

# СОДЕРЖАНИЕ

|           |  |           |
|-----------|--|-----------|
| <b>1</b>  | <b>Введение</b>  | <b>3</b>  |
| <b>2</b>  | <b>Используемые сокращения</b>                           | <b>3</b>  |
| <b>3</b>  | <b>Постановка задачи</b>                                 | <b>4</b>  |
| <b>4</b>  | <b>Интерфейс устройства</b>                              | <b>4</b>  |
| <b>5</b>  | <b>Формат данных</b>                                     | <b>4</b>  |
| <b>6</b>  | <b>Назначение контактов</b>                              | <b>6</b>  |
| <b>7</b>  | <b>Математическое обоснование алгоритмов</b>             | <b>6</b>  |
| 7.1       | Деление двух целых чисел в дополнительном коде . . . . . | 6         |
| 7.2       | Сложение чисел в экспоненциальной форме . . . . .        | 8         |
| <b>8</b>  | <b>Блок-схемы алгоритмов</b>                             | <b>10</b> |
| 8.1       | Деление двух целых чисел в дополнительном коде . . . . . | 10        |
| 8.2       | Сложение чисел в экспоненциальной форме . . . . .        | 10        |
| <b>9</b>  | <b>Описание микрокоманд</b>                              | <b>10</b> |
| <b>10</b> | <b>Функциональная схема<br/>операционного автомата</b>   | <b>12</b> |
| 10.1      | Деление двух целых чисел в дополнительном коде . . . . . | 12        |
| 10.2      | Сложение чисел в экспоненциальной форме . . . . .        | 13        |
| <b>11</b> | <b>Типовые примеры</b>                                   | <b>14</b> |
| <b>12</b> | <b>Управляющий автомат</b>                               | <b>15</b> |
| <b>13</b> | <b>Заполнение памяти</b>                                 | <b>17</b> |
| <b>14</b> | <b>Заключение</b>  | <b>19</b> |
|           | <b>БИБЛИОГРАФИЧЕСКИЙ СПИСОК</b>                          | <b>20</b> |

## 1 Введение

Целью данной курсовой работы является овладение навыками проектирования синхронных конечных автоматов, применение теоретических знаний, полученных в ходе обучения, на практике.

В любом устройстве обработки цифровой информации можно выделить два основных блока – операционный автомат (ОА) и управляющий автомат (УА). Микрооперации, выполняемые ОА, задаются множеством *управляющих* сигналов  $Y \{y_1, \dots, y_M\}$ , с каждым из которых отождествляется определенная микрооперация.

Значения логических условий, вычисляемые в операционном автомате, отображаются множеством *осведомительных* сигналов  $X = \{x_1, \dots, x_L\}$ , каждый из которых отождествляется с определенным логическим условием.

Управляющий автомат (УА) генерирует последовательность управляющих сигналов, предписанную микропрограммой и соответствующую значениям логическим условий. Управляющий автомат задает порядок выполнения действий в ОА, вытекающий из алгоритма выполнения операций. Наименование операции, которую необходимо выполнить в устройстве, определяется кодом  $g$  операции, поступающим в УА извне.

## 2 Используемые сокращения

ПЗ — плавающая запятая

УА — управляющий автомат

ОА — операционный автомат

IEEE — (*англ. Institute of Electrical and Electronics Engineers*) институт инженеров электротехники и электроники

ПЗУ — Постоянное запоминающее устройство (*англ. ROM — Read-only Memory*)

DI — Входная шина данных

DO — Выходная шина данных

COP — Код операции

RI — Сигнал готовности данных  
RO — Выходной сигнал готовности  
OW — Сигнал переполнения разрядной сетки  
COMP — компаратор  
CT — счетчик  
MX — мультиплексор  
RG — регистр  
SM — сумматор

### 3 Постановка задачи

Разработать вычислительное устройство, состоящее из двух взаимосвязанных частей — операционного и управляющего автоматов, и выполняющее следующие операции:

1. Деление двух целых чисел в дополнительном коде;
2. Сложение чисел, представленных в экспоненциальном формате.

Операнды представлены в виде 32-х двоичных разрядов. Управляющий автомат реализовать по схеме с регулярной адресацией в последовательном варианте.

### 4 Интерфейс устройства

Приведем интерфейс разрабатываемого вычислительного устройства, обрабатывающий и формирующий на выходе 32-х разрядные числа (см. рис. 1). Представим устройство в виде композиции управляющего автомата (УА) и операционного автомата (ОА) — рис. 2.

### 5 Формат данных

Данное вычислительное устройство оперирует с 32-х разрядными двоичными числами. От корректности их представления в конечном итоге зависит корректность производимых вычислений, поэтому соблюдение формата

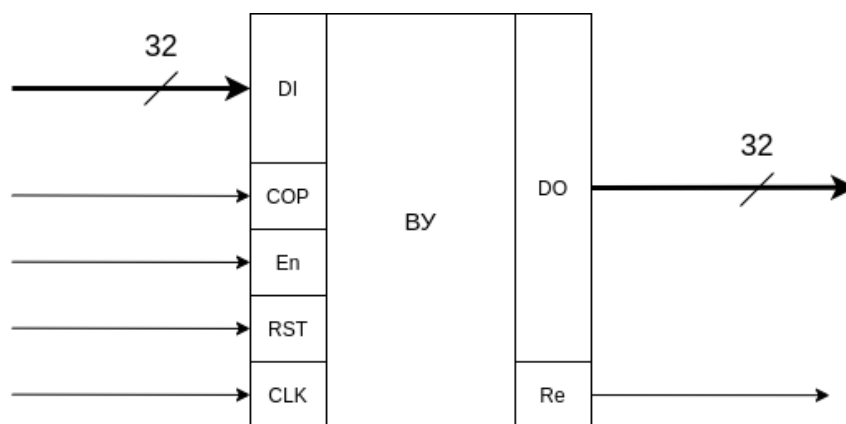


Рис. 1: Интерфейс устройства

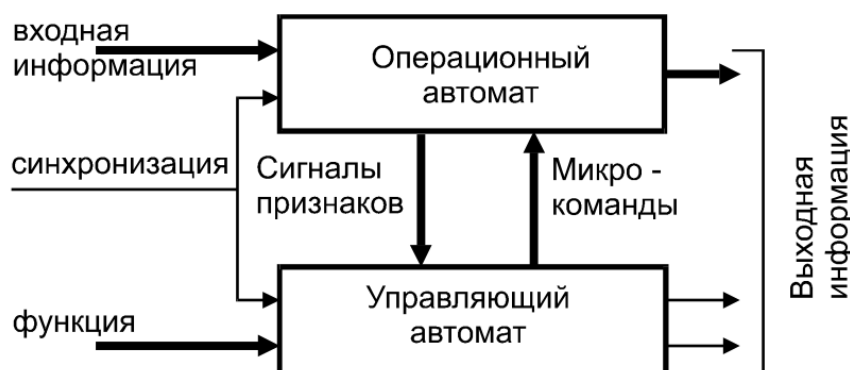


Рис. 2: Композиция УА и ОА

ввода/вывода критически важно при разработке вычислительного устройства.

Уточним принцип распознавания двоичных чисел, воспринимаемых вычислительным устройством.

При выборе первой операции (деления двух целых чисел) будем использовать дополнительный код. Число будет представлено в виде 32-х разрядного двоичного числа.

При выборе второй операции (сложения двух чисел) будем использовать стандарт IEEE 754, описывающий формат представления числе с плавающей точкой. Операнды представим в формате одинарной точности (binary32) (рис. 3). В этом формате под мантиссу числа отводится 24 разряда, а под экспоненту — 8 разрядов. При чем один разряд отдан под знак. Смещение экспоненты в данном случае  $2^7 - 1 = 127$ . Минимальное и максимальное значение мантиссы, соответственно,  $E_{min} = -126$   $E_{max} = 127$ .

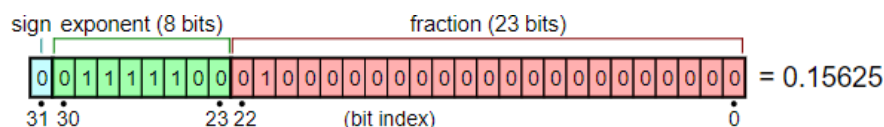


Рис. 3: Число в формате одинарной точности

## 6 Назначение контактов

Приведем назначение контактов разрабатываемого устройства согласно условному графическому обозначению интерфейса, приведенному в разделе 4 (см. таблицы 1 и 2).

| Вход | Назначение                         |
|------|------------------------------------|
| DI   | Входная 32-х разрядная шина данных |
| COP  | Код операции                       |
| En   | Разрешающий сигнал                 |
| RST  | Аппаратный сброс                   |
| CLK  | Синхронизирующий сигнал            |

Таблица 1: Входы устройства

| Выход | Назначение                          |
|-------|-------------------------------------|
| DO    | Выходная 32-х разрядная шина данных |
| Re    | Сигнал готовности результата        |

Таблица 2: Выходы устройства

## 7 Математическое обоснование алгоритмов

### 7.1 Деление двух целых чисел в дополнительном коде

Для деления двух целых чисел представленных в двоичном дополнительном коде реализуем алгоритм деления без восстановления остатка. Данный способ деления, в отличие от алгоритма с восстановлением остатка, является оптимальным по суммарному времени, так как обработка очередного разряда результата осуществляется за один такт, ведь не нужно выполнять сложение для восстановления частичного остатка.

Опишем алгоритм деления следующим образом:

1. Исходное значение частичного остатка полагается равным старшим разрядам делимого — если делимое отрицательное, то все биты регистра частичного остатка устанавливаем в «1», если делимое положительное, то все биты регистра частичного остатка устанавливаем в «0»;
2. Частичный остаток удваивается путем сдвига на один разряд влево. При этом в освобождающийся при сдвиге младший разряд заносится очередная цифра делимого.
3. Анализируем знаки остатка и делителя — в случае, если их знаки одинаковые, то выполняем вычитание делителя из остатка (прибавляем противоположное число), полученного на данном этапе. Иначе же прибавляем значение делителя к значению остатка
4. Анализируем значение остатка после выполнения арифметических действий — заносим в частное инвертированный знак остатка, вычисленного на данном этапе, вместе с этим сдвигая его влево.
5. Повторяем пункты 1-4 до тех пор, пока не будут сдвинуты все разряды делимого.

Стоит отметить, что для формирования правильного выходного результата после выполнения вышеперечисленных пунктов необходимо выполнить коррекцию значений частного и остатка в зависимости от знаков операндов. Для каждой комбинации знаков делимого и делителя реализована отдельная операция коррекции. См таблицу 3.

| Комбинация        | Коррекция   |
|-------------------|---|
| $A \geq 0, B > 0$ | Коррекция не требуется  |
| $A \geq 0, B < 0$ | Изменить знак частного, остаток должен быть положительным   |
| $A < 0, B > 0$    | Результат верен, если остаток = 0. Иначе прибавить к отрицательному частному единицу. Остаток должен быть отрицательным |
| $A < 0, B < 0$    | Перед делением изменить знак делимого. Остаток должен быть отрицательными.  |

Таблица 3: Коррекция результата

## 7.2 Сложение чисел в экспоненциальной форме

В арифметике с плавающей запятой сложение и вычитание — более сложные операции, чем умножение и деление. Обусловлено это необходимостью выравнивания порядков операндов. Алгоритм сложения и вычитания включает в себя следующие основные фазы:

1. Определение операнда, имеющего меньший порядок, и сдвиг его мантиссы вправо на число разрядов, равное разности порядков операндов;
2. Приравнивание порядка результата большему из порядков операндов;
3. Сложение или вычитание мантисс и определение знака результата;
4. Проверку на переполнение;

С начала производится проверка с целью выяснения, не равен ли нулю один из операндов. Если это имеет место, в качестве результата сразу берется другой операнд.

В следующей фазе осуществляется выравнивание порядков обоих операндов. Для пояснения рассмотрим например сложения десятичных чисел с плавающей запятой:  $123 \cdot 10^0 + 456 \cdot 10^{-2}$

Очевидно, что непосредственное сложение мантисс недопустимо, поскольку цифры мантисс, имеющие одинаковый вес, должны располагаться в эквивалентных позициях. Так, цифра 4 во втором числе должна суммироваться с цифрой 3 в первом. Этого можно добиться, если записать второе число так, чтобы порядки обоих чисел были равны:

$$123 \cdot 10^0 + 456 \cdot 10^{-2} = 123 \cdot 10^0 + 4,56 \cdot 10^0 = 127,56 \cdot 10^0$$

Выравнивание порядков операндов реализуется путем сдвига мантиссы меньшего из чисел на один разряд вправо с одновременным увеличением порядка этого числа на единицу. Действия повторяются до совпадения порядков. Если в процессе сдвига мантисса обращается в 0, то в качестве результата операции берется другой операнд.

Следующая фаза — сложение мантисс с учетом их знаков, что при одинаковых знаках мантисс может привести к переполнению. В последнем случае мантисса результата сдвигается вправо на один разряд, а порядок результата увеличивается на единицу. Это, в свою очередь, чревато переполнением

поля порядка. Тогда операция прекращается и формируется *признак переполнения*, сопровождаемый соответствующим предупреждением (обычно в виде сигнала прерывания).

В отличие от целочисленной арифметики, в операциях с ПЗ сложение и вычитание производятся приближенно, так как при выравнивании порядков происходит потеря младших разрядов одного из слагаемых. В этом случае погрешность всегда отрицательна и может доходить до единицы младшего разряда. При выполнении операции сложения предполагается, что числа, переданные на вход находятся в нормализованном виде, то есть имеют вид, представленный на сноске 1.

$$\begin{aligned} \frac{1}{2} &\leq |M| < 1 \\ M &= 0.1XXXX \\ M &= 1.0XXXX \\ M &= 1.00000 \end{aligned} \tag{1}$$

Результат суммы также нормализуется в соответствии с данными правилами. Числа, представленные в ином виде считаются ненормализованными и не обрабатываются цифровым устройством.

Стоит отметить, что для формирования правильного выходного результата необходимо выполнить нормализацию значений суммы в зависимости от вида операндов. Для каждой комбинации операндов реализована отдельная операция нормализации. См таблицу 4.

| Комбинация                           | Коррекция   |
|--------------------------------------|---|
| $ m_a \pm m_b  \geq 1$               | Мантисса не нормализована. Сдвинуть регистр мантиссы вправо, загрузить сигнал переноса сумматора. Увеличить порядок результата на 1. При этом может произойти переполнение счетчика в большую сторону |
| $\frac{1}{2} \leq  m_a \pm m_b  < 1$ | Нормализация результата не требуется  |
| $ m_a \pm m_b  < \frac{1}{2}$        | Мантисса не нормализована. Сдвигая мантиссу влево, уменьшать порядок, при этом может произойти переполнение порядка в отрицательную сторону   |

Таблица 4: Нормализация результата



## 8 Блок-схемы алгоритмов

### 8.1 Деление двух целых чисел в дополнительном коде

Введем обозначения операндов, используемых в данной операции (см. таблицу 5):

| Обозначение | Назначение           |
|-------------|----------------------|
| A           | Делимое в доп. коде  |
| B           | Делитель в доп. коде |
| RES         | Частное от деления   |
| REM         | Остаток от деления   |

Таблица 5: Деление чисел. Операнды

Опишем алгоритм выполнения деления с помощью блок схемы. См. рис. 7 в Приложении А.

### 8.2 Сложение чисел в экспоненциальной форме

Введем обозначения операндов, используемых в данной операции (см. таблицу 6):

| Обозначение | Назначение                  |
|-------------|-----------------------------|
| Pa          | Порядок первого операнда    |
| Pb          | Порядок второго операнда    |
| Ma          | Мантисса первого операнда   |
| Ma          | Мантисса второго операнда   |
| dP          | Разность порядков операндов |

Таблица 6: Суммирование чисел. Операнды

Опишем алгоритм выполнения суммы с помощью блок схемы. См. рис. 8. в Приложении А.

## 9 Описание микрокоманд

Приведем смысловую таблицу микрокоманд, в которой поясним, какие изменения происходят на каждом этапе. См. таблицу 7.

| <b>A</b> | <b>N</b> | <b>Назначение микрокоманды</b>  |
|----------|----------|---|
| 0        | m0       | Начальное состояние   |
| 1        | m01      | Служебное псевдосостояние   |
| 2        | m02      | Служебное псевдосостояние   |
| 3        | m03      | Служебное псевдосостояние   |
| 4        | m1       | Загрузка первого числа с входной шины данных                                |
| 5        | m04      | Служебное псевдосостояние   |
| 6        | m2       | Загрузка второго числа с входной шины данных                                |
| 7        | m3       | Начальная инициализация   |
| 8        | m05      | Служебное псевдосостояние   |
| 9        | m4       | Сдвиг RG_REM, RG_A влево  |
| 10       | m5       | Вычисление текущего значения частичного остатка                             |
| 11       | m6       | Вычисление текущего значения частичного остатка                             |
| 12       | m06      | Служебное псевдосостояние   |
| 13       | m07      | Служебное псевдосостояние   |
| 14       | m7       | Коррекция результата  |
| 15       | m09      | Служебное псевдосостояние   |
| 16       | m8       | Коррекция результата  |
| 17       | m08      | Служебное псевдосостояние   |
| 18       | m9       | Коррекция результата  |
| 19       | m10      | Ошибка «Деление на ноль»  |
| 20       | m020     | Служебное псевдосостояние   |
| 21       | m20      | Загрузка первого числа с входной шины данных (эксп. форма)                  |
| 22       | m021     | Служебное псевдосостояние   |
| 23       | m21      | Загрузка второго числа с входной шины данных (эксп. Форма)                  |
| 24       | m022     | Служебное псевдосостояние   |
| 25       | m023     | Служебное псевдосостояние   |
| 26       | m024     | Служебное псевдосостояние   |
| 27       | m22      | Смена операндов местами   |
| 28       | m23      | Нахождение разности порядков  |
| 29       | m025     | Служебное псевдосостояние   |
| 30       | m24      | Смена операндов местами   |
| 31       | m25      | Выравнивание порядков RG_Pa, сдвиг вправо REG_Ma, увеличение счетчика CT_Pa |
| 32       | m26      | Загрузка значения суммы операндов   |
| 33       | m27      | Обработка переполнения мантиссы   |
| 34       | m026     | Служебное псевдосостояние   |
| 35       | m28      | Сдвиг RG_Ma влево, декремент счетчика CT_Pa                                 |

*См. след. страницу*

**Таблица 7 – продолжение**

| <b>A</b> | <b>N</b> | <b>Назначение микрокоманды</b> |
|----------|----------|--------------------------------|
| 36       | m30      | Ошибка «Переполнение»          |
| 37       | m29      | Вывод значения на шину данных  |
| 38       | m31      | Ошибка «Деление на ноль»       |
| 39       | m32      | Вывод значения на шину данных  |
| 40       | m33      | Вывод значения на шину данных  |
| 41       | m34      | Вывод значения на шину данных  |

Таблица 7: Смысловая таблица микрокоманд

## **10 Функциональная схема операционного автомата**

### **10.1 Деление двух целых чисел в дополнительном коде**

Приведем названия и назначения каждого из регистров. См. таблицу 8.

| <b>Идентификатор</b> | <b>Назначение</b>                                    |
|----------------------|--|
| <i>RG_A</i>          | Сдвиговый регистр. Хранит разряды делимого           |
| <i>RG_B</i>          | Хранит разряды делителя                              |
| <i>RG_REM</i>        | Сдвиговый регистр. Хранит разряды частичного остатка |
| <i>RG_RES</i>        | Сдвиговый регистр. Хранит разряды результата         |

Таблица 8: Регистры операционного автомата

Укажем необходимые признаки, которые впоследствии будут вырабатываться управляющим автоматом в ходе выполнения первой операции. См. таблицу 9.

Приведем функциональную схему операционного автомата, выполняющего деление двух целых чисел в дополнительном коде по алгоритму деления без восстановления остатков. См. рис. 4.

| Признак      | Назначение   |
|--------------|--|
| $S$          | Хранит адрес следующей операции  |
| $H$          | Адресный вход мультиплексора   |
| $R0$         | Сигнализирует об окончании операции деления  |
| $ER$         | Сигнализирует об ошибке ввода – делитель равен нулю  |
| $L\_RG_A$    | Загрузка в регистр $RG\_A$   |
| $L\_RG_B$    | Загрузка в регистр $RG\_B$   |
| $L\_RG\_REM$ | Загрузка в регистр $RG\_REM$   |
| $CLR$        | Асинхронный сброс всех элементов   |
| $CT$         | Счет. Декремент счетчика, если $L\_CT == 1$  |
| $L\_CT$      | Загрузка счетчика  |
| $SHFT$       | Левый сдвиг в регистрах $RG\_A$ и $RG\_RES$  |
| $e$          | Управляющий сигнал для счетчика. Если $e = 1$ , следует выполнить загрузку, а если $e = 0$ – инкрементировать счетчик. |

Таблица 9: Деление чисел. Осведомительные сигналы (признаки)

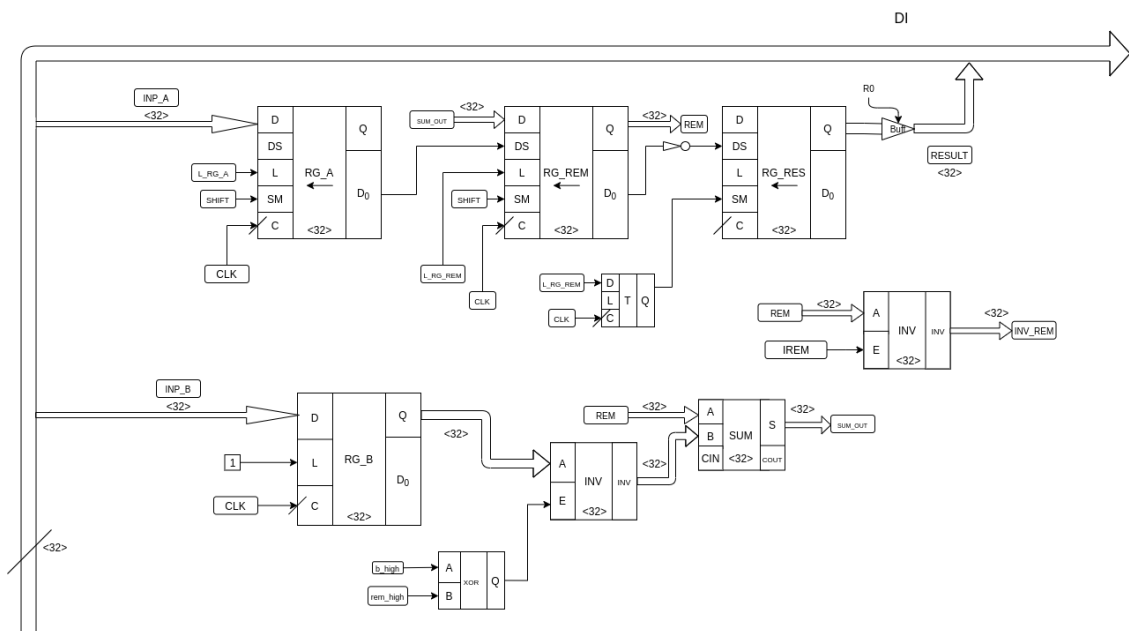


Рис. 4: Деление чисел. Операционный автомат

## 10.2 Сложение чисел в экспоненциальной форме

Приведем названия и назначения каждого из регистров, используемых в данном устройстве. См. таблицу 10.

Укажем необходимые признаки, которые впоследствии будут вырабатываться управляющим автоматом в ходе выполнения второй операции. См. таблицу 11.

Приведем функциональную схему операционного автомата, выполняющего сложение двух чисел, представленных в экспоненциальном формате. См. рис. 5.

| Идентификатор | Назначение   |
|---------------|--|
| $RG\_Ma$      | Универсальный сдвиговый регистр. Хранит разряды мантиссы А       |
| $CT\_Mb$      | Счетчик. Хранит разряды мантиссы В                               |
| $CT\_Pa$      | Счетчик. Хранит разряды порядка числа А                          |
| $CT\_Pb$      | Счетчик. Хранит разряды порядка числа В                          |
| $CT\_dP$      | Счетчик. Хранит разряды разницы порядков чисел А и В             |
| $T\_SUM$      | Триггер. Хранит разряд сигнала переноса суммы мантис чисел А и В |

Таблица 10: Регистры операционного автомата

| Признак     | Назначение   |
|-------------|--|
| $S$         | Хранит адрес следующей операции  |
| $H$         | Адресный вход мультиплексора   |
| $R0$        | Сигнализирует об окончании операции деления  |
| $ER$        | Сигнализирует об ошибке ввода – операнды не нормализованы  |
| $OWF$       | Сигнализирует об ошибке обработки – переполнение   |
| $L\_Ma$     | Загрузка в регистр $RG\_Ma$  |
| $SHFT\_Ma$  | Правый сдвиг регистра $RG\_Ma$ если $SHIFT\_Ma\_Left = 0$ и левый, если $SHIFT\_Ma\_Left = 1$                          |
| $RST$       | Асинхронный сброс всех элементов   |
| $CNT\_Pa$   | Счет. Декремент счетчика, если $L\_CT\_Pa == 1$  |
| $CNT\_dP$   | Счет. Декремент счетчика, если $L\_CT\_dP == 1$  |
| $L\_CT\_Pa$ | Загрузка счетчика $CT\_Pa$   |
| $CHANGE$    | Выбор источника загрузки в регистры мантис и порядка чисел А и В   |
| $e$         | Управляющий сигнал для счетчика. Если $e = 1$ , следует выполнить загрузку, а если $e = 0$ – инкрементировать счетчик. |

Таблица 11: Сложение чисел в экспоненциальной форме. Осведомительные сигналы (признаки)

## 11 Типовые примеры

Приведем пример вычисления частного от деления чисел  $(-13_{10} = 10011_2) \div 3_{10} = 00011_2$ . См. таблицу 12.

В результате вычислений получим частное  $(-5_{10}) = 11011_2$  и остаток  $2_{10} = 00010_2$ .

Приведем пример вычисления суммы двух 16-ти разрядных чисел, используя рассматриваемый алгоритм (см. таблицу 13).

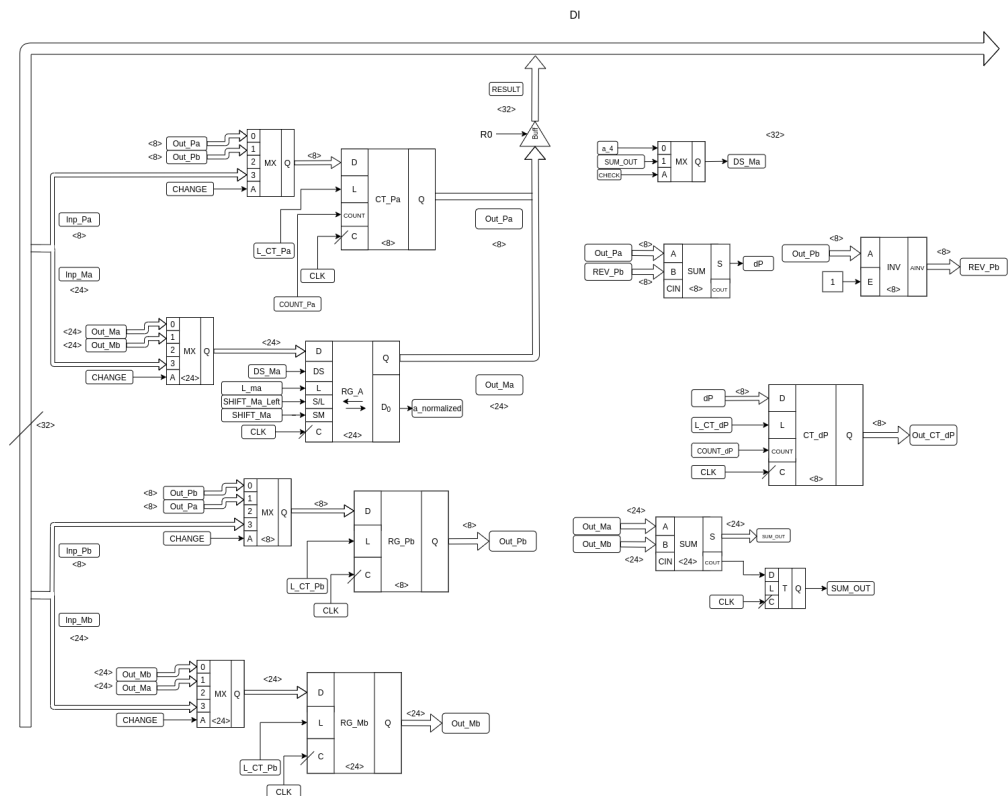


Рис. 5: Сложение чисел. Операционный автомат

| Частное | Остаток | Делимое | Операция                                    |
|---------|---------|---------|---|
|         | 11111   | 10011   |   |
|         | 11111   | 0011x   | Сдвиг остатка                               |
|         | 00011   |         | Сложение с делителем                        |
| 1       | 00010   |         | Результат сложения — положительный остаток  |
|         | 00100   | 011xx   | Сдвиг остатка                               |
|         | 11101   |         | Вычитание делителя                          |
| 1       | 00001   |         | Результат вычитания — положительный остаток |
|         | 00010   | 11xxx   | Сдвиг остатка                               |
|         | 11101   |         | Вычитание делителя                          |
| 0       | 11111   |         | Результат вычитания — отрицательный остаток |
|         | 11111   | 1xxxx   | Сдвиг остатка                               |
|         | 00011   |         | Сложение с делителем                        |
| 1       | 00010   |         | Результат вычитания — положительный остаток |
|         | 00101   | xxxxx   | Сдвиг остатка                               |
|         | 11101   |         | Вычитание делителя                          |
| 1       | 00010   |         | Результат вычитания — положительный остаток |
|         | 11111   |         | Восстановленный отрицательный остаток       |

Таблица 12: Пример деления целых чисел в доп. коде

## 12 Управляющий автомат

Управляющий автомат был построен по схеме с регулярной адресацией (последовательный вариант). Рассмотрим строение управляющего автомата.

|   | Ma          | Pa    | Mb         | Pb    | dP    |
|---|-------------|-------|------------|-------|-------|
| Входные данные  | 01010101000 | 00111 | 0101111111 | 00100 | —     |
| Находим разность порядков   |             |       |            |       | 00011 |
| Сдвигаем меньшее число вправо, увеличивая его порядок               | 01010101000 | 00111 | 01011111   | 00111 | 00000 |
| Суммируем мантиссы. Записываем результат в разряды первого операнда | 01100000111 | 00111 | 01011111   | 00111 | 00000 |
| Проверяем нормализованность результата                              | 01100000111 | 00111 | 01011111   | 00111 | 00000 |
| Ответ   | 01100000111 | 00111 |            |       |       |

Таблица 13: Пример вычисления суммы двух чисел в экспоненциальном формате

См рисунок 6.

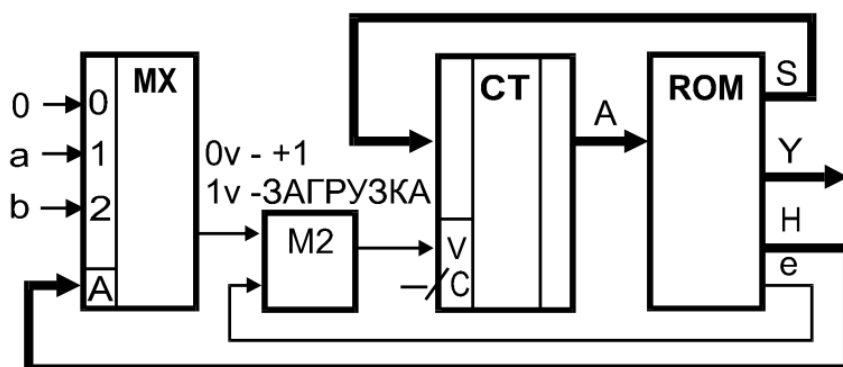


Рис. 6: УА с регулярной адресацией

На вход УА подаются сигналы от операционного автомата соответствующие логическим блокам алгоритма. С выхода управляющего автомата снимаются микроинструкции хранящиеся в ПЗУ (ROM) УА. Микроинструкции обеспечивают наличие необходимых управляющих сигналов на элементах операционного автомата в соответствии с выбранным блоком алгоритма. Также в ПЗУ содержится адресная часть позволяющая в следующем такте работы выбрать новый адрес управляющей памяти.

Мультиплексор обеспечивает выбор входного сигнала поступившего от ОА в соответствии с адресом хранящимся в ПЗУ. Элемент M2 позволяет инвертировать значения входного сигнала что обеспечивает подстройку УА под конкретные схемотехнические решения.

Счётчик при поступлении на вход W нуля производит загрузку адреса микроинструкции на вход S', а при поступлении единицы осуществляет инкрементацию адреса, хранящегося в счётчике.

## 13 Заполнение памяти

В конкретной реализации на информационные входы мультиплексора подаются следующие сигналы: (см. таблицу 14).

| Входной сигнал | Назначение   |
|----------------|--|
| RI             | Признак готовности ввода                                 |
| COP            | Код операции   |
| B_IS_NULL      | Второй операнд равен нулю                                |
| CT_IS_NULL     | Первый операнд равен нулю                                |
| DIFF_SIGN      | Знаки делителя и частного остатка различны               |
| A>=0           | Делимое неотрицательно                                   |
| B>0            | Делитель положителен                                     |
| REM_IS_NULL    | Остаток равен нулю                                       |
| op_normalize   | Операнды нормализованы                                   |
| Mb_IS_NULL     | Мантисса второго операнда равна нулю                     |
| Ma_IS_NULL     | Мантисса первого операнда равна нулю                     |
| Pa<Pb          | Порядок первого операнда больше порядка второго операнда |
| CT_dP_IS_NULL  | Счетчик разности порядков обнулен                        |
| dP>24          | Разность порядков превышает возможный диапазон           |
| ma+mb >1       | Переполнение мантиссы                                    |
| CT_Pa_IS_NULL  | Порядок первого операнда равен нулю                      |
| CT_Pa_IS_MAX   | Переполнение счетчика порядка первого операнда           |

Таблица 14: Управляющий автомат. Анализируемые сигналы

Приведем таблицу заполнения ПЗУ управляющего автомата, построенного по схеме с регулярной адресацией в последовательном варианте в таблице 15.

| A | Y          | S  | H | e | A  | Y           | S  | H  | e |
|---|------------|----|---|---|----|-------------|----|----|---|
| 0 | m0         | 1  | 0 | 0 | 21 | m20         | 22 | 0  | 0 |
| 1 | <b>m01</b> | 1  | 1 | 1 | 22 | <b>m021</b> | 22 | 1  | 1 |
| 2 | <b>m02</b> | 2  | 2 | 0 | 23 | m21         | 38 | 9  | 1 |
| 3 | <b>m03</b> | 20 | 1 | 1 | 24 | <b>m022</b> | 39 | 10 | 0 |
| 4 | m1         | 5  | 0 | 0 | 25 | <b>m023</b> | 40 | 11 | 0 |
| 5 | <b>m04</b> | 5  | 1 | 1 | 26 | <b>m024</b> | 28 | 12 | 1 |
| 6 | m2         | 7  | 0 | 0 | 27 | m22         | 28 | 0  | 0 |
| 7 | m3         | 19 | 3 | 0 | 28 | m23         | 32 | 13 | 0 |
| 8 | <b>m05</b> | 12 | 4 | 0 | 29 | <b>m025</b> | 41 | 14 | 0 |
| 9 | m4         | 11 | 5 | 1 | 30 | m24         | 31 | 0  | 0 |

См. след. страницу



**Таблица 15 – продолжение**

| <b>A</b> | <b>Y</b>    | <b>S</b> | <b>H</b> | <b>e</b> | <b>A</b> | <b>Y</b>    | <b>S</b> | <b>H</b> | <b>e</b> |
|----------|-------------|----------|----------|----------|----------|-------------|----------|----------|----------|
| 10       | m5          | 8        | 0        | 0        | 31       | m25         | 31       | 13       | 1        |
| 11       | m6          | 8        | 0        | 0        | 32       | m26         | 34       | 15       | 1        |
| 12       | <b>m06</b>  | 17       | 6        | 0        | 33       | m27         | 36       | 16       | 0        |
| 13       | <b>m07</b>  | 15       | 7        | 0        | 34       | <b>m026</b> | 37       | 9        | 0        |
| 14       | m7          | 0        | 0        | 0        | 35       | m28         | 34       | 17       | 1        |
| 15       | <b>m09</b>  | 0        | 8        | 0        | 36       | m30         | 0        | 0        | 0        |
| 16       | m8          | 0        | 0        | 0        | 37       | m29         | 0        | 0        | 0        |
| 17       | <b>m08</b>  | 0        | 7        | 0        | 38       | m31         | 0        | 0        | 0        |
| 18       | m9          | 0        | 0        | 0        | 39       | m32         | 0        | 0        | 0        |
| 19       | m10         | 0        | 0        | 0        | 40       | m33         | 0        | 0        | 0        |
| 20       | <b>m020</b> | 20       | 1        | 1        | 41       | m34         | 0        | 0        | 0        |

Таблица 15: Заполнение ПЗУ

Приведем таблицу переходов для управляющего автомата выполненного по схеме с регулярной адресацией. См. таблицу **16** в **Приложении А**.

## 14 Заключение

В ходе данной курсовой работы был рассмотрен алгоритм деления двух целых чисел без восстановления остатка дополнительном коде и алгоритм нахождения суммы двух чисел, представленных в формате с одинарной точностью.

Рассмотренные алгоритмы изучены и реализованы на спроектированном вычислительном устройстве — синхронном конечном автомате, управляющий автомат которого выполнен по схеме с регулярной адресацией.

В ходе выполнения курсовой работы были приобретены необходимые навыки и умения в проектировании операционных блоков вычислительных устройств.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Антик М.И. Теория автоматов в проектировании цифровых схем [Электронный ресурс]: Учебное пособие / Антик М.И., Казанцева Л.В. — М.: МИРЭА – Российский технологический университет, 2020.
2. Антик М.И., Синхронные цифровые автоматы [Электронный ресурс]: монография / М.И. Антик, А.М. Романов. — М.: МГТУ МИРЭА, 2014. — Электрон. опт. диск (ISO), НТБ МИРЭА А72
3. Антик М.И., Триггеры [Электронный ресурс]: учебное пособие для студентов, обучающихся по направлениям подготовки 230100 и спец. 230101 "Вычислительные машины, системы, комплексы и сети"/ М.И. Антик, А.М. Романов. — М.: МГТУ МИРЭА, 2012. — Электрон. опт. диск (ISO), НТБ МИРЭА А72
4. Зайцев Е.И., Прикладная теория цифровых автоматов: учебное пособие / Е.И. Зайцев, В. В. Макаров. — М.: МИРЭА, 2018. — 112 с. — Библиогр.: с. 111 (7 назв.), НТБ МИРЭА 3-17
5. Горбатов В.А. Теория автоматов: учеб. для студентов вузов – М.: АСТ: Астрель. 2008. – 559 с.
6. Карпов Ю.Г. Теория автоматов /– СПб: Питер. 2003. – 208 с.

# ПРИЛОЖЕНИЕ А

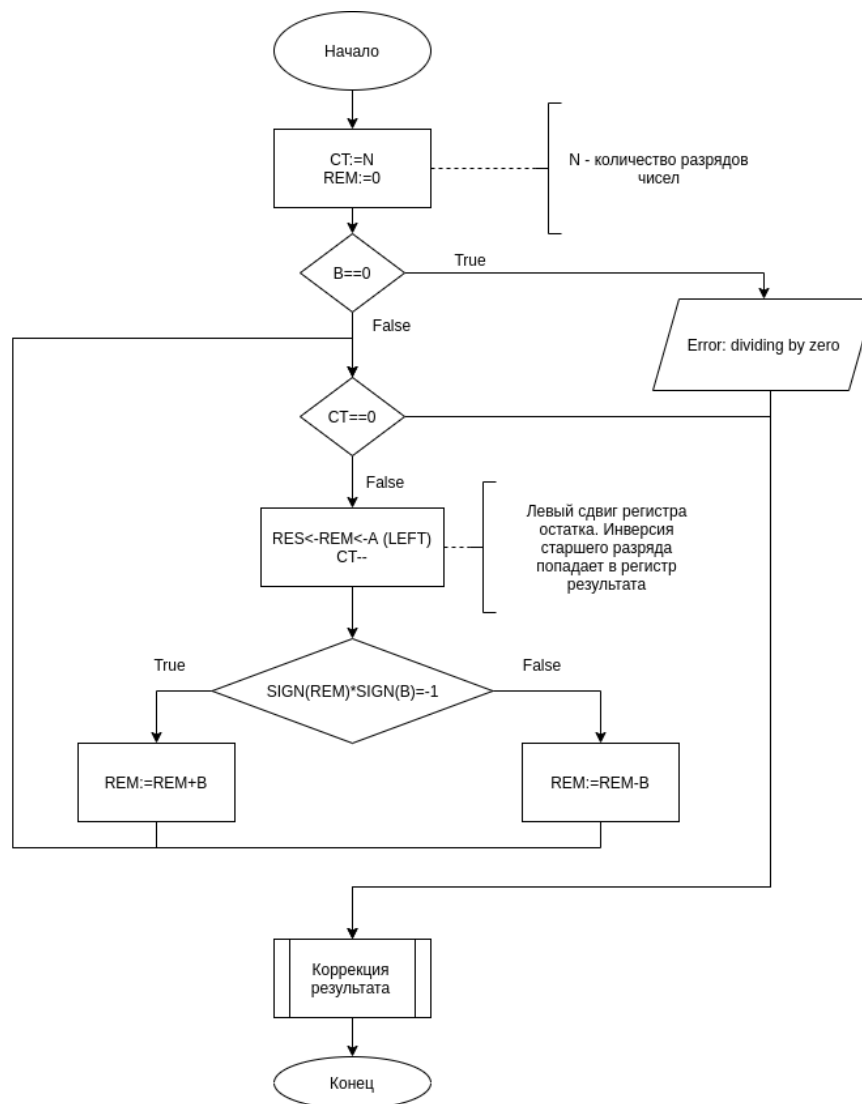


Рис. 7: Алгоритм деления чисел

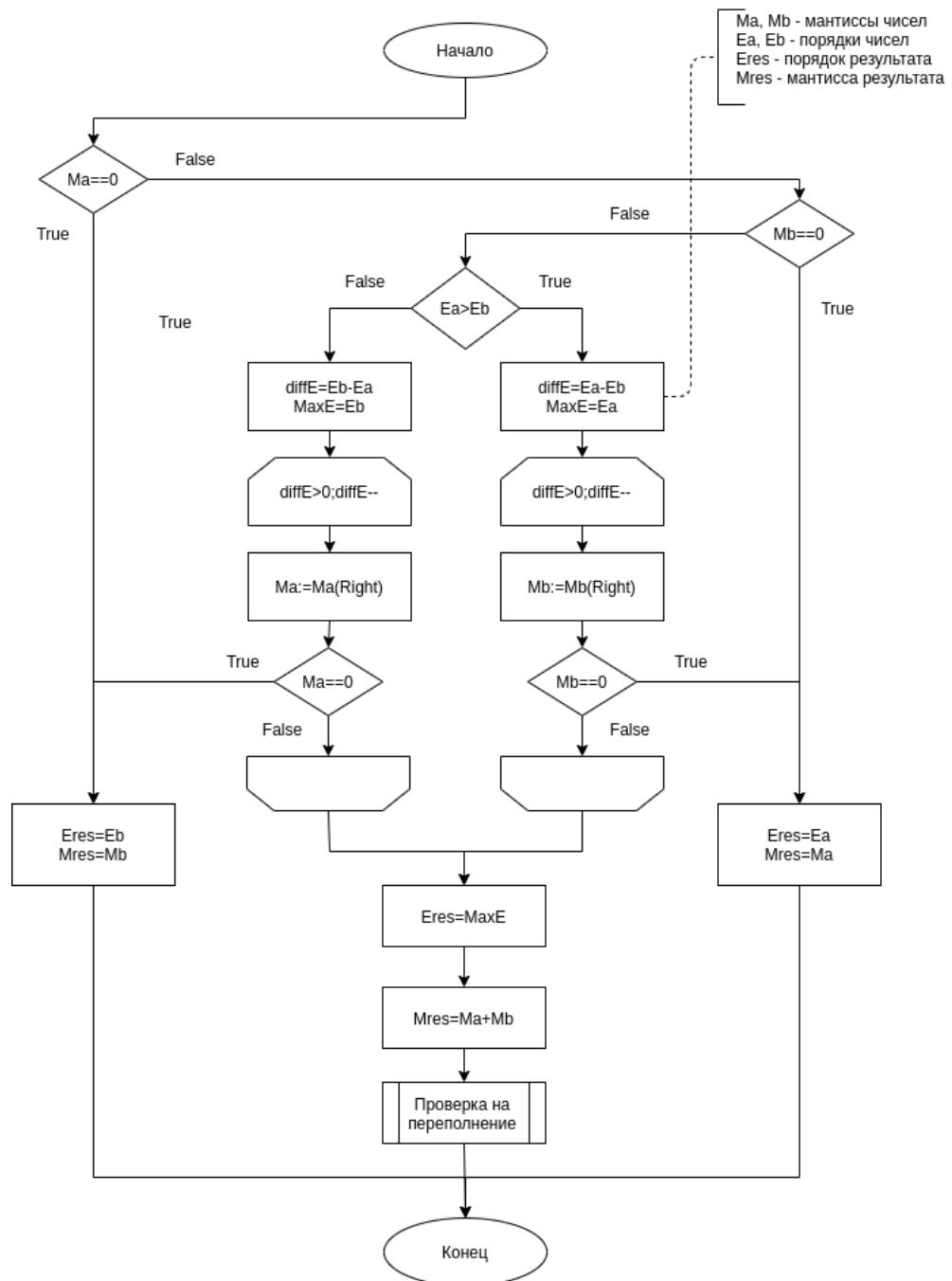


Рис. 8: Алгоритм суммирования чисел в доп. коде

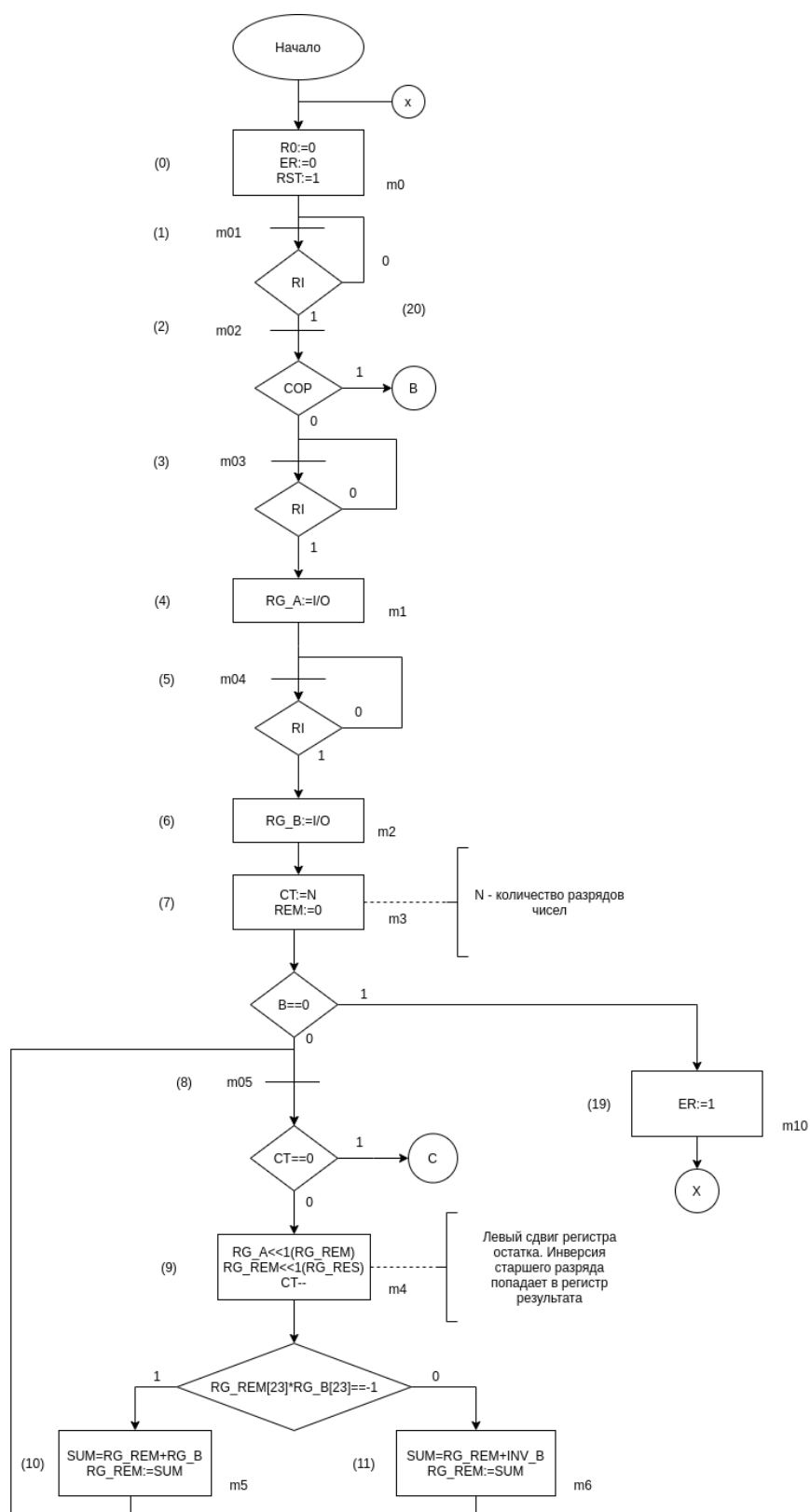


Рис. 9: Блок-схема работы автомата. Часть 1

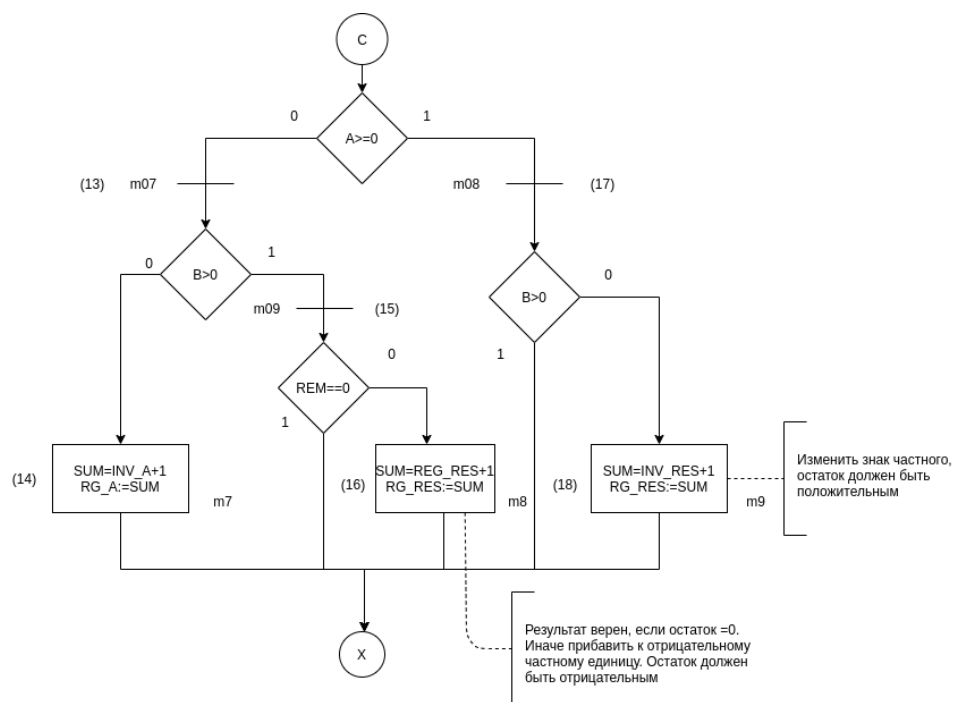


Рис. 10: Блок-схема работы автомата. Часть 2

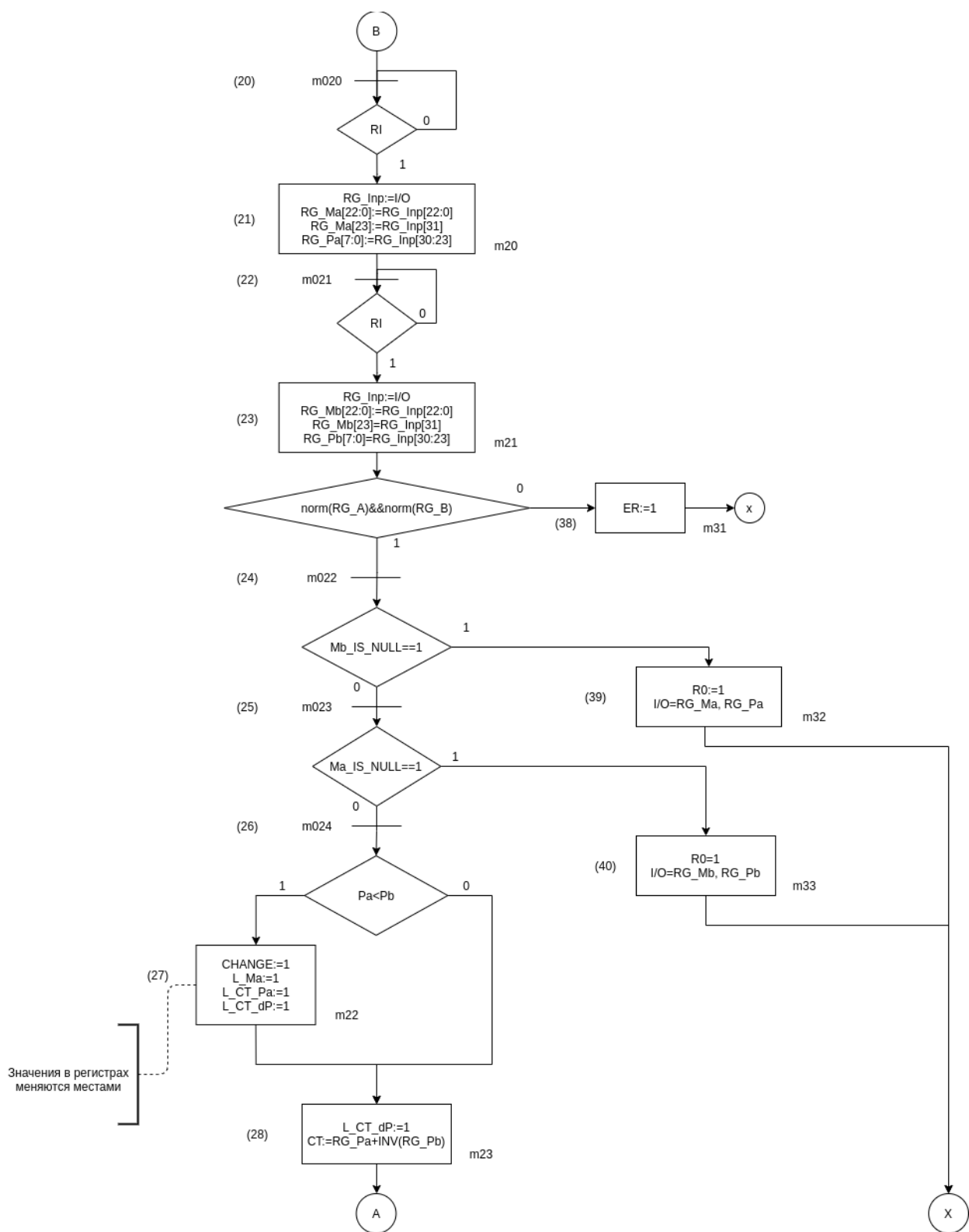


Рис. 11: Блок-схема работы автомата. Часть 3



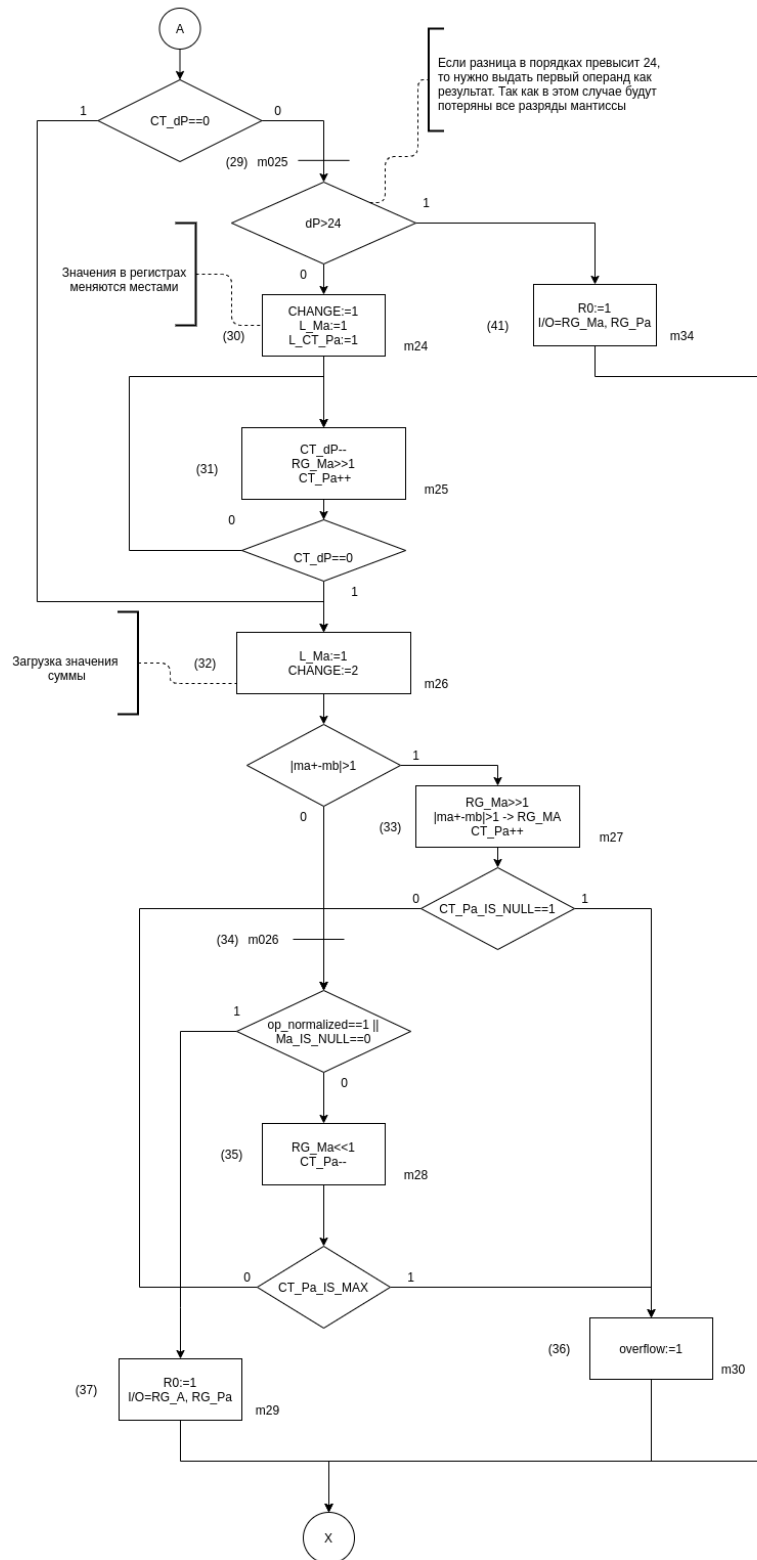


Рис. 12: Блок-схема работы автомата. Часть 4

| A  | Y    | R0 | ER | RST | L_ma | SHIFT_Ma | SHIFT_Ma_LEFT | L_CT_Pa | CHANGE | COUNT_Pa | L_CT_dP | COUNT_dP | OWF | L_RG_A | L_RG_B | L_REM | SHIFT | CT | L_CT |
|----|------|----|----|-----|------|----------|---------------|---------|--------|----------|---------|----------|-----|--------|--------|-------|-------|----|------|
| 0  | m0   | 0  | 0  | 1   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 1  | m01  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 2  | m02  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 3  | m03  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 4  | m1   | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 1      | 0      | 0     | 0     | 0  | 0    |
| 5  | m04  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 6  | m2   | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 1      | 0     | 0     | 0  | 0    |
| 7  | m3   | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 1    |
| 8  | m05  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 9  | m4   | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 1     | 1  | 1    |
| 10 | m5   | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 1     | 0     | 0  | 0    |
| 11 | m6   | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 1     | 0     | 0  | 0    |
| 12 | m06  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 13 | m07  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 14 | m7   | 1  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 1      | 0      | 0     | 0     | 0  | 0    |
| 15 | m09  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 16 | m8   | 1  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 1      | 0      | 0     | 0     | 0  | 0    |
| 17 | m08  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 18 | m9   | 1  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 1      | 0      | 0     | 0     | 0  | 0    |
| 19 | m10  | 0  | 1  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 20 | m020 | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 21 | m20  | 0  | 0  | 0   | 1    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 22 | m021 | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 23 | m21  | 0  | 0  | 0   | 1    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 24 | m022 | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 25 | m023 | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 26 | m024 | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 27 | m22  | 0  | 0  | 0   | 1    | 0        | 0             | 1       | 1      | 0        | 1       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 28 | m23  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 1       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 29 | m025 | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 30 | m24  | 0  | 0  | 0   | 1    | 0        | 0             | 1       | 1      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 31 | m25  | 0  | 0  | 0   | 0    | 1        | 0             | 0       | 0      | 1        | 0       | 1        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 32 | m26  | 0  | 0  | 0   | 1    | 0        | 0             | 0       | 10     | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 33 | m27  | 0  | 0  | 0   | 0    | 1        | 0             | 0       | 0      | 1        | 0       | 1        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 34 | m026 | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 35 | m28  | 0  | 0  | 0   | 0    | 1        | 1             | 1       | 0      | 1        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 36 | m30  | 0  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 1   | 0      | 0      | 0     | 0     | 0  | 0    |
| 37 | m29  | 1  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 38 | m31  | 0  | 1  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 39 | m32  | 1  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 40 | m33  | 1  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |
| 41 | m34  | 1  | 0  | 0   | 0    | 0        | 0             | 0       | 0      | 0        | 0       | 0        | 0   | 0      | 0      | 0     | 0     | 0  | 0    |

Таблица 16: Таблица заполнения памяти управляющего автомата