ОГЛАВЛЕНИЕ

1 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

OC — 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

2 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.

3 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 авто-

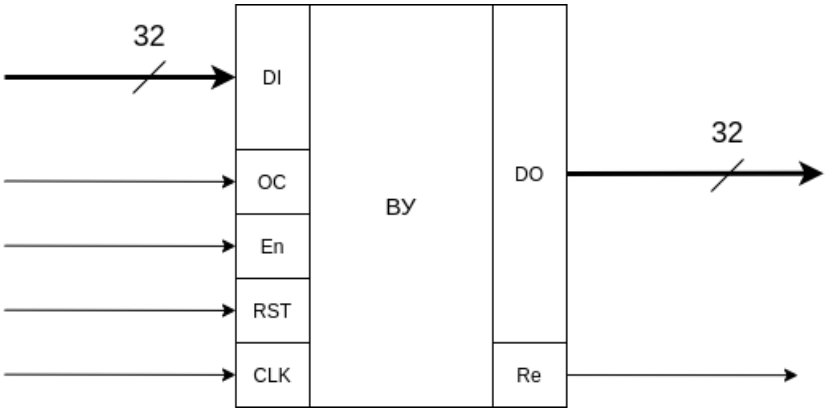

Рис. 1: Интерфейс устройства

мата (УА) и операционного автомата (ОА) — рис. 2.

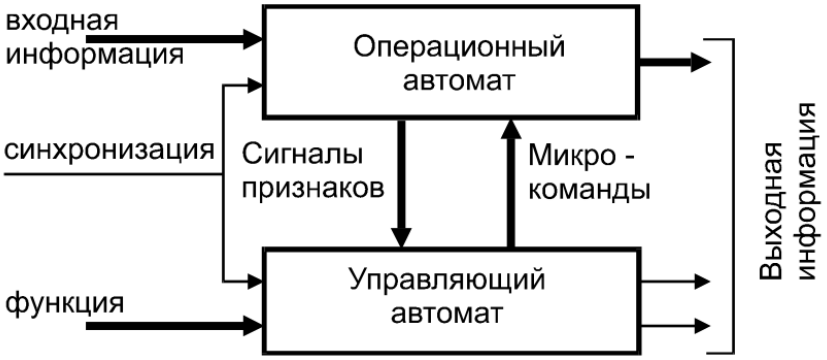

Рис. 2: Композиция УА и ОА

4 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 ввода/вывода критически важно при разработке вычислительного устройства.

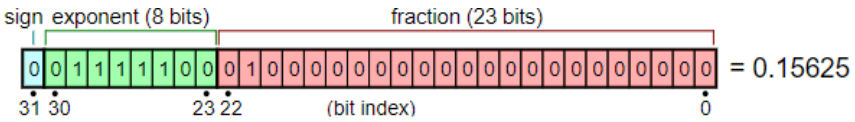

Рис. 3: Число в формате одинарной точности

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, $E_{min} = -126 E_{max} = 127$.

5 Назначение контактов

Приведем назначение контактов разрабатываемого устройства согласно условному графическому обозначению интерфейса, приведенному в разделе 3 (см. таблицы 1 и 2).

Вход	**Назначение**
DI	Входная 32-х разрядная шина данных
OC	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

6 Математическое обоснование алгоритмов

6.1 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 ре-

зультата 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**
$A \geqslant 0, B > 0$	Коррекция не требуется
$A \geqslant 0, B < 0$	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

6.2 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 па 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:

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

Выравнивания порядков можно достичь сдвигом мантиссы меньшего из чисел вправо, с одновременным увеличением порядка этого числа, либо сдвигом мантиссы большего из чисел влево и уменьшением его порядка. Оба варианта сопряжены с потерей цифр мантиссы, но выгоднее сдвигать *меньшее* из чисел, так как при этом теряются младшие разряды мантиссы.

Таким образом, 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.

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

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**
$\lvert m_a \pm m_b \rvert \geqslant 1$	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
$\frac{1}{2} \leqslant \lvert m_a \pm m_b \rvert < 1$	Нормализация результата не требуется
$\lvert m_a \pm m_b \rvert < \frac{1}{2}$	Мантисса не нормализована. Сдвигая мантиссу влево, уменьшать порядок, при этом может произойти переполнение порядка в отрицательную сторону

Таблица 4: Нормализация результата

7 Блок-схемы алгоритмов

7.1 Деление двух целых чисел в дополнительном коде

Введем обозначения операндов, используемых в данной операции (таблица 5):

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

Теперь приведем блок-схему работу автомата, реализующий данный алгоритм, с указанием микрокоманд. См. рис. 8 в Приложении А. Ис-

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

Таблица 5: Операнды

пользуем счетчик для подсчета обработанных разрядов и регистры для хранения и использования разрядов делителя и делимого.

7.2 Сложение чисел в экспоненциальной форме

Опишем алгоритм выполнения суммы с помощью блок схемы. См. рис. 9. в Приложении А.

Теперь приведем блок-схему работу автомата, реализующий данный алгоритм, с указанием микрокоманд. См. рис. 10 в Приложении А.

8 Описание микрокоманд

8.1 Деление двух целых чисел в дополнительном коде

Укажем необходимые признаки, которые впоследствии будут вырабатываться управляющим автоматом в ходе выполнения первой операции. См. таблицу 6.

Признак	Назначение
S	Хранит адрес следующей операции
H	Адресный вход мультиплексора
$R0$	Сигнализирует об окончании операции деления
$ERROR$	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
$COUNT\_CT$	Счет. Декремент счетчика, если $L\_CT == 1$
L_CT	Загрузка счетчика
$SHIFT$	Левый сдвиг в регистрах RG_A и RG_RES

Таблица 6: Деление чисел. Осведомительные сигналы (признаки)

8.2 Сложение чисел в экспоненциальной форме

Укажем необходимые признаки, которые впоследствии будут вырабатываться управляющим автоматом в ходе выполнения второй операции. См. таблицу 7.

Признак	**Назначение**
S	Хранит адрес следующей операции
H	Адресный вход мультиплексора
$R0$	Сигнализирует об окончании операции деления
$overflow$	Сигнализирует об ошибке обработки – переполнение
L_Ma	Загрузка в регистр RG_Ma
$SHIFT\_Ma$	Правый сдвиг регистра RG_Ma если $SHIFT\_Ma\_Left = 0$ и левый, если $SHIFT\_Ma\_Left = 1$
RST	Асинхронный сброс всех элементов
$COUNT\_Pa$	Счет. Декремент счетчика, если $L\_CT\_Pa == 1$
L_CT_Pa	Загрузка счетчика CT_Pa
$CHANGE$	Выбор источника загрузки в регистры мантисс и порядка чисел A и B
e	Управляющий сигнал для счетчика. Если $e = 1$, следует выполнить загрузку, а если $e = 0$ – инкрементировать счетчик.

Таблица 7: Сложение чисел в экспоненциальной форме. Осведомительные сигналы (признаки)

9 Функциональная схема операционного автомата

9.1 Деление двух целых чисел в дополнительном коде

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

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

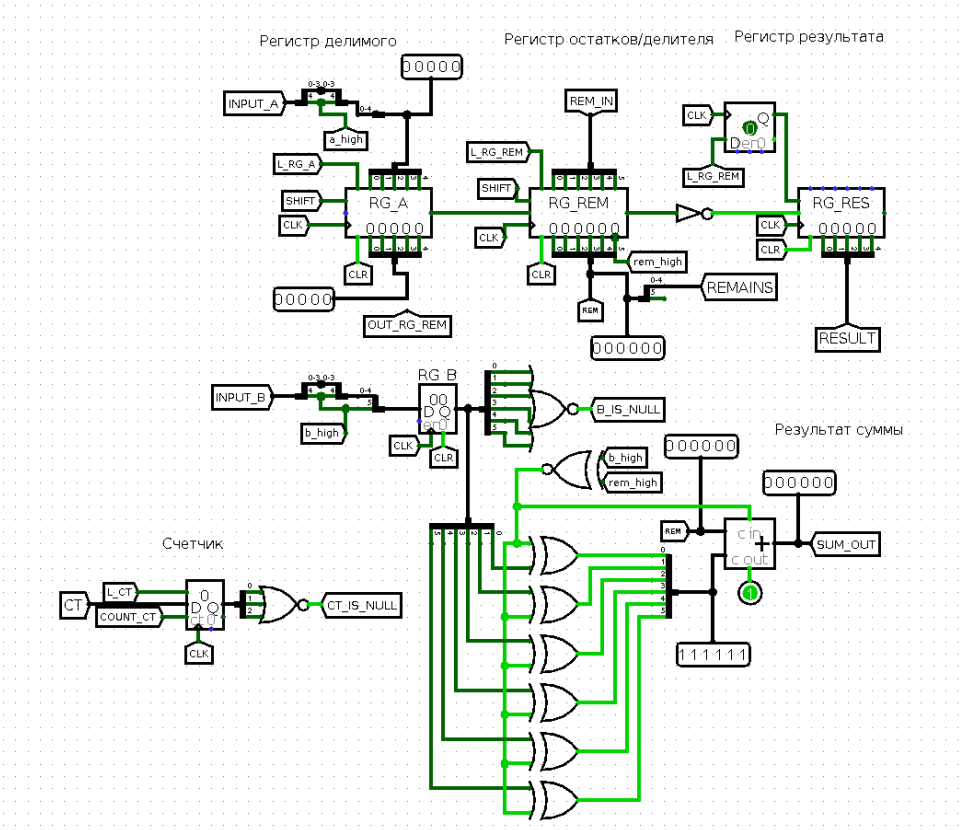

Рис. 4: Деление чисел. Операционный автомат

Идентификатор	**Назначение**
RG_A	Сдвиговый регистр. Хранит разряды делимого
RG_B	Хранит разряды делителя
RG_REM	Сдвиговый регистр. Хранит разряды частичного остатка
RG_RES	Сдвиговый регистр. Хранит разряды результата

Таблица 8: Регистры операционного автомата

9.2 Сложение чисел в экспоненциальной форме

Приведем функциональную схему операционного автомата, выполняющего сложение двух чисел, представленных в экспоненциальном формате. См. рис. 11.

Приведем названия и назначения каждого из регистров, используемых в данном устройстве. См. таблицу 9.

Идентификатор	**Назначение**
RG_Ma	Универсальный сдвиговый регистр. Хранит разряды мантиссы A
CT_Mb	Счетчик. Хранит разряды мантиссы B
CT_Pa	Счетчик. Хранит разряды порядка числа A
CT_Pb	Счетчик. Хранит разряды порядка числа B
CT_dP	Счетчик. Хранит разряды разницы порядков чисел A и B
REG_SUM	Триггер. Хранит разряд сигнала переноса суммы мантисс чисел A и B

Таблица 9: Регистры операционного автомата

10 Типовые примеры

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

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 10: Пример деления целых чисел в доп. коде

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

11 Управляющий автомат

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

автомата. См рисунок 5.

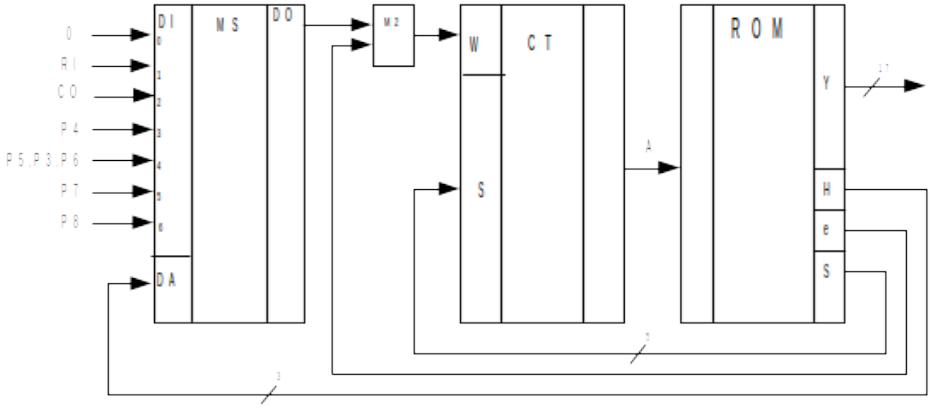

Рис. 5: УА с регулярной адресацией

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 М2 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.

12 Заполнение памяти

12.1 Деление двух целых чисел в дополнительном коде

Приведем таблицу переходов для управляющего автомата выполненного по схеме с регулярной адресацией для операции нахождения частного двух чисел в дополнительном коде. См. таблицу 11.

A	Y	H	S	e
0	m0	0	1	1
1	m1	0	4	1
2	m01	0	2	x
3	m2	2	0	0
4	m3	0	4	0
5	m02	3	3	x
6	m6	0	0	0
7	m5	0	3	0
8	m4	2	2	x
9	m03	1	1	x

Таблица 11: Деление чисел в доп. коде. Таблица переходов

Также приведем таблицу заполнения памяти ПЗУ управляющего автомата. См. таблицу 14 в Приложении А.

12.2 Сложение чисел в экспоненциальном формате

Приведем таблицу переходов для управляющего автомата выполненного по схеме с регулярной адресацией для операции нахождения суммы двух чисел в экспоненциальной форме. См. таблицу 12.

A	Y	H	S	e
0	m1	x	1	x
1	m2	2	15	0
2	m01	1	16	0
3	m02	3	5	1
4	m6	x	5	x
5	m03	5	9	0
6	m04	4	18	0
7	m8	0	19	1
8	m9	5	19	1
9	m10	8	11	1
10	m11	6	13	0
11	m05	9	14	0

См. след. страницу

Таблица 12 – продолжение

A	Y	H	S	e
12	m12	7	11	1
13	m13	0	0	1
14	m14	0	0	1
15	m3	0	0	1
16	m4	x	17	x
17	m5	0	0	1
18	m7	0	0	1
19	m15	0	8	1

Таблица 12: Сложение чисел в экспоненциальной форме. Таблица переходов

Также приведем таблицу заполнения памяти ПЗУ управляющего автомата. См. таблицу 14 в Приложении А.

13 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.

Освоенные в течение курса теоретические положения были применены на практике, что позволило использовать аппарат теории автоматов для решения реальных задач проектирования дискретных устройств с памятью.

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 З-17
5. Горбатов В.А. Теория автоматов: учеб. для студентов втузов – М.: АСТ: Астрель. 2008. – 559 с.
6. Карпов Ю.Г. Теория автоматов /– СПб: Питер. 2003. – 208 с.

ПРИЛОЖЕНИЕ А

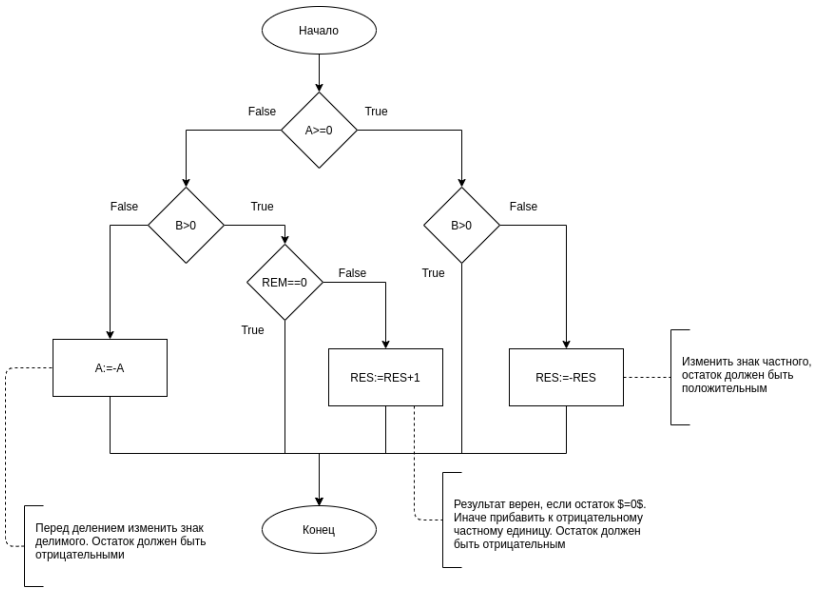

Рис. 6: Алгоритм коррекции результата

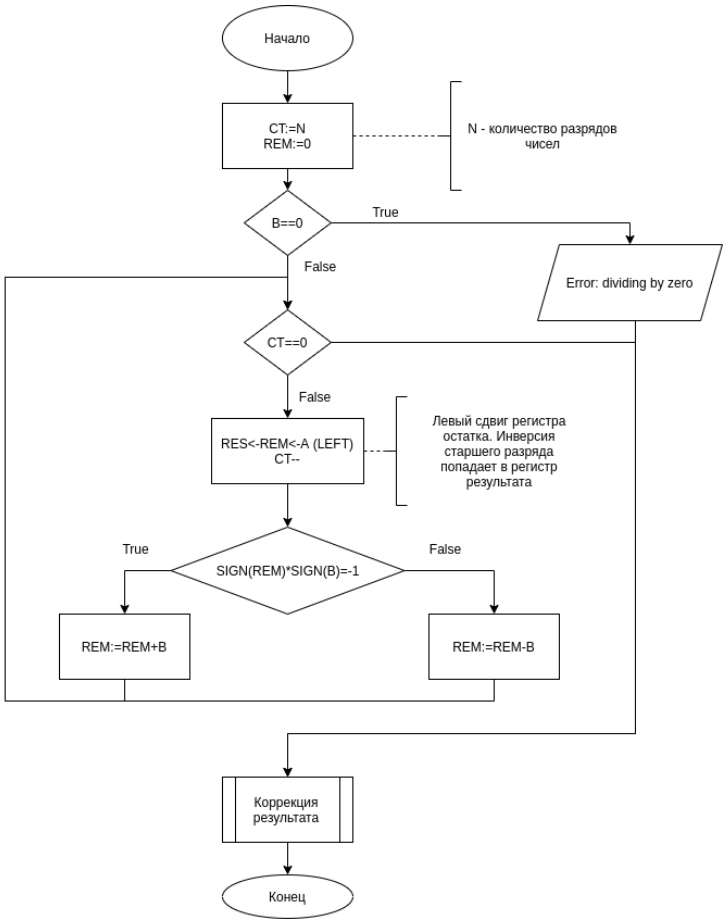

Рис. 7: Алгоритм деления чисел

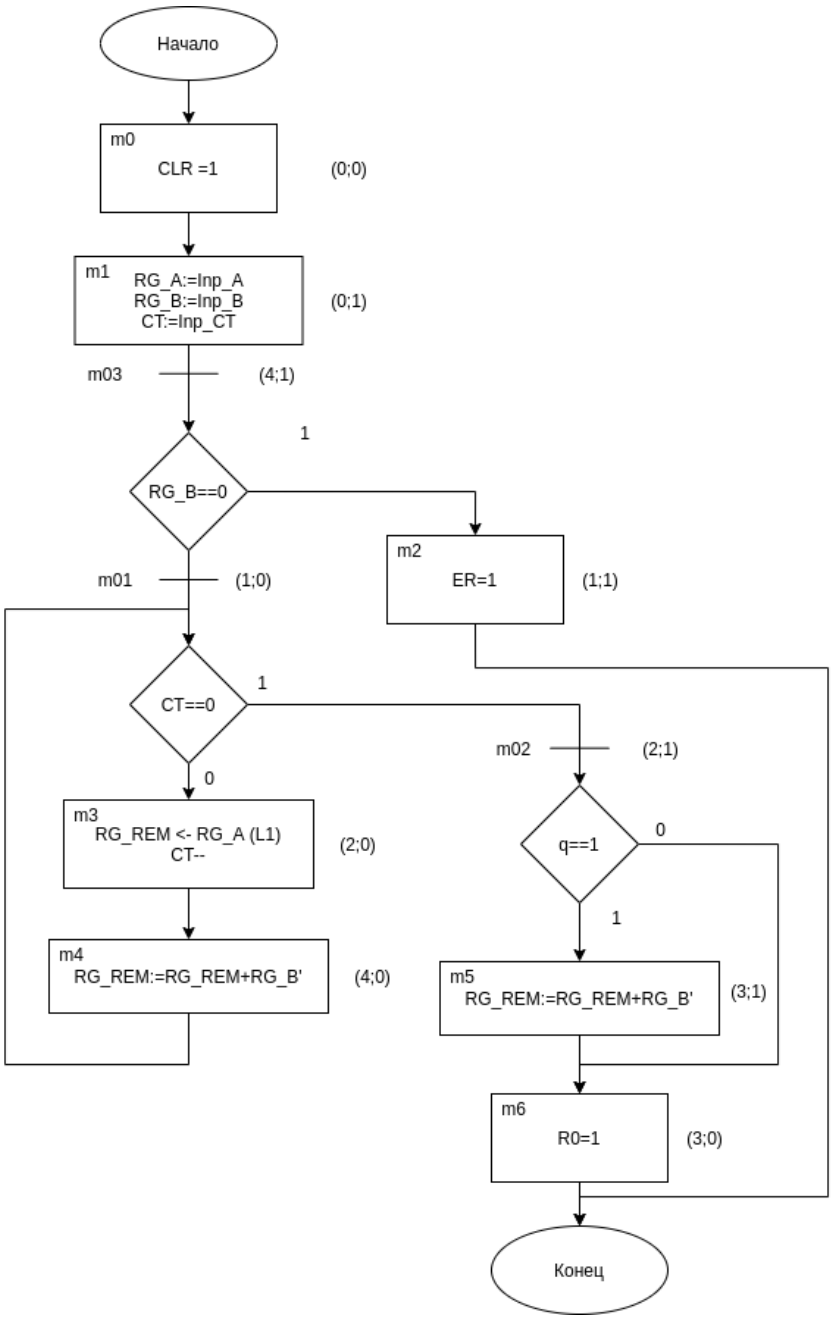

Рис. 8: Деление двух чисел. Блок-схема работы автомата

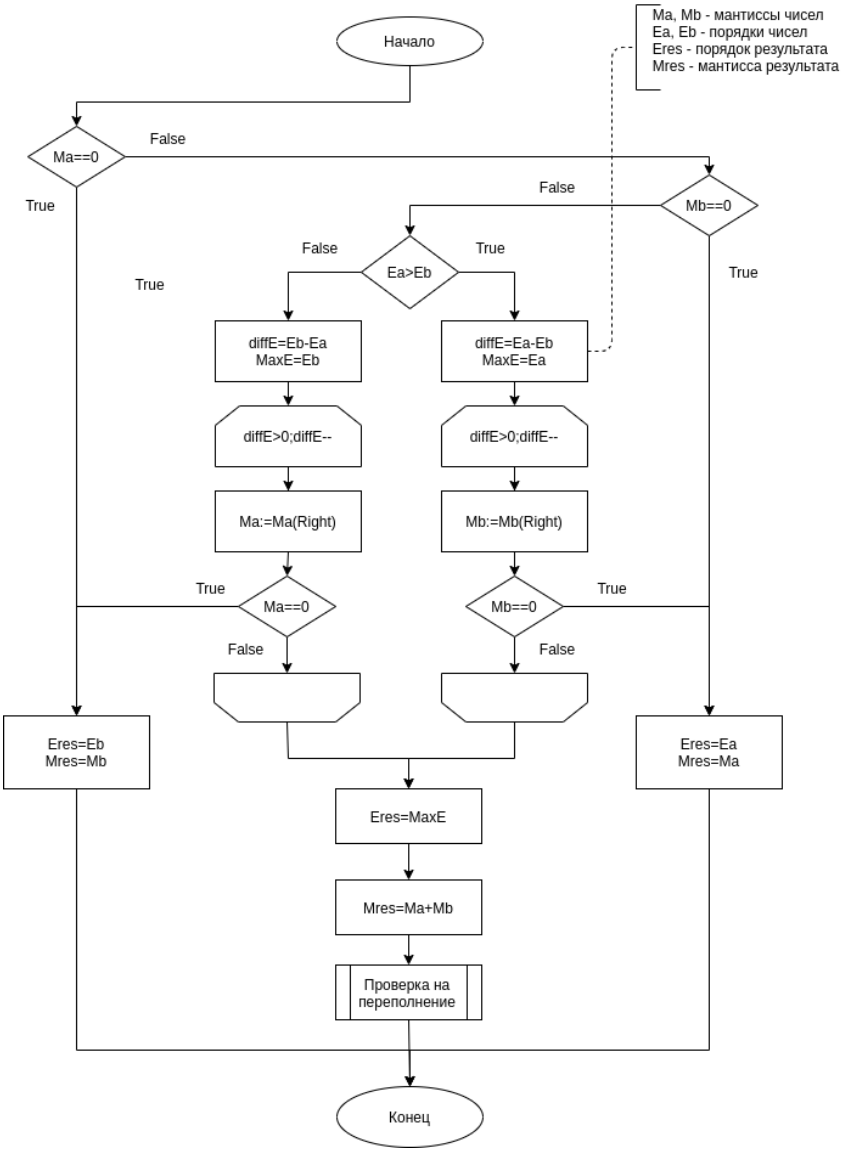

Рис. 9: Алгоритм суммирования чисел в доп. коде

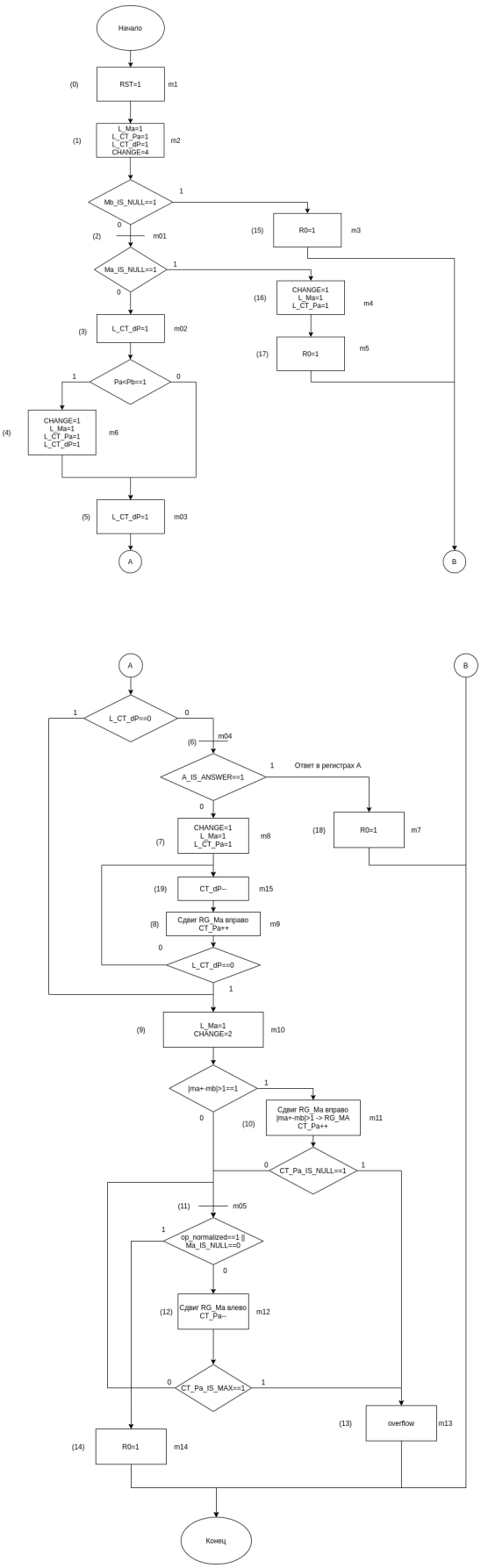

Рис. 10: Алгоритм сложения чисел. Блок-схема работы автомата

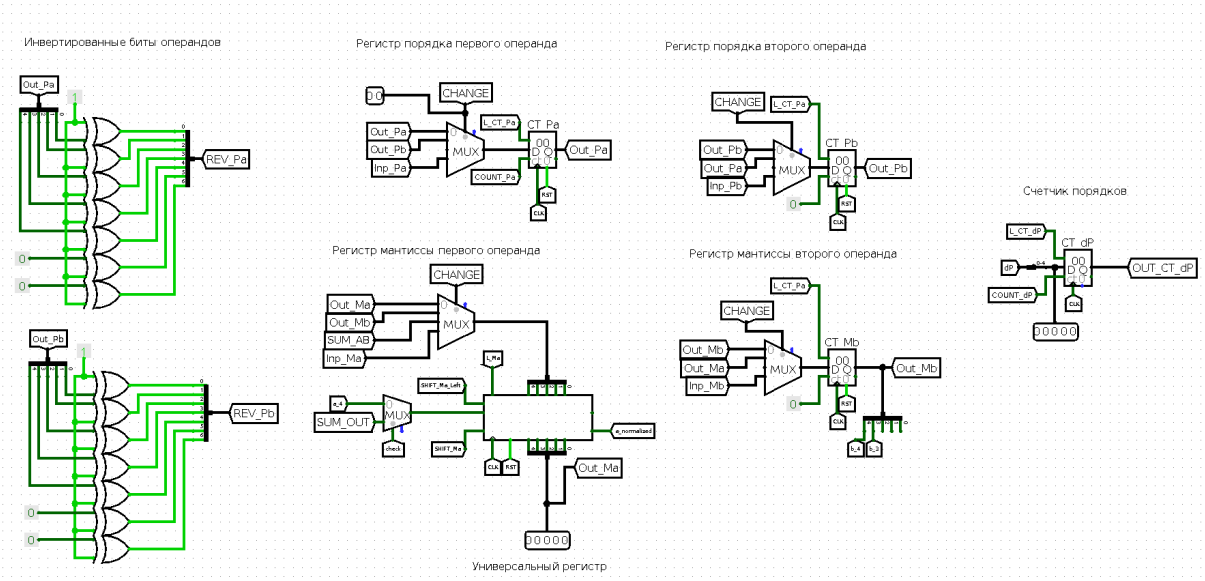

Рис. 11: Сложение чисел. Операционный автомат

A	N	e	L_RG_A	L_B	L_REM	SHIFT	CT	L_CT	CLR	ER	R0	H	S
0	m0	1	0	0	0	0	0	0	1	0	0	00	001
1	m1	1	1	1	1	0	0	1	0	0	0	00	100
2	m01	x	0	0	0	0	0	0	0	0	0	10	010
3	m2	0	0	0	0	0	0	0	0	1	0	00	000
4	m3	0	0	0	0	1	1	1	0	0	0	11	100
5	m02	x	0	0	0	0	0	0	0	0	0	00	011
6	m6	0	0	0	0	0	0	0	0	0	1	00	000
7	m5	0	0	0	1	0	0	0	0	0	0	00	011
8	m4	x	0	0	1	0	0	0	0	0	0	10	010
9	m03	x	0	0	0	0	0	0	0	0	0	01	001

Таблица 13: Деление чисел в доп. коде. Таблица заполнения памяти

A	N	S	R0	RST	L_ma	SFT_Ma	SFT_Ma_L	L_CT_Pa	COUNT_Pa	CHANGE	L_CT_dP	COUNT_dP	m_n	H	e
0	m1	1	0	1	0	0	0	0	0	00	0	0	0	0	0
1	m2	15	0	0	1	0	0	1	0	11	1	0	0	2	0
2	m01	16	0	0	0	0	0	0	0	00	0	0	0	1	0
3	m02	5	0	0	0	0	0	0	0	00	1	0	0	3	1
4	m6	5	0	0	1	0	0	1	0	01	1	0	0	0	0
5	m03	9	0	0	0	0	0	0	0	00	1	0	0	5	0
6	m04	18	0	0	0	0	0	0	0	00	0	0	0	4	0
7	m8	19	0	0	1	0	0	1	0	01	0	0	0	0	1
8	m9	19	0	0	0	1	0	0	1	00	0	0	0	5	1
9	m10	11	0	0	1	0	0	0	0	10	0	0	0	8	1
10	m11	13	0	0	0	1	0	0	1	00	0	0	0	6	0
11	m05	14	0	0	0	0	0	0	0	00	0	0	0	9	0
12	m12	11	0	0	0	1	1	1	1	00	0	0	0	7	1
13	m13	0	0	0	0	0	0	0	0	00	0	0	1	0	1
14	m14	0	1	0	0	0	0	0	0	00	0	0	0	0	1
15	m3	0	1	0	0	0	0	0	0	00	0	0	0	0	1
16	m4	17	0	0	1	0	0	1	0	01	0	0	0	0	0
17	m5	0	1	0	0	0	0	0	0	00	0	0	0	0	1
18	m7	0	1	0	0	0	0	0	0	00	0	0	0	0	1
19	m15	8	0	0	0	0	0	0	0	00	1	1	0	0	1

Таблица 14: Сложение чисел в экспоненциальной форме. Таблица заполнения памяти