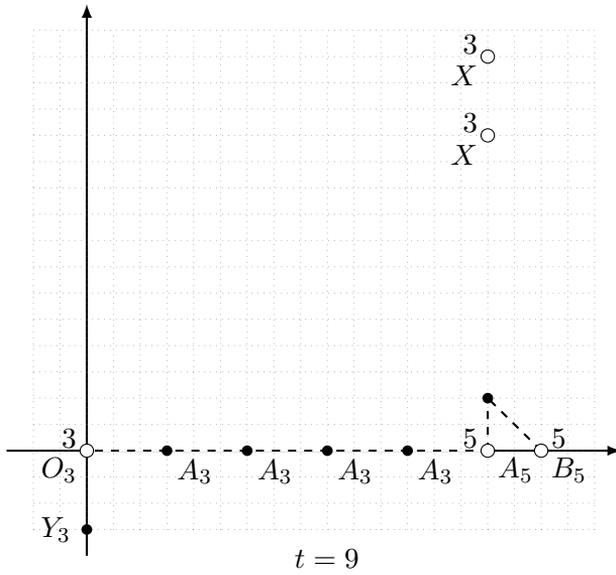
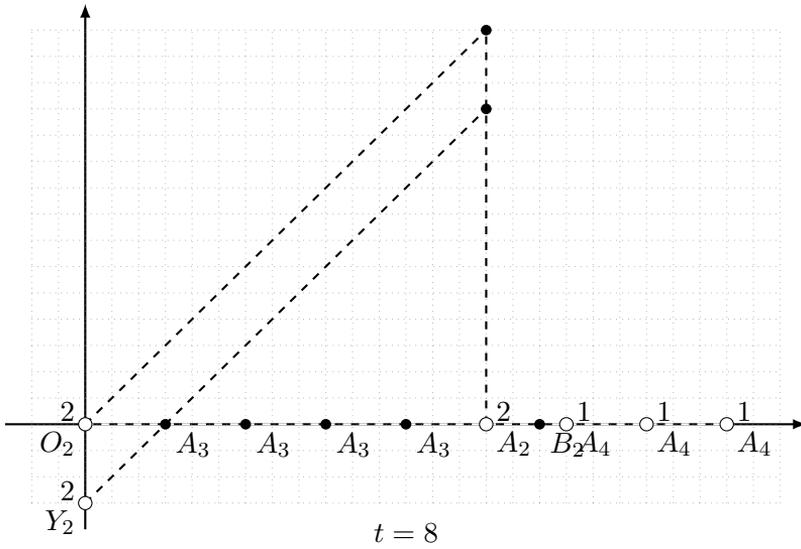


Рисунок 12. Деление чисел, такты 8, 9

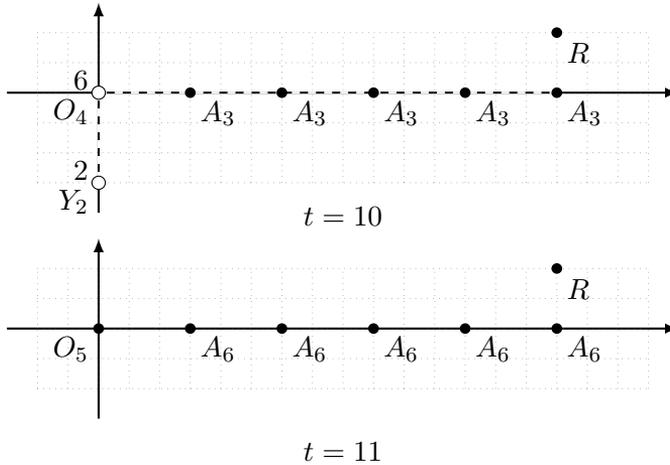


Рисунок 13. Деление чисел, такты 10, 11

Таким образом задача деления чисел  $b$  и  $a$ , где  $b/a \leq 2^n$ , решается нашим двумерным клеточным автоматом с локаторами за время  $T_1 + T_2 = 3n + 8$ . Теорема 2 доказана.

## Список литературы

- [1] Карацуба А., Офман Ю., “Умножение многозначных чисел на автоматах”, *Доклады АН СССР*, **145**:2 (1962), 293–294.
- [2] Schönhage A., Strassen V., “Schnelle Multiplikation großer Zahlen”, *Computing*, 1971, № 7, 281–292.
- [3] Fürer M., “Faster integer multiplication”, *STOC 2007 Proceedings*, 2007, 57–66.
- [4] David Harvey, Joris van der Hoeven, “Integer multiplication in time  $O(n \log n)$ ”, *Annals of Mathematics*, **193**:2 (2021), 563–617.
- [5] Christoph Burnikel C., Ziegler J., “Fast Recursive Division”, *Max-Planck-Institut für Informatik*, 1998.
- [6] Гасанов Э. Э., “Линейный по порядку алгоритм умножения чисел с помощью двумерного клеточного автомата с локаторами”, *Международная научная конференция "Математика в созвездии наук" к юбилею академика В.А. Садовниченко, Москва, Россия, 1-2 апреля 2024*, стр. 316-318.

- [7] Гасанов Э. Э., “Клеточные автоматы с локаторами”, *Интеллектуальные системы. Теория и приложения*, **24**:2 (2020), 120–133.
- [8] Калачев Г. В., “Замечания к определению клеточного автомата с локаторами”, *Интеллектуальные системы. Теория и приложения*, **24**:4 (2020), 47–56.
- [9] Ибрагимова Д. Э., “Сложение векторов на прямой с помощью клеточного автомата с локаторами”, *Интеллектуальные системы. Теория и приложения*, **26**:4 (2022), 134–162.
- [10] Гасанов Э. Э., “Клеточные автоматы с локаторами как модель устройств с беспроводной связью”, *Математические вопросы кибернетики*, **21** (2023), 5–51.
- [11] Васильев Д. И., “Поиск ближайшего соседа на прямой с помощью клеточного автомата с локаторами”, *Интеллектуальные системы. Теория и приложения*, **24**:3 (2020), 99–119.

### **Fast algorithms for multiplication and division of natural numbers using cellular automata with locators**

**Gasanov E.E., Khaybullin B.F.**

For multiplication and division of  $n$ -digit natural numbers, algorithms with complexity of order  $n^{\log_2 3}$  and even order  $n^{\log n}$  are known. In this paper, an algorithm for multiplying  $n$ -digit natural numbers in  $2n + 2$  cycles is proposed. Here, the digit of number  $a$  is understood as the number  $\lfloor \log_2 a \rfloor$ . For division of natural numbers with remainder, an algorithm with a running time of  $3n + 8$  cycles is proposed, where  $n$  is the digit of the quotient. The proposed algorithms use two-dimensional cellular automata with locators as calculators.

*Keywords:* multiplication of natural numbers, division of natural numbers, cellular automata with locators.

## **References**

- [1] Karatsuba A., Ofman Yu., “Multiplication of multi-digit numbers on automata”, *Reports of the USSR Academy of Sciences*, **145**:2 (1962), 293–294 (In Russian).
- [2] Schönhage A., Strassen V., “Schnelle Multiplikation großer Zahlen”, *Computing*, 1971, № 7, 281–292.
- [3] Fürer M., “Faster integer multiplication”, *STOC 2007 Proceedings*, 2007, 57–66.

- [4] David Harvey, Joris van der Hoeven, “Integer multiplication in time  $O(n \log n)$ ”, *Annals of Mathematics*, **193**:2 (2021), 563–617.
- [5] Christoph Burnikel C., Ziegler J., “Fast Recursive Division”, *Max-Planck-Institut für Informatik*, 1998.
- [6] Gasanov E. E., “Linear in order algorithm for multiplication of numbers using a two-dimensional cellular automaton with locators”, *International scientific conference "Mathematics in the constellation of sciences" to the anniversary of academician V.A. Sadovnichy, Moscow, Russia, April 1-2, 2024*, p.316-318 (In Russian).
- [7] Gasanov E. E., “Cellular automata with locators”, *Intelligent Systems. Theory and Applications*, **24**:2 (2020), 120–133 (In Russian).
- [8] Kalachev G. V., “Notes on the Definition of a Cellular Automaton with Locators”, *Intelligent Systems. Theory and Applications*, **24**:4 (2020), 47–56 (In Russian).
- [9] Ibragimova D. E., “Vector addition on a line using a cellular automaton with locators”, *Intelligent Systems. Theory and Applications*, **26**:4 (2022), 134–162 (In Russian).
- [10] Gasanov E. E., “Cellular automata with locators as a model for wireless communication devices”, *Mathematical issues of cybernetics*, **21** (2023), 5–51 (In Russian).
- [11] Vasilev D. I., “Finding the nearest neighbor on a line using a cellular automaton with locators”, *Intelligent Systems. Theory and Applications*, **24**:3 (2020), 99–119 (In Russian).