



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий
Кафедра Вычислительной техники

Отчет по практическим работам

по дисциплине
«Теория автоматов»

Выполнил студент группы ИВБО-02-19

К. Ю. Денисов

Проверил ассистент

А. С. Боронников

Москва 2021

Содержание

1 Практическая работа №1.

Умножитель 4-разрядных чисел без знака	4
1.1 Общее строение автомата	4
1.2 Алгоритм работы автомата	4
1.3 Реализация Операционного автомата	5
1.4 Реализация управляющего автомата	6
1.5 Тестирование работы автомата	7
1.6 Вывод	8

2 Практическая работа №2.

Умножитель 4-разрядных чисел в дополнительном коде	9
2.1 Общее строение автомата	9
2.2 Алгоритм работы автомата	9
2.3 Реализация Операционного автомата	10
2.4 Реализация управляющего автомата	12
2.5 Тестирование работы автомата	13
2.6 Вывод	14

3 Практическая работа №3.

Делитель 4-разрядных чисел без знака	15
3.1 Общее строение автомата	15
3.2 Алгоритм работы автомата	15
3.3 Реализация Операционного автомата	16
3.4 Реализация управляющего автомата	17
3.5 Тестирование работы автомата	18
3.6 Вывод	19

4 Практическая работа №4.

Делитель 4-разрядных чисел в дополнительном коде	20
4.1 Общее строение автомата	20
4.2 Алгоритм работы автомата	20
4.3 Реализация Операционного автомата	21
4.4 Реализация управляющего автомата	22

4.5	Тестирование работы автомата	23
4.6	Вывод	25
5	Практическая работа №5.	
	Сложение чисел с плавающей точкой	26
5.1	Общее строение автомата	26
5.2	Алгоритм работы автомата	27
5.3	Реализация Операционного автомата	27
5.4	Реализация управляющего автомата	29
5.5	Вывод	30
	ПРИЛОЖЕНИЕ А	31

1 Практическая работа №1.

Умножитель 4-разрядных чисел без знака

1.1 Общее строение автомата

В ходе данной практической работы был реализован автомат, выполняющий умножение 4-разрядных чисел без знака. Управляющий автомат был построен по схеме с адресным ПЗУ в последовательном варианте. Рассмотрим строение управляющего автомата. См рисунок 1.

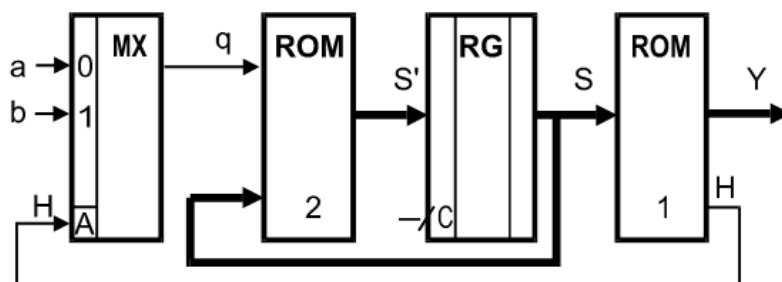


Рис. 1: УА с адресным ПЗУ; последовательный вариант

В конкретной реализации роль мультиплексора выполняет логический элемент ИЛИ, на входы которого подаются сигналы $CT0$ — признак нуля в счетчике и осведомительный сигнал (признак) H — указывающий на присутствие логического ветвления в текущем месте алгоритма.

1.2 Алгоритм работы автомата

Опишем алгоритм работы автомата с помощью блок схемы. Используем сумматор для сложения текущего значения СЧП и множимого, счетчик для подсчета обработанных разрядов и регистры для хранения и использования разрядов рассматриваемых чисел. Обозначим микрокоманды от m_0 до m_4 . См. рисунок 2.

В алгоритме присутствует условие, это означает, что при реализации операционного автомата текущее значение счетчика необходимо проверять при переходе $m_0 \rightarrow m_1$ и $m_3 \rightarrow m_1$.

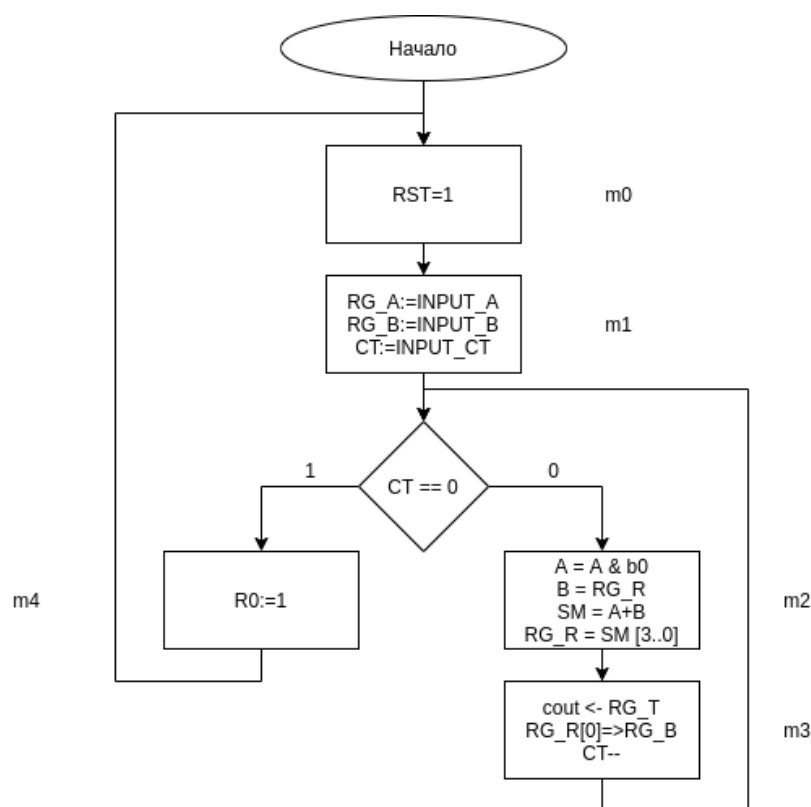


Рис. 2: Алгоритм умножения двух 4-разрядных чисел

После построения алгоритма работы автомата следует перейти к реализации операционной части.

1.3 Реализация Операционного автомата

Построим операционный автомат, выполняющий умножение двух 4-разрядных чисел посредством использования четырех регистров, в том числе двух сдвиговых. Приведем названия и назначения каждого из регистров. См. таблицу 1.

Укажем необходимые признаки, которые впоследствии будут вырабатываться управляющим автоматом. См. таблицу 2. Соединим все элементы в соответствии с алгоритмом задачи. См. рисунок 3.

Идентификатор	Назначение
<i>RG_A</i>	Хранит разряды множимого
<i>RG_B</i>	Сдвиговый регистр. Хранит разряды множителя
<i>RG_R</i>	Сдвиговый регистр. Хранит разряды СЧП, служит для хранения старших разрядов результата
<i>RG_D</i>	Сдвиговый регистр. Хранит младшие разряды результата

Таблица 1: Регистры операционного автомата

Признак	Назначение
Н	Указывает на условность–безусловность перехода
<i>EMIT_R0</i>	Сигнализирует об окончании операции умножения
<i>LOAD_R</i>	Загрузка в регистр <i>RG_R</i>
<i>RST</i>	Асинхронный сброс всех элементов
<i>COUNT_CT</i>	Загрузка счетчика. Декремент, если <i>DECR_CT</i> == 1
<i>DECR_CT</i>	Декремент счетчика
<i>LOAD_AB</i>	Загрузка в регистры <i>RG_A</i> и <i>RG_B</i>
<i>SHIFT_RB</i>	Сдвиг в регистрах <i>RG_R</i> и <i>RG_B</i>

Таблица 2: Осведомительные сигналы (признаки)

1.4 Реализация управляющего автомата

Приступим к построению управляющего автомата, определяющего последовательность выполнения микрокоманд для умножения двух 4-разрядных чисел.

Определим разрядность адресного ПЗУ и ПЗУ микрокоманд. Адрес должен иметь 4 разряда, где ведущим разряд — текущее значение параметра *CT0*. Микрокоманда представлена в виде 8 бит — 8 признаков, расположенных в следующем порядке: Н, *EMIT_R0*, *LOAD_R*, *RST*, *CONT_CT*, *DECR_CT*, *LOAD_AB*, *SHIFT_RB*. Адрес текущей команды будет храниться в 4-разрядном регистре.

Заполним память в соответствии в алгоритмом, подключим ПЗУ и регистр последовательным способом. См рисунок 4.

двоичных числа $1111_2 * 1111_2$ ожидая получить двоичное число 11100001_2 . Укажем входные данные, будем подавать тактовые сигналы до тех пор, пока индикатор не сообщит нам о завершении операции, сравним практические результаты с ожидаемыми. См рисунок 5. Умножение выполнено корректно. Ожидаемые и полученные результаты совпадают.

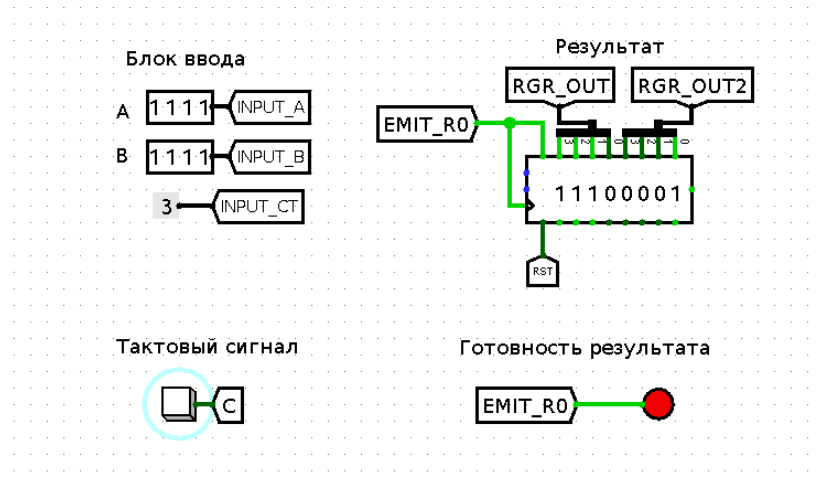


Рис. 5: Схема операционного автомата

1.6 Вывод

В ходе данной практической работы было рассмотрено строение и работа управляющего автомата с адресным ПЗУ. Используя полученные знания на практике, на основе данного управляющего автомата построено вычислительное устройство (операционный и управляющий автомат), реализующее операцию умножения двух 4-разрядных чисел без знака. Работа данного устройства испытана, проверена корректность полученных результатов.

2 Практическая работа №2.

Умножитель 4-разрядных чисел в дополнительном коде

2.1 Общее строение автомата

В ходе данной практической работы был реализован автомат, выполняющий умножение 4-разрядных чисел в дополнительном коде. Управляющий автомат был построен по схеме с двумя адресами в памяти в последовательном варианте. Рассмотрим строение управляющего автомата. См рисунок 6.

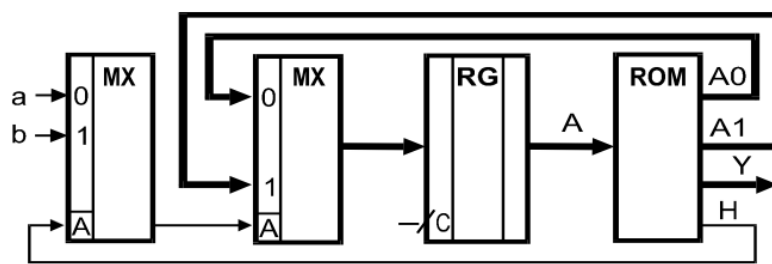


Рис. 6: УА с двумя адресами в памяти; последовательный вариант

В конкретной реализации на адресный вход мультиплексора подается сигнал $CT0$ — признак нуля в счетчике. Основываясь на значении данного сигнала выбирается один из двух альтернативных адресов последующих состояний автомата.

2.2 Алгоритм работы автомата

Опишем алгоритм работы автомата с помощью блок схемы. Используем сумматор для сложения текущего значения СЧП и множимого, счетчик для подсчета обработанных разрядов и регистры для хранения и использования разрядов рассматриваемых чисел. Обозначим микрокоманды от m_0 до m_4 . См. рисунок 7.

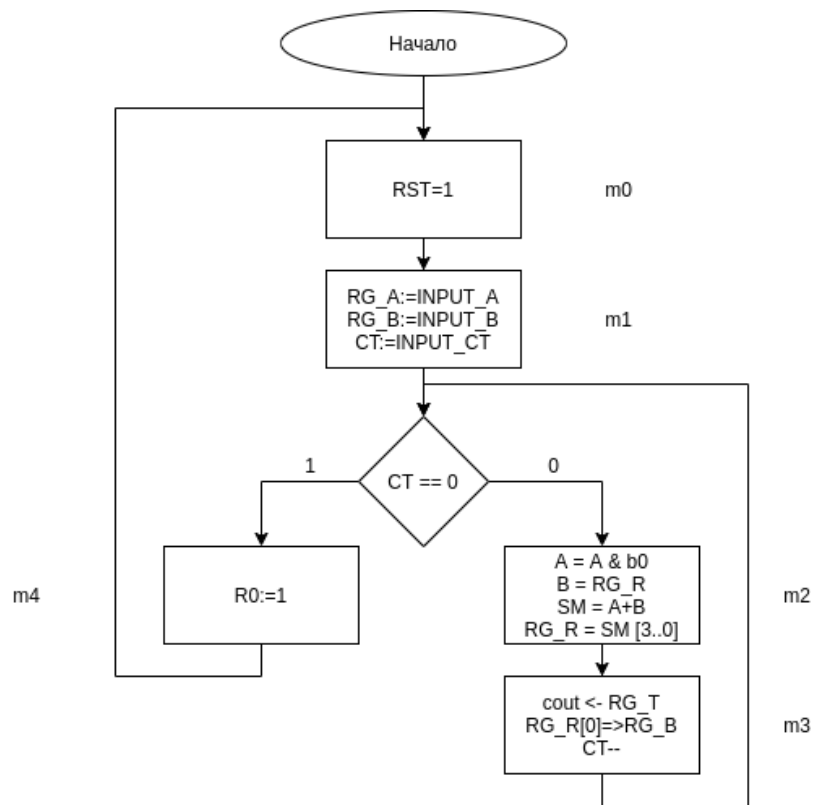


Рис. 7: Алгоритм умножения двух 4-разрядных чисел

В алгоритме присутствует условие, это означает, что при реализации операционного автомата текущие значение счетчика необходимо проверять при переходе $m_0 \rightarrow m_1$ и $m_3 \rightarrow m_1$.

После построения алгоритма работы автомата следует перейти к реализации операционной части.

2.3 Реализация Операционного автомата

Построим операционный автомат, выполняющий умножение двух 4-разрядных чисел посредством использования четырех регистров, в том числе двух сдвиговых. Приведем названия и назначения каждого из регистров. См. таблицу 3.

Укажем необходимые признаки, которые впоследствии будут вырабатываться управляющим автоматом. См. таблицу 4.

При умножении по данному алгоритму следует обратить внимание на необходимость использования модифицированного дополнительного кода для множимого, поскольку при получении частичных произве-

Идентификатор	Назначение
RG_A	Хранит разряды множимого
RG_B	Сдвиговый регистр. Хранит разряды множителя
RG_R	Сдвиговый регистр. Хранит разряды СЧП, служит для хранения старших разрядов результата

Таблица 3: Регистры операционного автомата

Признак	Назначение
$EMIT_R_0$	Сигнализирует об окончании операции умножения
$LOAD_R$	Загрузка в регистр RG_R
RST	Асинхронный сброс всех элементов
$COUNT_CT$	Загрузка счетчика. Декремент, если $DECR_CT == 1$
$DECR_CT$	Декремент счетчика
$LOAD_AB$	Загрузка в регистры RG_A и RG_B
$SHIFT_RB$	Сдвиг в регистрах RG_R и RG_B

Таблица 4: Осведомительные сигналы (признаки)

дений возможно временное переполнение разрядной сетки. Использование модифицированного дополнительного кода позволяет его зафиксировать без потери знака. Это переполнение устраняется последующим сдвигом частичного произведения вправо.

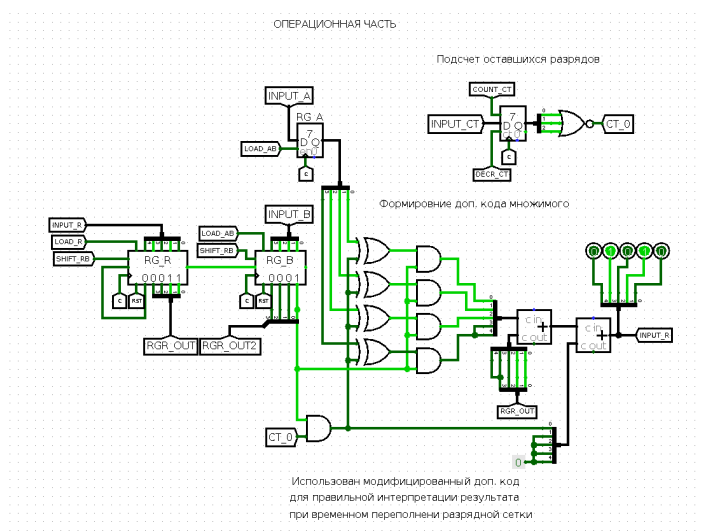


Рис. 8: Схема операционного автомата

Соединим все элементы в соответствии с алгоритмом задачи. См. рисунок 8.

2.4 Реализация управляющего автомата

Приступим к построению управляющего автомата, определяющего последовательность выполнения микрокоманд для умножения двух 4-разрядных чисел в дополнительном коде.

Определим разрядность ПЗУ, в котором будут содержаться альтернативные адреса переходов. Микроинструкция представлена в виде 13 разрядов, где 6 разрядов занимают два альтернативных адреса переходов, а оставшиеся 7 — признаки, расположенные в следующем порядке: RST, SHIFT_RB, LOAD_AB, LOAD_R, DECR_CT, COUNT_CT, EMIT_R0. Адрес текущей команды будет храниться в 3-разрядном регистре.

Альтернативные адреса будут подаваться на вход мультиплексора, управляемого сигналом *CT0*, затем, выбранный адрес будет загружен в регистр текущего состояния, выход которого подключен к постоянному запоминающему устройству.

Заполним память в соответствии с алгоритмом, подключим ПЗУ и регистр последовательным способом. См рисунок 9.

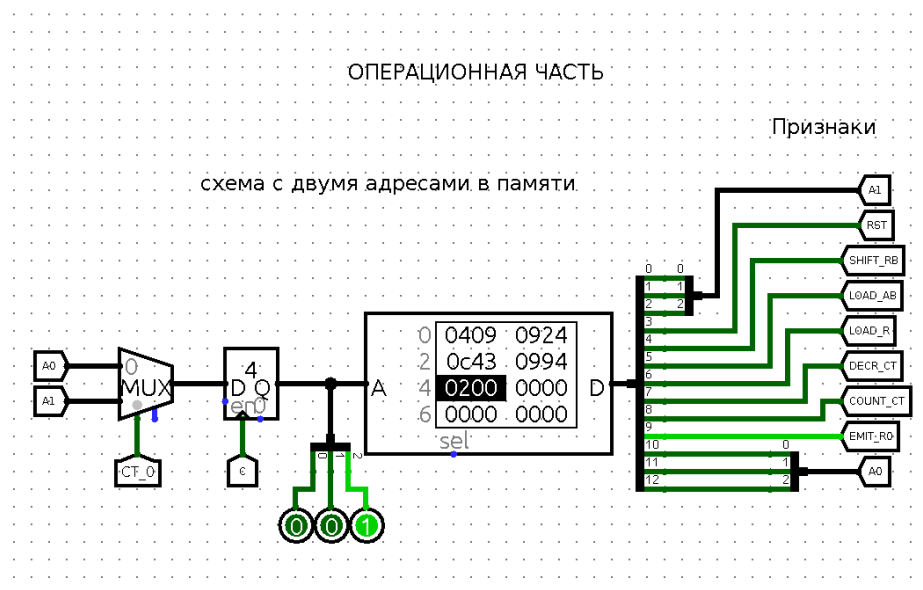


Рис. 9: Схема операционного автомата

2.5 Тестирование работы автомата

После реализации операционного и управляющего автомата следует приступить к объединению данных устройств, тестированию их совместной работы. Подключим признаки к входам соответствующих логических элементов и цифровых устройств с помощью туннелей. Добавим блок ввода исходных данных, используя контакты, блок вывода — регистр результата и индикатор завершения операции умножения.

Проведем проверку корректности выходных результатов построенного цифрового устройства. Перемножим два 4-разрядных двоичных числа $1111_{\text{доп}_2} * 1111_{\text{доп}_2}$, которые в данном случае интерпретируются как дополнительный код отрицательного десятичного числа -1_{10} ожидая получить результат $-1_{10} = 00000001_{\text{доп}_2}$. Укажем входные данные, будем подавать тактовые сигналы до тех пор, пока индикатор не сообщит нам о завершении операции, сравним практические результаты с ожидаемыми. См рисунок 10. Умножение выполнено корректно. Ожидаемые и полученные результаты совпадают.

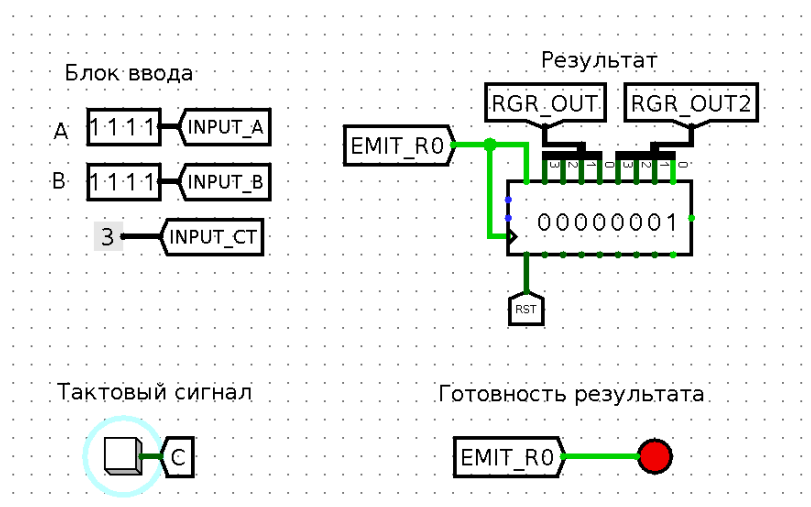


Рис. 10: Схема операционного автомата

Теперь перемножим положительное число $7_{10} = 0111_{\text{доп}_2}$ и отрицательное число $-2_{10} = 1110_{\text{доп}_2}$, ожидая получить результат $-14_{10} = 11110010_{\text{доп}_2}$. Введем исходные данные и сравним результаты вычислений. См. рисунок 11. Как видно на рисунке, вычисления привели к верному ответу. С помощью данного автомата можно перемножать числа в диапазоне $(-7; 7)$.

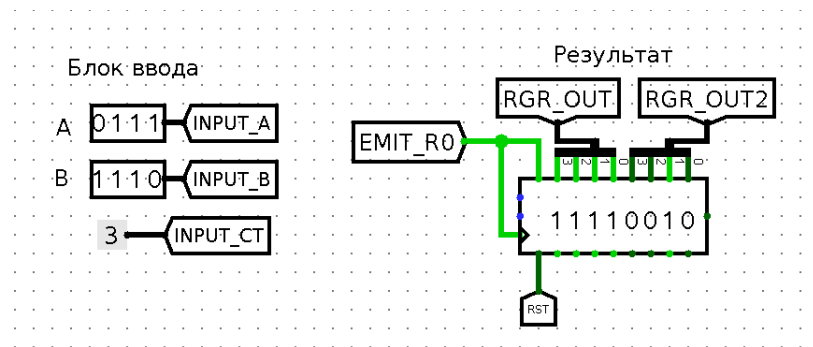


Рис. 11: Схема операционного автомата

2.6 Вывод

В ходе данной практической работы было рассмотрено строение и работа последовательного варианта управляющего автомата с двумя адресами в памяти. Используя полученные знания на практике, на основе данного управляющего автомата построено вычислительное устройство (операционный и управляющий автомат), реализующее операцию умножения двух 4-разрядных чисел в дополнительном коде. Работа данного устройства испытана, проверена корректность полученных результатов.

После построения алгоритма работы автомата следует перейти к реализации операционной части.

3.3 Реализация Операционного автомата

Построим операционный автомат, выполняющий деление двух 4-разрядных чисел посредством использования четырех регистров, в том числе трех сдвиговых. Приведем названия и назначения каждого из регистров. См. таблицу 5.

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 5: Регистры операционного автомата

Укажем необходимые признаки, которые впоследствии будут вырабатываться управляющим автоматом. См. таблицу 6. С целью реализации левого сдвига в сдвиговых регистрах разряды делимого и текущего значения частичного остатка загружаются в регистры в обратном порядке. Это позволяет отказаться от универсального сдвигового регистра, так как реализации данного алгоритма необходим только левый сдвиг, и как следствие, упрощает схему.

Для исключения возникновения ошибки при левом сдвиге разрядов частичного остатка в случае, когда два его старших разряда различны, использованы 6-разрядные регистры. См. сноску 1.

Соединим все элементы в соответствии с алгоритмом задачи. См. рисунок 21 в приложении **Приложении А**.

$$\begin{aligned}
 -|B| &\neq 10X...X \\
 -|B| &= 110X...X
 \end{aligned}
 \tag{1}$$

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: Осведомительные сигналы (признаки)

3.4 Реализация управляющего автомата

Приступим к построению управляющего автомата, определяющего последовательность выполнения микрокоманд для деления двух 4-разрядных чисел без знака.

Определим разрядность ПЗУ, участвующего в построении УА по схеме с одним адресом в памяти. Адрес должен иметь 4 разряда, где 3 старших разряда — текущее значение параметра S , а младший разряд — значение q , генерируемое мультиплексором. Микрокоманда представлена в виде 15 бит — 12 признаков, расположенных в следующем порядке: $S, H, R0, ERROR, CLR, L_CT, COUNT_CT, SHIFT, L_RG_REM, L_RG_B, L_RG_A, e$. Адрес текущей команды будет храниться в 4-разрядном регистре.

Заполним память в соответствии в алгоритмом, подключим ПЗУ и регистр последовательным способом. См рисунок 13.

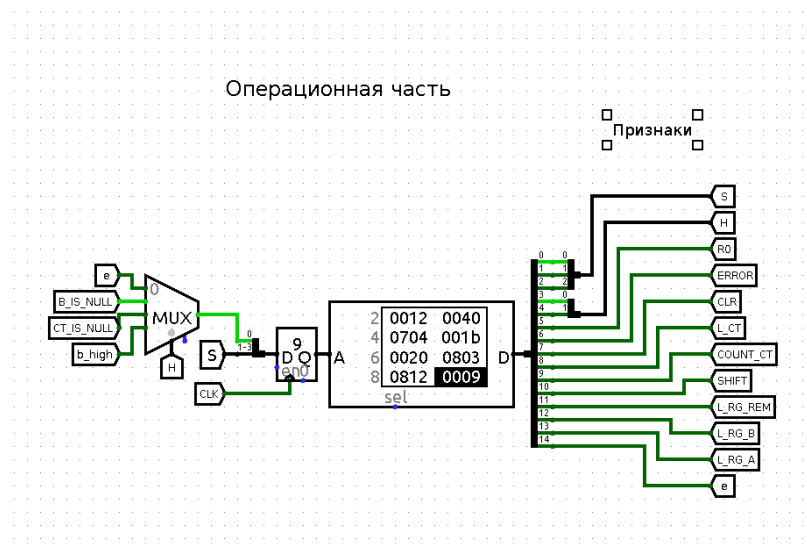


Рис. 13: Схема операционного автомата

3.5 Тестирование работы автомата

После реализации операционного и управляющего автомата следует приступить к объединению данных устройств, тестированию их совместной работы. Подключим признаки к входам соответствующих логических элементов и цифровых устройств с помощью туннелей. Добавим блок ввода исходных данных, используя контакты, блок вывода — регистр результата операции деления и регистр остатка, индикатор завершения операции деления.

Проведем проверку корректности выходных результатов построенного цифрового устройства. Разделим два наибольших 4-разрядных двоичных числа $1111_2 \div 1111_2$ ожидая получить частное 1_2 и остаток 0_2 . Укажем входные данные, будем подавать тактовые сигналы до тех пор, пока индикатор не сообщит нам о завершении операции, сравним практические результаты с ожидаемыми. См рисунок 14. Умножение выполнено корректно. Ожидаемые и полученные результаты совпадают.

Протестируем работу автомата на входных данных, которые гипотетически могут привести к ошибке — поделим число 101_2 на число 1111_2 . Ошибка состоит в том, что в случае использования регистров недостаточной разрядности, левый сдвиг при вычислении очередного остатка произойдет некорректно, что приведет к неверному результа-

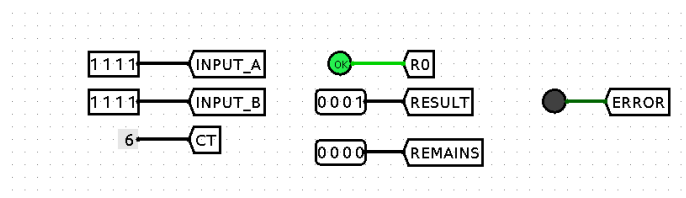


Рис. 14: Проверка работы автомата

ту. Введем данные, наблюдаем за работой автомата. См рисунок 15. Операция выполнена верно. Использование регистров большей разрядности исключило возможность ошибки.

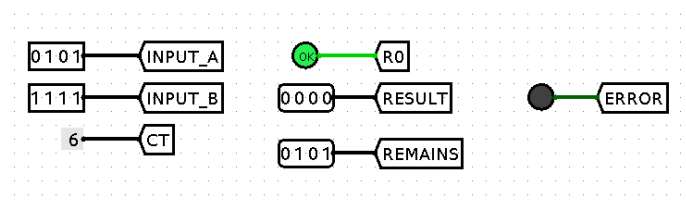


Рис. 15: Вторая проверка работы автомата

3.6 Вывод

В ходе данной практической работы было рассмотрено строение и работа управляющего автомата, построенного по схеме с одним адресом в памяти. Используя полученные знания на практике, на основе данного управляющего автомата построено вычислительное устройство (операционный и управляющий автомат), реализующее операцию деления двух 4-разрядных чисел без знака.

Работа данного устройства испытана, проверена корректность полученных результатов при работе со всеми категориями входных данных.

4 Практическая работа №4.

Делитель 4-разрядных чисел в дополнительном коде

4.1 Общее строение автомата

В ходе данной практической работы был реализован автомат, выполняющий деление 4-разрядных чисел в дополнительном коде (алгоритм без восстановления остатка). Управляющий автомат был построен по схеме с сокращенным тактом. Рассмотрим строение управляющего автомата. См рисунок 16.

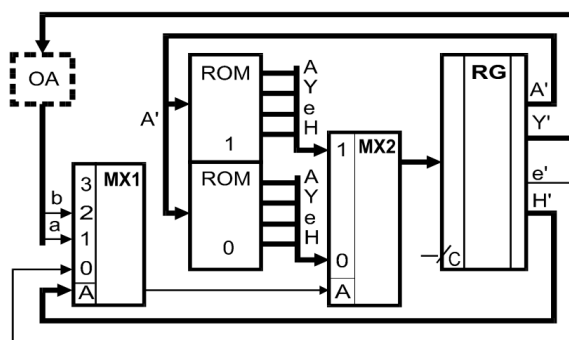


Рис. 16: УА с сокращенным тактом

В конкретной реализации на информационные входы мультиплексора подаются сигналы e – сигнал с ПЗУ, B_IS_NULL – признак нулевого делителя, CT_IS_NULL – признак окончания счета, b_high – значение старшего бита частичного остатка на текущей итерации, а на адресный вход подается двухбитовый сигнал H .

4.2 Алгоритм работы автомата

Опишем алгоритм работы автомата с помощью блок схемы. Используем сумматор для нахождения текущего значения частичного остатка (ЧО), счетчик для подсчета обработанных разрядов и регистры для хранения и использования разрядов делителя и делимого. Обозначим микрокоманды от m_0 до m_4 . См. рисунок 22 в Приложении А.

После построения алгоритма работы автомата следует перейти к реализации операционной части.

4.3 Реализация Операционного автомата

Построим операционный автомат, выполняющий деление двух 4-разрядных чисел в дополнительном коде посредством использования четырех регистров, в том числе трех сдвиговых. Приведем названия и назначения каждого из регистров. См. таблицу 7.

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 7: Регистры операционного автомата

Укажем необходимые признаки, которые впоследствии будут вырабатываться управляющим автоматом. См. таблицу 8. С целью реализации левого сдвига в сдвиговых регистрах разряды делимого и текущего значения частичного остатка загружаются в регистры в обратном порядке. Это позволяет отказаться от универсального сдвигового регистра, так как реализации данного алгоритма необходим только левый сдвиг, и как следствие, упрощает схему.

Для исключения возникновения ошибки при левом сдвиге разрядов частичного остатка в случае, когда два его старших разряда различны, использованы 6-разрядные регистры. См. сноску 2.

$$\begin{aligned}
 -|B| &\neq 10X...X \\
 -|B| &= 110X...X
 \end{aligned}
 \tag{2}$$

Стоит отметить, что для формирования правильного выходного результата необходимо выполнить коррекцию значений частного и остатка в зависимости от знаков операндов. Для каждой комбинации знаков

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 8: Осведомительные сигналы (признаки)

Комбинация	Коррекция
$A \geq 0, B > 0$	Коррекция не требуется
$A \geq 0, B < 0$	Перевести частное в доп. код
$A \leq 0, B > 0$	Результат верен, если остаток = 0. Иначе прибавить к отрицательному частному единицу, перевести остаток в доп. код
$A \leq 0, B \leq 0$	Изменить знак делимого, перевести остаток в доп. код

Таблица 9: Коррекция результата

делимого и делителя реализована отдельная операция коррекции. См таблицу 9.

Соединим все элементы в соответствии с алгоритмом задачи. См. рисунки 23 и 24 в приложении **Приложении А**.

4.4 Реализация управляющего автомата

Приступим к построению управляющего автомата, определяющего последовательность выполнения микрокоманд для деления двух 4-разрядных чисел в дополнительном коде.

Определим разрядность двух ПЗУ, участвующих в построении УА по схеме с сокращенным тактом. Адрес должен иметь 3 разряда — текущее значение параметра S . Микрокоманда представлена в виде 15 бит — 12 признаков, расположенных в следующем порядке $S, H, R0, ERROR, CLR, L_CT, COUNT_CT, SHIFT, L_RG_REM, L_RG_B, L_RG_A, e$. Текущая команда хранится в 15 ти разрядом регистре. В первом ПЗУ, имеющем метку «0» будут храниться микрокоманды, вторая часть адреса которых содержит нулевой бит. Автомат переходит в эти состояния, когда условия, заключенные в операторе выбора, отображенном на диаграмме, не выполняются. Второй ПЗУ с меткой «1» хранит микрокоманды, в которые автомат переходит при выполнении условий, заключенных в операторе выбора. Данные микрокоманды имеют единичный бит во второй части адреса.

Микрокоманды, расположенные в ПЗУ по текущему адресу попадают на вход мультиплексора, адресным входом которого является значение признака, рассматриваемого в текущем состоянии.

Заполним память в соответствии в алгоритмом, подключим ПЗУ, мультиплексоры и регистр последовательным способом. См рисунок 25 в Приложении А.

4.5 Тестирование работы автомата

После реализации операционного и управляющего автомата следует приступить к объединению данных устройств и тестированию их совместной работы. Подключим признаки к входам соответствующих логических элементов и цифровых устройств с помощью туннелей. Добавим блок ввода исходных данных, используя контакты, блок вывода — регистр результата операции деления и регистр остатка, индикатор завершения операции деления.

Проведем проверку корректности выходных результатов построенного цифрового устройства. Разделим два наибольших 4-разрядных двоичных числа $1111_2 \div 1111_2$ ожидая получить частное 1_2 и остаток 0_2 . Укажем входные данные, будем подавать тактовые сигналы до тех пор, пока индикатор не сообщит нам о завершении операции, сравним

практические результаты с ожидаемыми. См рисунок 17. Умножение выполнено корректно. Ожидаемые и полученные результаты совпадают.

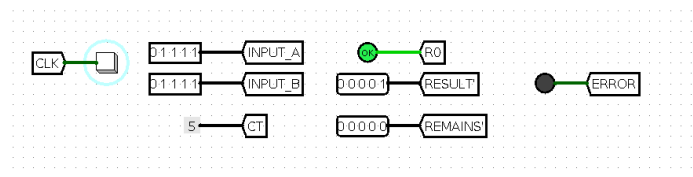


Рис. 17: Проверка работы автомата

Протестируем работу автомата на входных данных, которые гипотетически могут привести к ошибке — поделим число 101_2 на число 1111_2 . Ошибка состоит в том, что в случае использования регистров недостаточной разрядности, левый сдвиг при вычислении очередного остатка произойдет некорректно, что приведет к неверному результату. Введем данные, пронаблюдаем за работой автомата. См рисунок 18. Операция выполнена верно. Использование регистров большей разрядности исключило возможность ошибки.

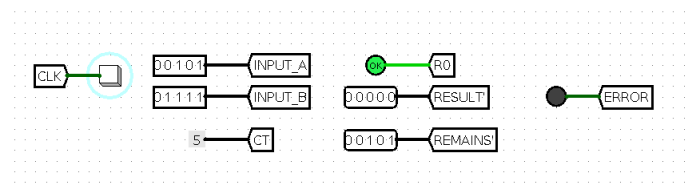


Рис. 18: Вторая проверка работы автомата

Проведем тестирование работы автомата на отрицательных числах. Рассмотрим три варианта, при которых:

1. Делимое неотрицательно, делитель отрицателен;
2. Делимое отрицательно, делитель положителен;
3. Делимое отрицательно, делитель отрицателен.

Рассмотрим конкретные примеры.

№	Комбинация		Ожидаемый	Полученный
1	$8 \div -4$	$01000_2 \div 11100_2$	$11100_2, 00000_2$	$11100_2, 00000_2$
2	$-8 \div 3$	$11000_2 \div 00011_2$	$11110_2, 11110_2$	$11110_2, 11110_2$
3	$-13 \div -3$	$10011_2 \div 11101_2$	$00100_2, 11111_2$	$00100_2, 11111_2$

4.6 Вывод

В ходе данной практической работы было рассмотрено строение и работа управляющего автомата, построенного по схеме с укороченным тактом. Используя полученные знания на практике, на основе данного управляющего автомата построено вычислительное устройство (операционный и управляющий автомат), реализующее операцию деления двух 4-разрядных чисел в дополнительном коде.

Работа данного устройства испытана, проверена корректность полученных результатов при работе со всеми категориями входных данных.

5 Практическая работа №5.

Сложение чисел с плавающей точкой

5.1 Общее строение автомата

В ходе данной практической работы был реализован автомат, выполняющий сложение чисел с плавающей точкой, где мантисса числа представлена в виде 5 разрядов в доп.коде, а порядок в виде 5-ти разрядного положительного целого числа в смещенном коде ($C = 16$). Управляющий автомат был построен по схеме с регулярной адресацией (последовательный вариант). Рассмотрим строение управляющего автомата. См рисунок 19.

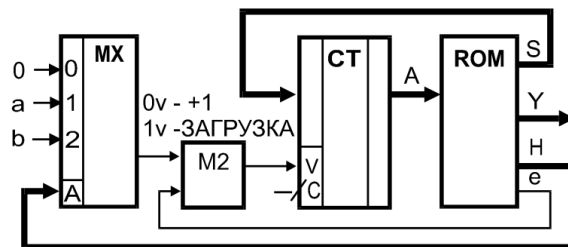


Рис. 19: УА с регулярной адресацией

В конкретной реализации на информационные входы мультиплексора подаются следующие сигналы:

- Константа нуля;
- Ma_IS_NULL (признак нуля мантиссы A);
- Mb_IS_NULL (признак нуля мантиссы B);
- $A < B$ (признак того, что порядок числа A меньше порядка числа B);
- A_IS_ANSWER (признак того, что ответ хранится в регистрах числа A);
- $CT_dP_IS_NULL$ (признак того, что счетчик разницы порядков хранит "ноль");
- $CT_Pa_IS_NULL$ (признак переполнения счетчика порядка числа A в большую сторону);

- $CT_Pa_IS_MAX$ (признак переполнения счетчика порядка числа A в меньшую сторону);
- $|m_a \pm m_b| > 1$ (признак того, что модуль алгебраической суммы операндов больше единицы);
- $op_normalized$ (признак нормализации операндов);

В схему введен элемент M2, позволяющий инвертировать значение входного сигнала, что облегчает распределение микроинструкций по ячейкам управляющей памяти.

5.2 Алгоритм работы автомата

Опишем алгоритм работы автомата с помощью блок схемы. См. рисунок 5.5 и 26 в Приложении А.

После построения алгоритма работы автомата следует перейти к реализации операционной части.

5.3 Реализация Операционного автомата

Построим операционный автомат, выполняющий сложение двух чисел в формате с плавающей точкой. Приведем названия и назначения каждого из регистров, используемых в данном устройстве. См. таблицу 10.

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 10: Регистры операционного автомата

Укажем необходимые признаки, которые впоследствии будут вырабатываться управляющим автоматом. См. таблицу 11. С целью реализации левого и правого сдвига в регистре RG_Ma был построен элемент

Признак	Назначение
<i>S</i>	Хранит адрес следующей операции
<i>H</i>	Адресный вход мультиплексора
<i>R0</i>	Сигнализирует об окончании операции деления
<i>overflow</i>	Сигнализирует об ошибке обработки – переполнение
<i>L_Ma</i>	Загрузка в регистр <i>RG_Ma</i>
<i>SHIFT_Ma</i>	Правый сдвиг регистра <i>RG_Ma</i> если <i>SHIFT_Ma_Left</i> = 0 и левый, если <i>SHIFT_Ma_Left</i> = 1
<i>RST</i>	Асинхронный сброс всех элементов
<i>COUNT_Pa</i>	Счет. Декремент счетчика, если <i>L_CT_Pa</i> == 1
<i>L_CT_Pa</i>	Загрузка счетчика <i>CT_Pa</i>
<i>CHANGE</i>	Выбор источника загрузки в регистры мантисс и порядка чисел А и В
<i>e</i>	Управляющий сигнал для счетчика. Если <i>e</i> = 1, следует выполнить загрузку, а если <i>e</i> = 0 – инкрементировать счетчик.

Таблица 11: Осведомительные сигналы (признаки)

памяти, позволяющий выбрать направление сдвига с помощью двух управляющих сигналов. Данный элемент был размещен в отдельном файле и загружался в основной файл как внешняя библиотека Logisim. Устройство данного регистра можно увидеть на рисунке 30 в Приложении А.

При выполнении операции сложения предполагается, что числа, переданные на вход находятся в нормализованном виде, то есть имеют вид, представленный на сноске ??.

$$\begin{aligned}
 \frac{1}{2} &\leq |M| < 1 \\
 M &= 0.1XXXX \\
 M &= 1.0XXXX \\
 M &= 1.00000
 \end{aligned}
 \tag{3}$$

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 12.

Соединим все элементы в соответствии с алгоритмом задачи. См.

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 12: Нормализация результата

рисунки 27 и 28 в приложении **Приложении А**.

5.4 Реализация управляющего автомата

Приступим к построению управляющего автомата, определяющего последовательность выполнения микрокоманд для сложения двух чисел в формате с плавающей точкой.

Определим разрядность ПЗУ, участвующего в построении УА по схеме с постоянной адресацией. Адрес должен иметь 5 разрядов — текущее значение параметра S . Микрокоманда представлена в виде 22 бит — 15 признаков, расположенных в следующем порядке: S , $R0$, RST , L_ma , $SHIFT_Ma$, $SHIFT_Ma_LEFT$, L_CT_Pa , $COUNT_Pa$, $CHANGE$, L_CT_dP , $COUNT_dP$, m_n , H , e . Адрес текущей команды хранится в 5-ти разрядном счетчике. К входам мультиплексора подключены сигналы, значения которых анализируются в данном состоянии автомата. Они описаны выше. См. **список**.

Заполним память в соответствии в алгоритмом, подключим ПЗУ, мультиплексор и счетчик последовательным способом. См рисунок 29 в **Приложении А**.

Ввод и вывод результатов осуществляется с помощью блока ввода и вывода. Числа передаются в нормализованном формате со смещением порядка в $C=16$. См. рисунок 31 в **Приложении А**.

5.5 Вывод

В ходе данной практической работы было рассмотрено строение и работа управляющего автомата, построенного по схеме регулярной адресацией. Используя полученные знания на практике, на основе данного управляющего автомата построено вычислительное устройство (операционный и управляющий автомат), реализующее операцию сложения двух чисел в формате с плавающей точкой.

ПРИЛОЖЕНИЕ А

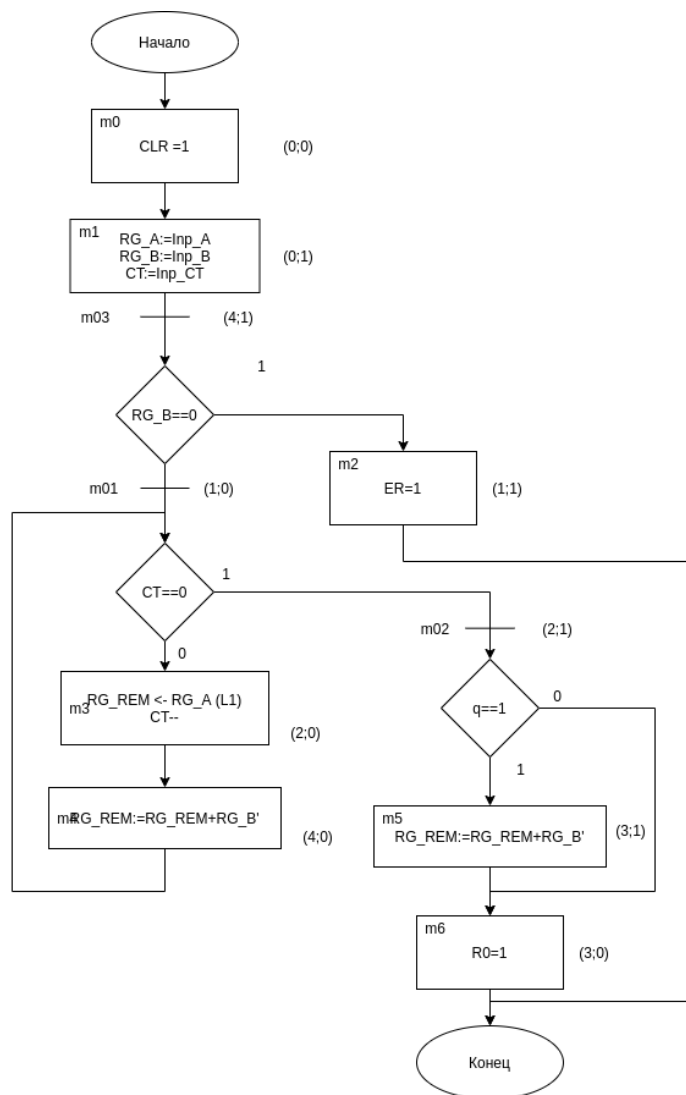


Рис. 20: Алгоритм деления двух 4-разрядных чисел

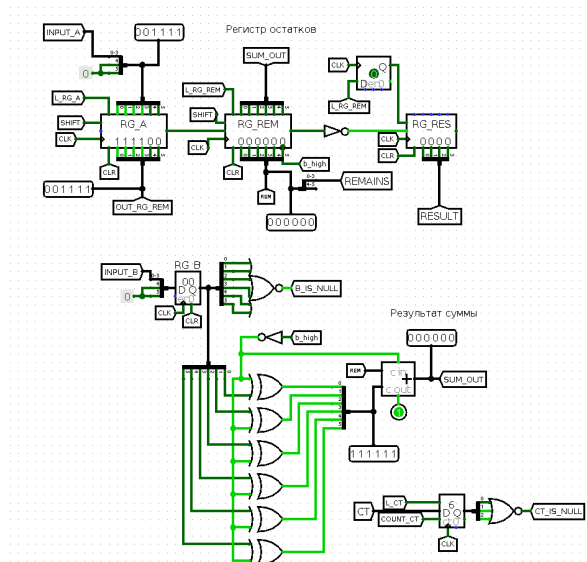


Рис. 21: Схема операционного автомата

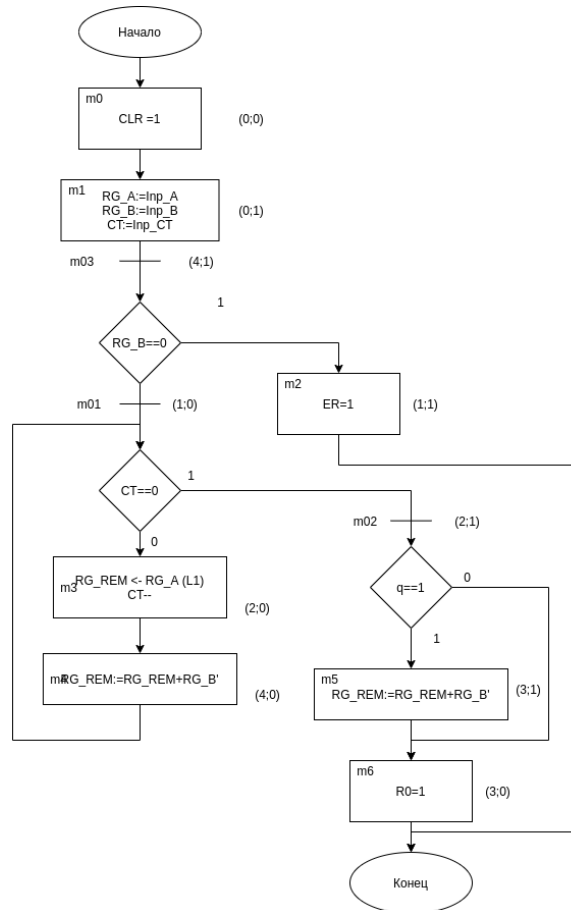


Рис. 22: Алгоритм деления двух 4-разрядных чисел

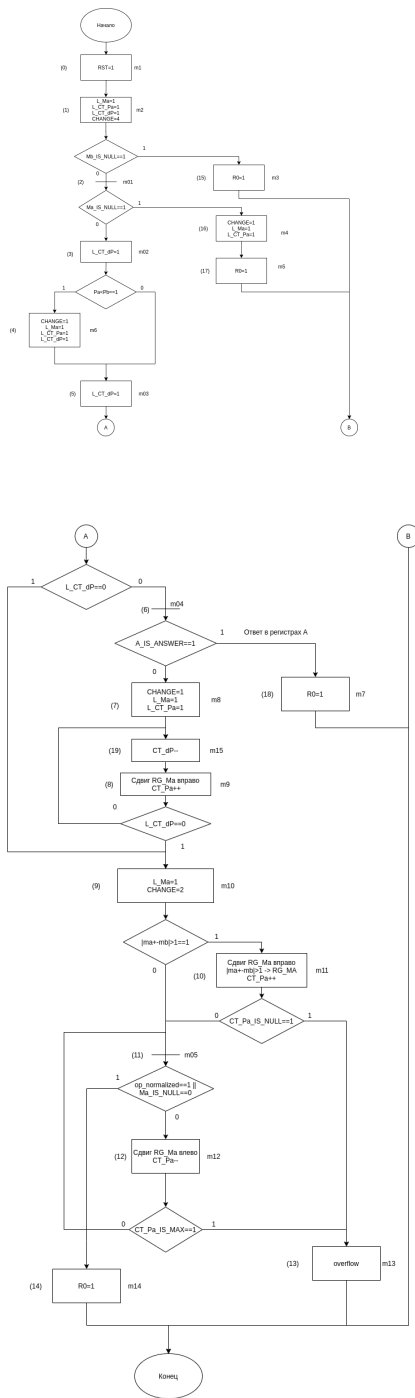


Рис. 26: Алгоритм сложения чисел с плавающей точкой. Часть 2

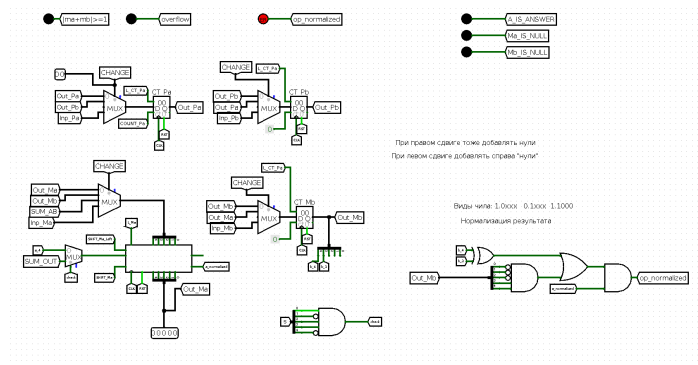


Рис. 27: Схема операционного автомата, часть 1

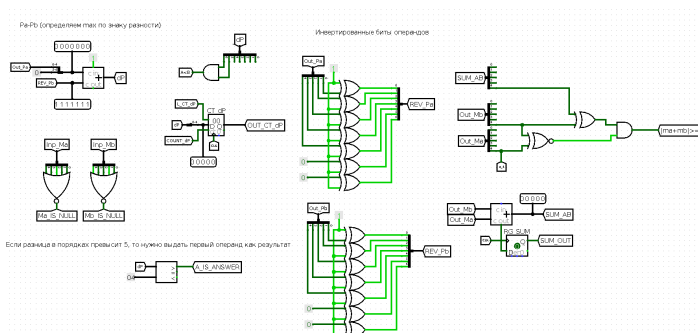


Рис. 28: Схема операционного автомата, часть 2

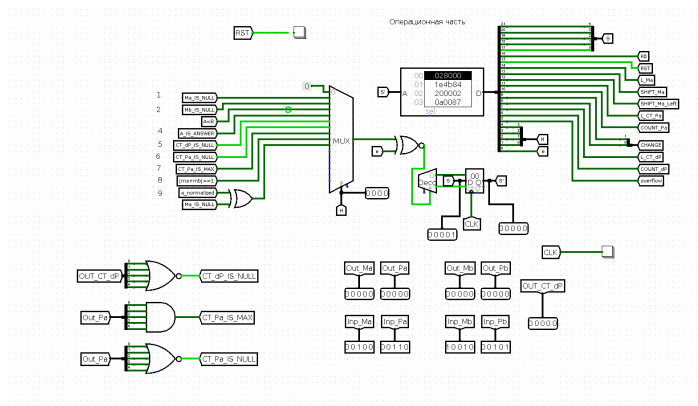


Рис. 29: Схема управляющего автомата

