МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Рыбинский государственный авиационный технический университет
имени П. А. Соловьева»

Институт информационных технологий и систем управления

Кафедра вычислительных систем

Пояснительная записка

к программной части курсового проекта

по дисциплине «Микропроцессорные системы»

на тему «Генератор импульсов»

Студент группы ИВБ1–20 Морозов В. В.

Руководитель канд. техн. наук, профессор Комаров В. М.

Нормконтролер канд. техн. наук, профессор Комаров В. М.

Рыбинск 2023

Содержание

Введение

При современном уровне развития элементной базы вычислительной техники наиболее целесообразно для построения устройств цифровой обработки информации использовать микропроцессоры. Использование микропроцессоров обеспечивает возможность построения таких устройств в виде одной печатной платы. При этом они имеют малые габариты и массу, небольшую стоимость и потребляемую мощность.

1 Анализ технического задания

Из анализа технического задания следует, устройство должно обеспечивать:

1. Генерацию импульсов прямоугольной формы;
2. Изменение полярности импульсов;
3. Изменение частоты от 20 Гц до 150 Гц с шагов в 10 Гц;
4. Изменение амплитуды импульсов от 0 В до 5 В с шагов в 1 В;
5. Изменение длительности импульсов от 1мс до 5 мс с шагов в 1 мс;
6. Вывод сгенерированных импульсов на дисплей, состоящий из 16 матричных индикаторов размерностью 8 строк и 8 столбцов;
7. Отображение заданной частоты на дисплей, состоящий из 3 знакосинтезирующих семисегментных индикаторов;
8. Отображение заданной амплитуды на дисплей, состоящий из 1 знакосинтезирующего семисегментного индикатора;
9. Отображение заданной длительности импульсов на дисплей, состоящий из 1 знакосинтезирующего семисегментного индикатора;
10. Для задания частоты используются кнопки без фиксации (+ 10 Гц и – 10 Гц)
11. Для задания амплитуды используются кнопки без фиксации (+ 1 В и – 10 В)
12. Для задания длительности импульсов используются кнопки без фиксации (+ 1 мс и – 1 мс);
13. Для изменения полярности используется кнопка без фиксации.

На основании перечисленных требований можно представить разрабатываемое устройство в виде «черной сферы» (рисунок 1.1), а также изобразить лицевую панель устройства (рисунок 1.2). Исходя из этих представлений начинается техническое проектирование устройства.

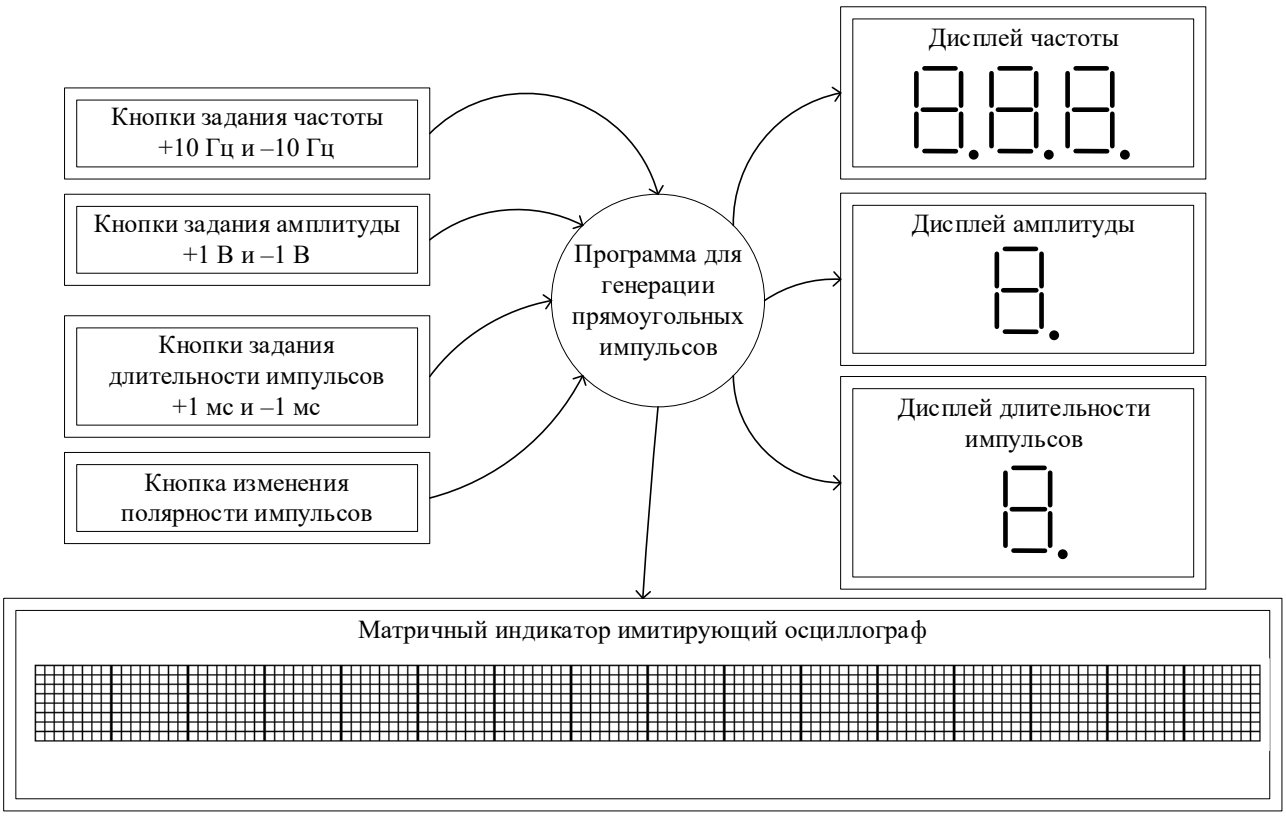


Рисунок 1.1 – Представление устройства охранной сигнализации в виде «черной сферы»

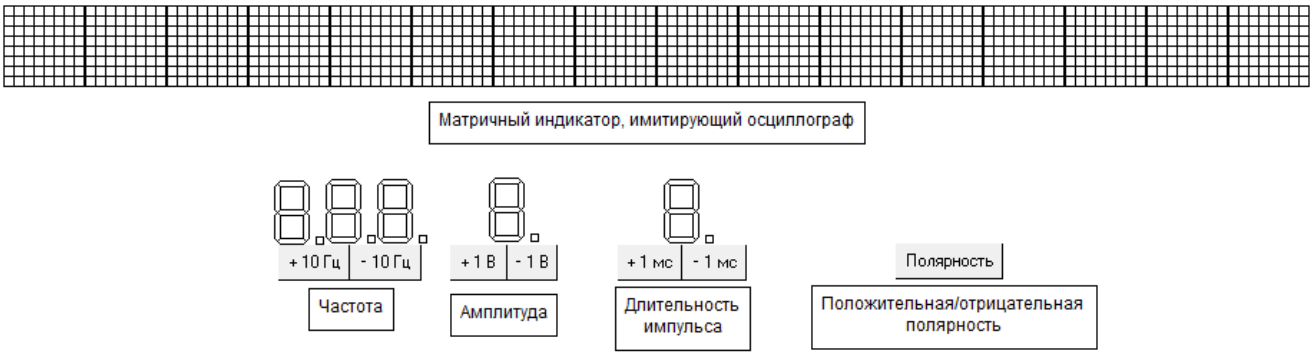


Рисунок 1.2 – Лицевая панель устройства для генерации импульсов

2 Декомпозиция программы

Анализируя требования к программе с учётом лицевой панели устройства, получим исходную схему представления поставленной задачи.

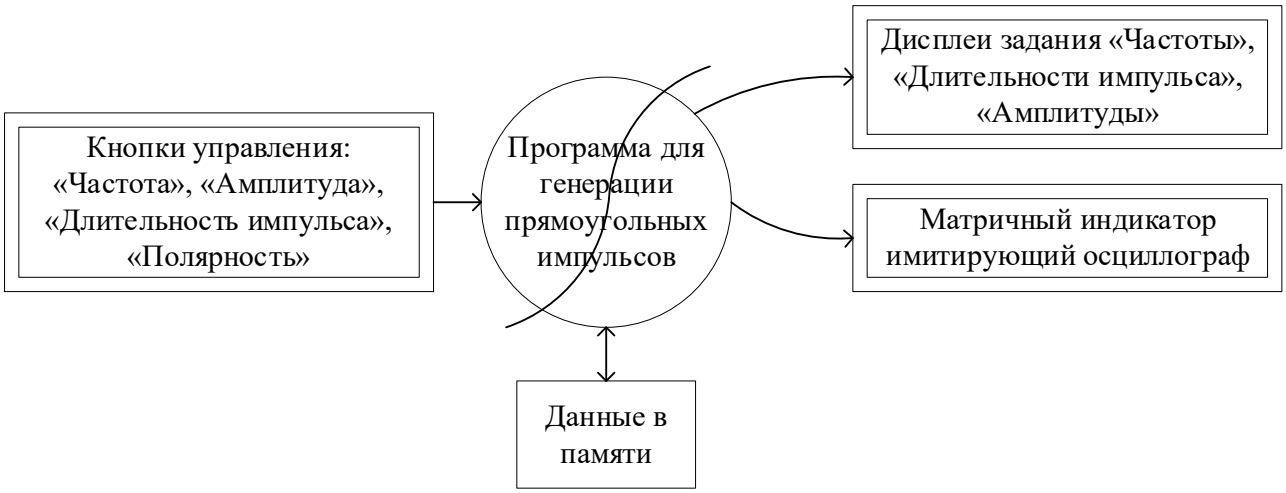


Рисунок 2.1 – Исходная схема представления задачи «Генератор импульсов»

Из этой схемы следует, что проектируемая программа должна обрабатывать входную информацию с кнопок управления, выходная информация с которых имеет два значения: лог. «0» и лог. «1», отображать выходную информацию на 3 дисплеях, состоящих из закосинтезирующих семисегментных индикаторов, и дисплее, состоящем из матричных индикаторов.

На первом этапе (рисунок 2.2) представим программу в виде двух задач: «Обработка входной информации», «Обработка выходной информации». Подзадачи «Обработка входной информации» и «Обработка выходной информации», выделенные на первом этапе, являются сложными и требуют дальнейшего разбиения.

Рисунок 2.2 – Статическая модель программы (после первого этапа декомпозиции)

На втором этапе (рисунок 2.3) подзадачу «Обработка входной информации» разобьем на подзадачи «Ввод с кнопок», «Контроль ввода»

Подзадачу «Обработка выходной информации» разобьем на подзадачи «Формирование информации», «Преобразование в десятичный код», «Формирование массива отображения», «Формирование изображения импульсов», «Вывод изображения импульсов на матрицу», «Вывод числовой информации».

Подзадачи, выделенные на втором этапе, являются простыми и не требуют дальнейшей декомпозиции.

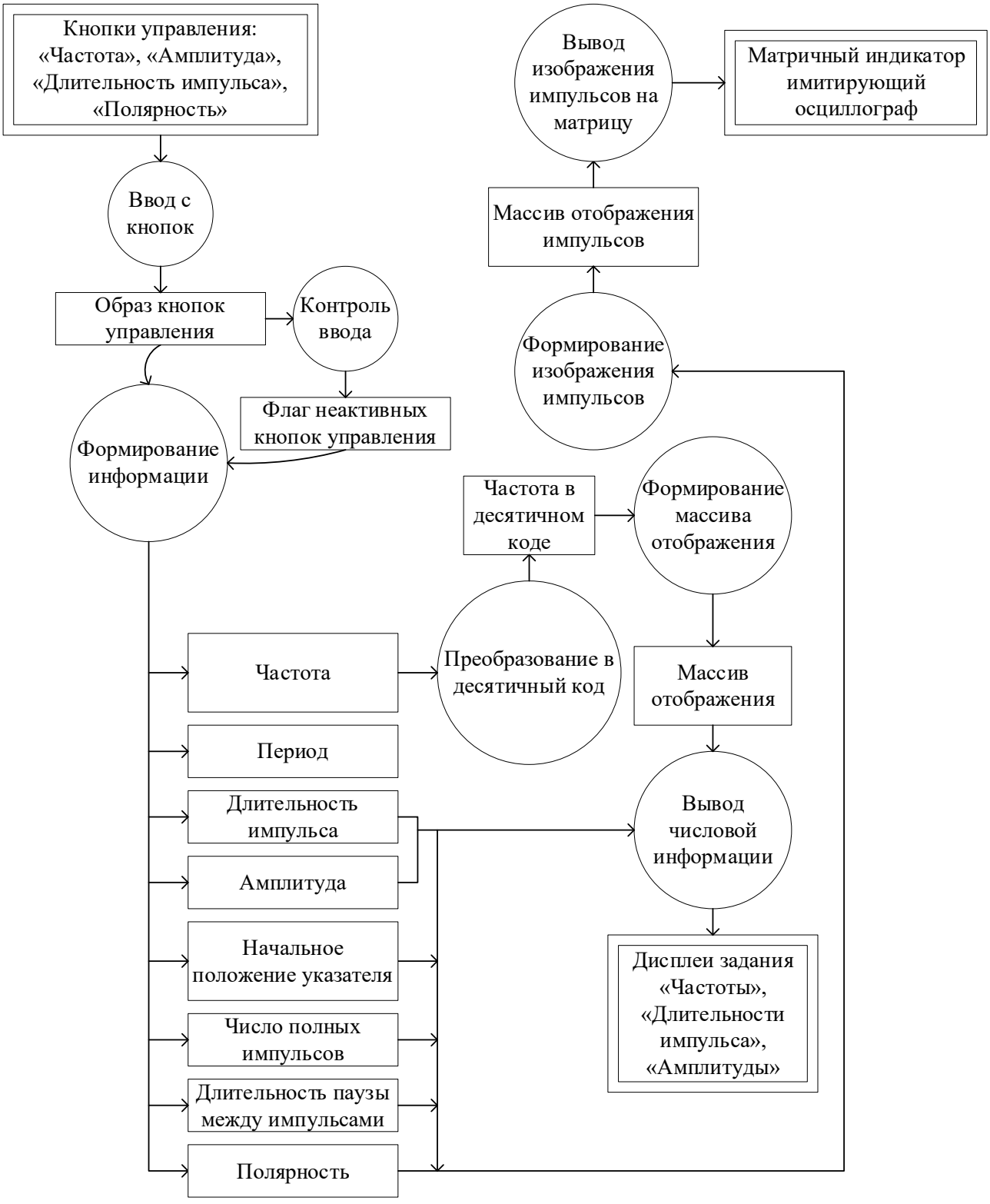


Рисунок 2.3 – Статическая модель программы (после второго этапа декомпозиции)

3 Разработка структуры данных программы

Процесс решения любой задачи состоит из активации тех или иных программных модулей, выполняющих некоторую подзадачу. Передача информации между модулями осуществляется с помощью наборов данных.

Дальнейшее проектирование программы заключается в переходе от ее статической модели к динамической, то есть к ее алгоритмическому описанию.

Таблица 3.1 – Структура данных программы

Наименование данных	Символьное имя	Формат данных	Кодирование данных								Примечание
			D7	*D6*	*D5*	*D4*	*D3*	*D2*	*D1*	*D0*	
Амплитуда	*Amplitude*	Числовой байт	0	0	0	0	0	1	0	1	A = 5 В
Частота	*Frequancy*	Числовой байт	1	0	0	1	0	1	1	0	f = 150 Гц
Длительность импульса	*Pulse-Duration*	Числовой байт	0	0	0	0	0	1	0	1	tи = 5 мс
Период	*Pulse-Period*	Числовой байт	0	0	0	0	0	1	1	0	T = 6 мс
Число полных импульсов	*Pulses-Count*	Числовой байт	0	0	0	1	0	1	0	1	21
Длительность паузы между импульсами	*Pause-Period*	Числовой байт	0	0	0	0	0	0	0	1	tп = 1 мс
Начальное положение указателя	*Start-Position*	Числовой байт	0	0	1	0	0	0	0	0	5 строка матрицы (20h)
Образ кнопки	*KeyImage*	Упакованный байт	*	0	0	0	0	0	0	0	

Наименование данных	Символьное имя	Формат данных	Кодирование данных								Примечание
			D7	*D6*	*D5*	*D4*	*D3*	*D2*	*D1*	*D0*	
Частота в десятичном коде	*Frequancy BCD*	Массив двоично-десятичных байтов (2 байта)	0 *	1 *	0 *	1 *	0 0	0 0	0 0	0 1	50 1
Массив отображения частоты	*Frequancy Image*	Массив байтов (3 байта)	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 1 0	0 0 0	0 1 1	0 5 1
Массив отображения импульсов	*Pulses-Image*	Массив байтов (128 байтов)	* . *	* . *	* . *	* . *	* . *	* . *	* . *	* . *	1 колонка 128 колонка
Флаг неактивных кнопок управления	*NoInput-ErrorFlag*	Флаговый байт	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	Есть нажатие Нет нажатия
Флаг полярности	*Polarity-Flag*	Флаговый байт	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1	Положительная полярность Отрицательная полярность

4 Алгоритмизация программы

Вычислительный процесс, реализуемый программой, представляет собой последовательность дискретных шагов по преобразованию данных.

Для решения поставленной задачи необходимы следующие программные модули с соответствующими символьными именами:

- «Функциональная подготовка» (*Initialization*);
- «Ввод с кнопок» (*KeyRead*);
- «Контроль ввода» (*KeyCheck*);
- «Формирование информации» (*DataSetting*);
- «Преобразование частоты из двоичного в десятичный код» (*BinaryToBCD*);
- «Формирование массива отображения» (*UnpackFrequancyBCD*);
- «Вывод числовой информации» (*DisplayData*);
- «Формирование изображения импульсов» (*PulseImageForming*);
- «Вывод изображения импульсов на матрицу» (*MatrixOutput*).

Символьные имена присвоены соответствующим программным модулям с целью их дальнейшего использования. Выбранные имена отражают содержательный смысл этих модулей.

Последовательность модулей в логической конструкции «Следование» определяется логикой решения задачи.

Алгоритм макроуровня программы проектируемого устройства в одноуровневом представлении приведен на рисунке 4.1. Для повышения информативности ГСА на ней изображены входные и выходные данные, обрабатываемые каждым программным модулем. Это позволяет проследить, что каждый модуль располагается в той точке программного кольца, в которой существуют все обрабатываемые им данные.

Далее необходимо перейти к двухуровневому представлению алгоритма. Этот переход осуществляется формально. Алгоритм программы проектируемого устройства в двухуровневом представлении приведен на рисунке 4.2. Из этого

алгоритма следует, что на макроуровне программы находятся лишь команды вызова программных модулей.

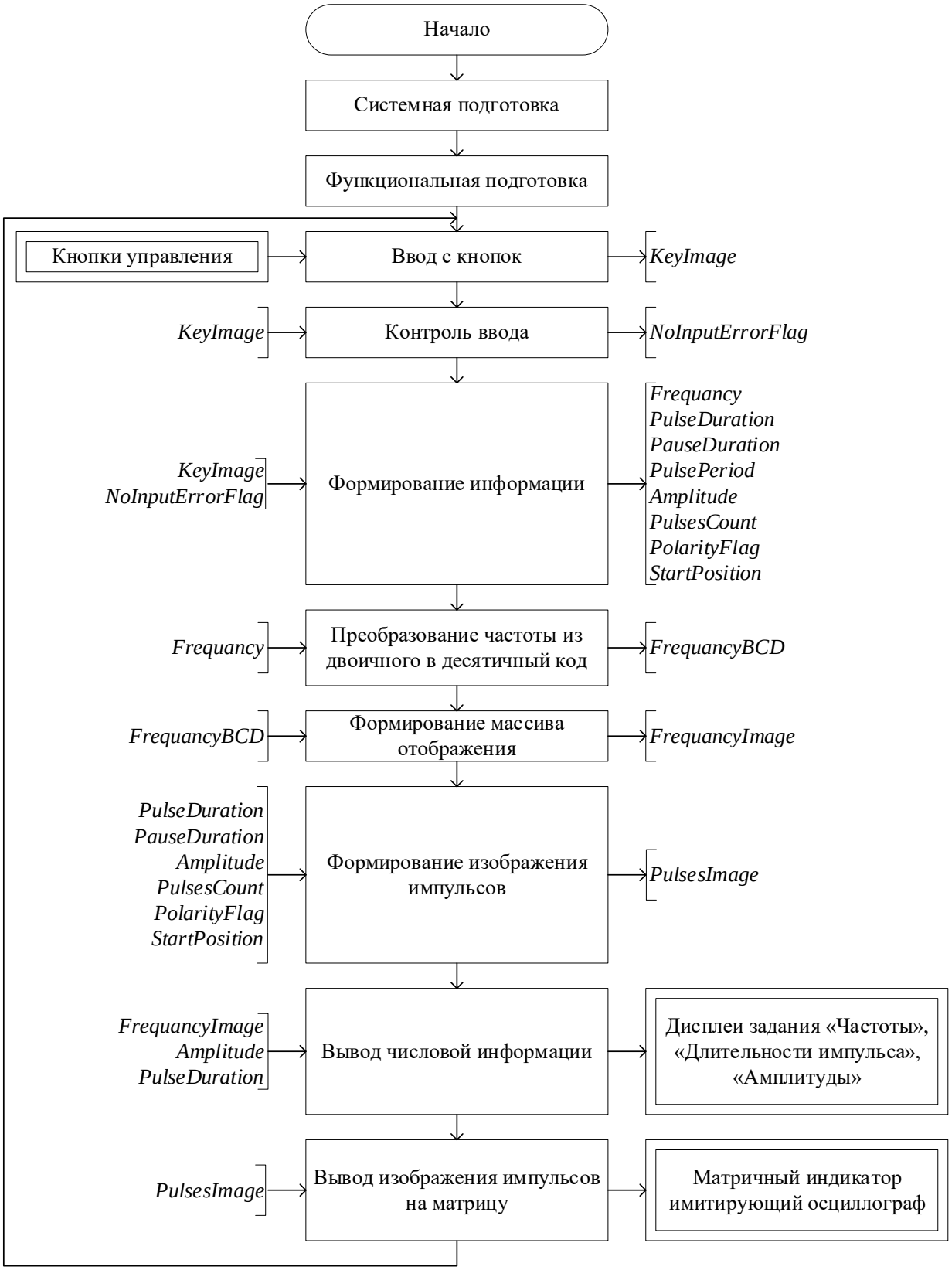


Рисунок 4.1 – Алгоритм программы «Генератор импульсов» (одноуровневое представление)

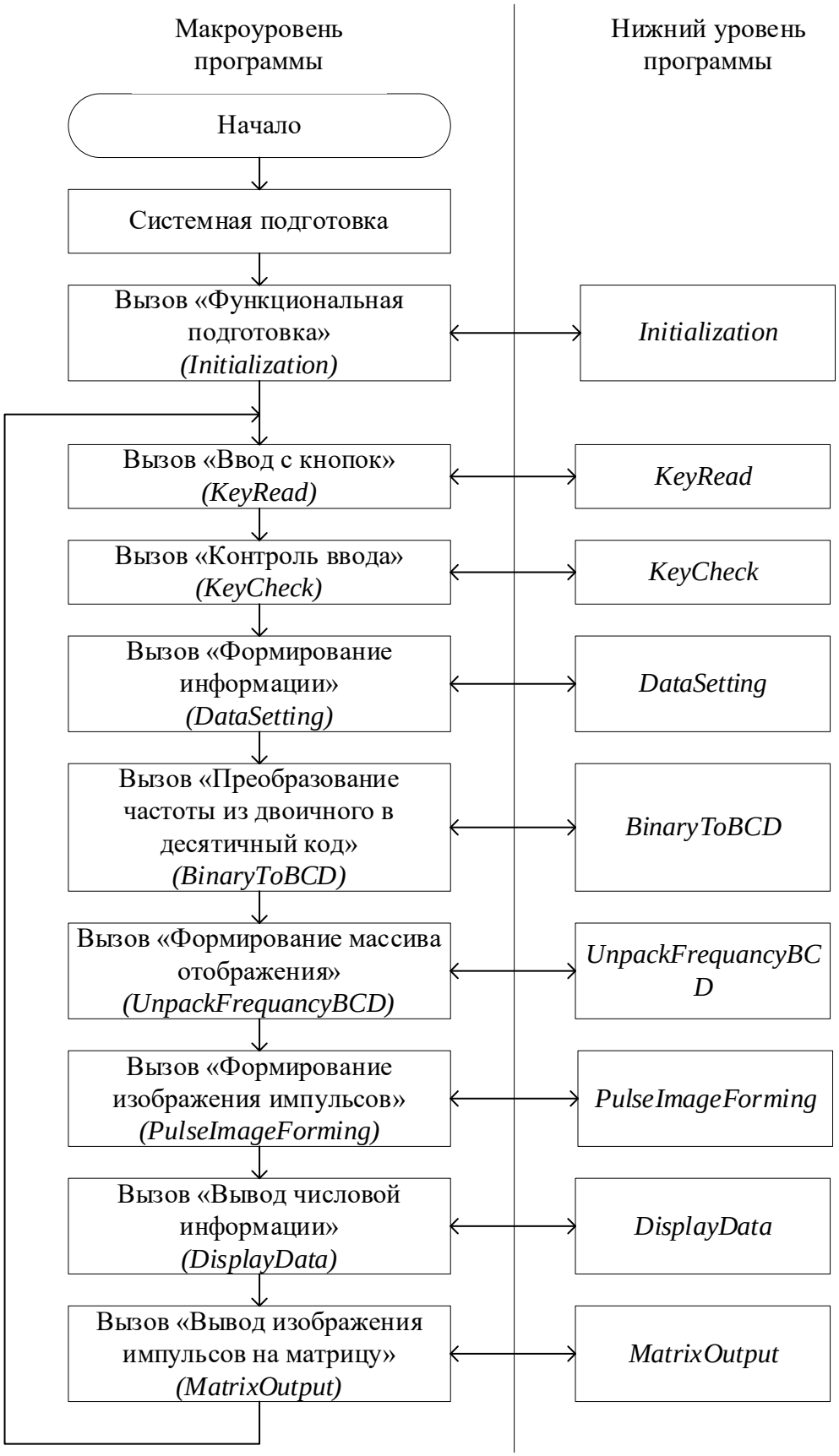


Рисунок 4.2 – Алгоритм программы «Генератор импульсов» (двухуровневое представление)

После разработки алгоритма макроуровня требуется разработать алгоритмы всех программных модулей нижнего уровня. Алгоритмы программных модулей и подмодулей приведены на рисунках 4.3 – 4.31.

Модуль «Ввод с кнопок» (KeyRead)

Этот модуль служит для ввода состояния кнопок задания параметров с последующим размещением их образа в памяти.

ГСА этого модуля приведена на рисунке 4.3.

Электромеханическим переключателям свойственно явление дребезга контактов, который необходимо гасить. Гашение дребезга контактов можно выполнить программным путем. ГСА подмодуля гашения дребезга *VibrDestr* приведена на рисунке 4.4. Критерием окончания дребезга является считывание с кнопок одного и того же состояния заданное количество раз.

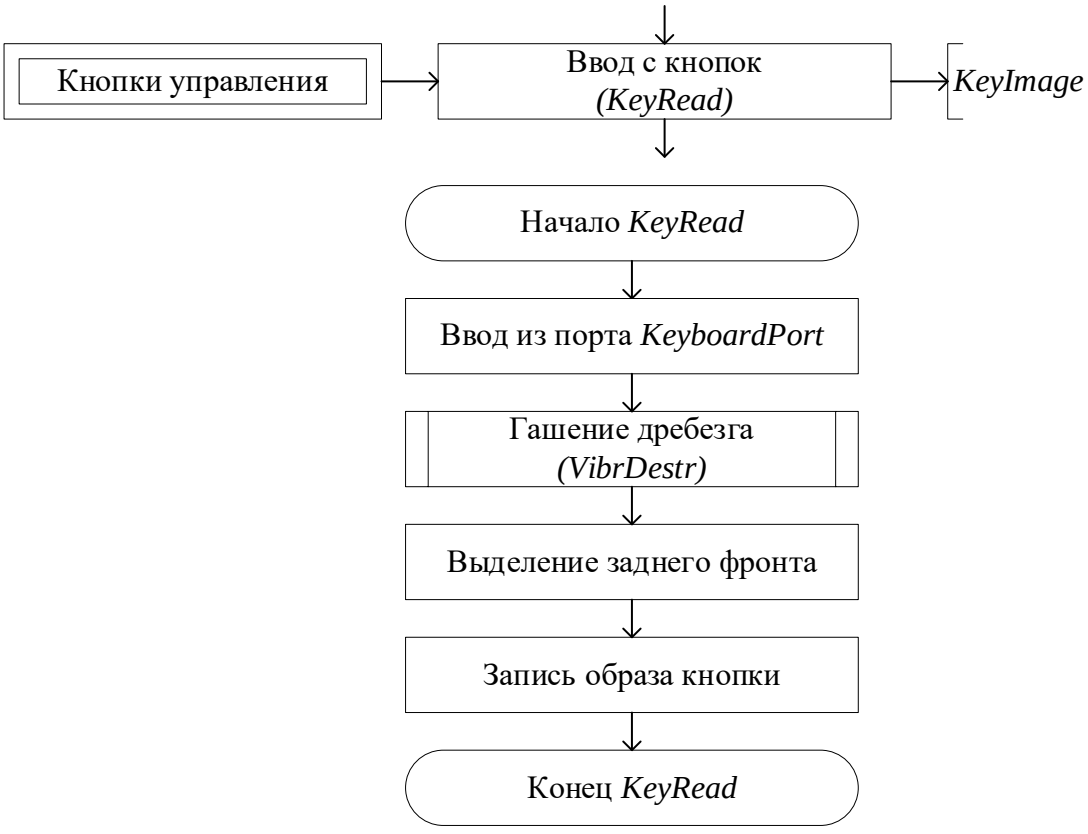


Рисунок 4.3 – Алгоритм модуля «Ввод с кнопок» (общий алгоритм)

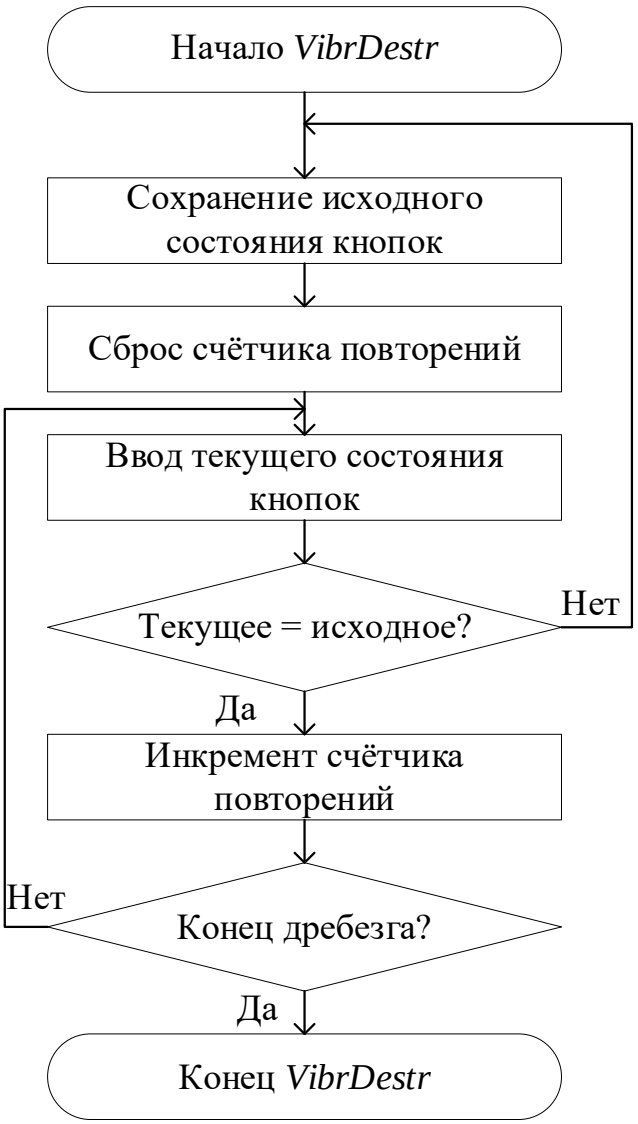


Рисунок 4.4 – Алгоритм подмодуля «Гашение дребезга»

Модуль «Контроль ввода» (KeyCheck)

Этот модуль служит для проверки наличия нажатия кнопок с последующим формированием флага отсутствия нажатия и размещением его в памяти.

ГСА этого модуля приведена на рисунке 4.5.

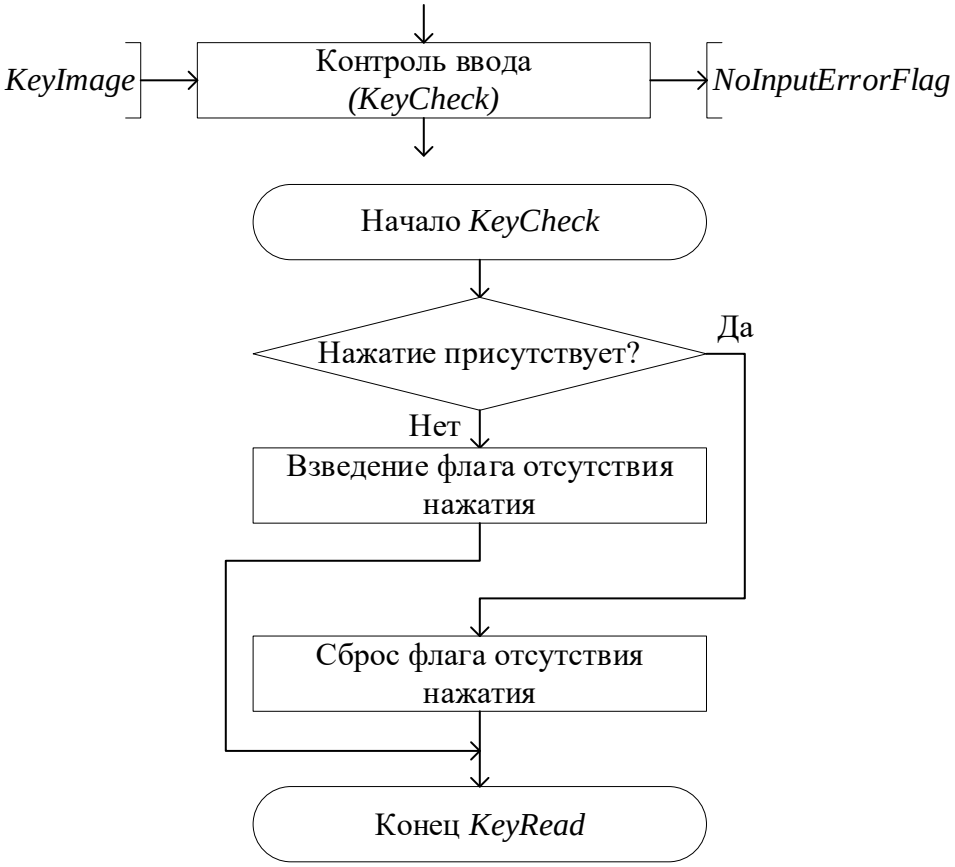


Рисунок 4.5 – Алгоритм модуля «Контроль ввода»

Модуль «Формирование информации» (DataSetting)

Этот модуль служит для задания параметров генерации импульсов исходя из образа нажатой кнопки и флага отсутствия нажатия с последующим размещением этих параметров в памяти. Для удобства этот модуль разбит на ряд подмодулей.

ГСА этого модуля приведена на рисунке 4.6.

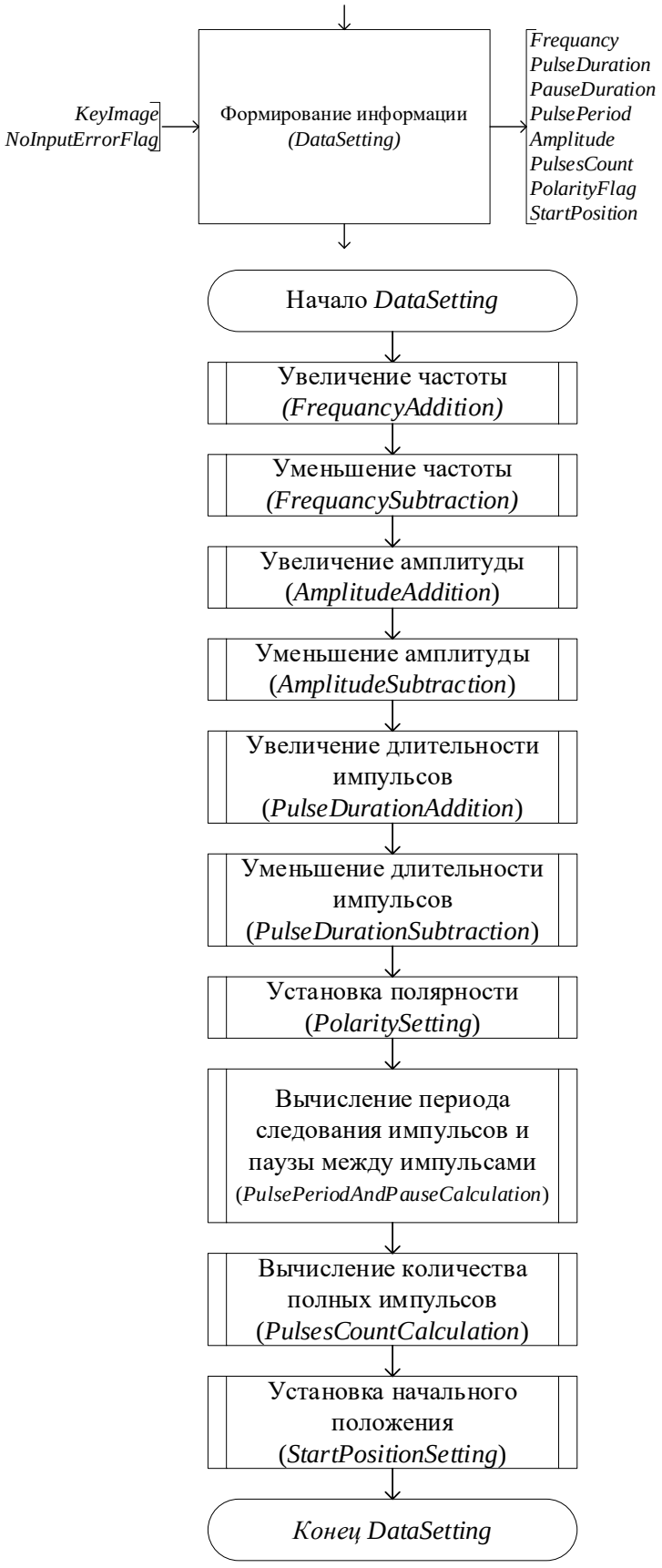


Рисунок 4.6 – Алгоритм модуля «Формирование информации» (общий алгоритм)

Подмодуль «Увеличение частоты» (FrequancyAddition)

Этот подмодуль служит для увеличения параметра частоты импульсов непосредственно в ячейке памяти, то есть происходит сложение константы с ячейкой памяти.

ГСА этого модуля приведена на рисунке 4.7.

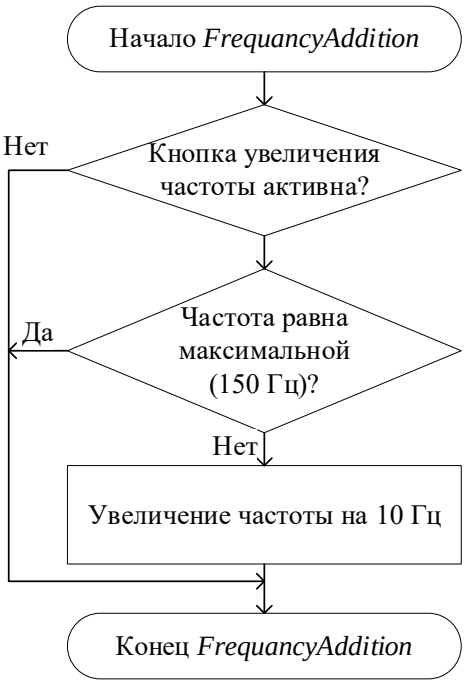


Рисунок 4.7 – Алгоритм подмодуля «Увеличение частоты»

Подмодуль «Уменьшение частоты» (FrequancySubtraction)

Этот подмодуль служит для уменьшения параметра частоты импульсов непосредственно в ячейке памяти, то есть происходит вычитание константы из ячейки памяти.

ГСА этого модуля приведена на рисунке 4.8.

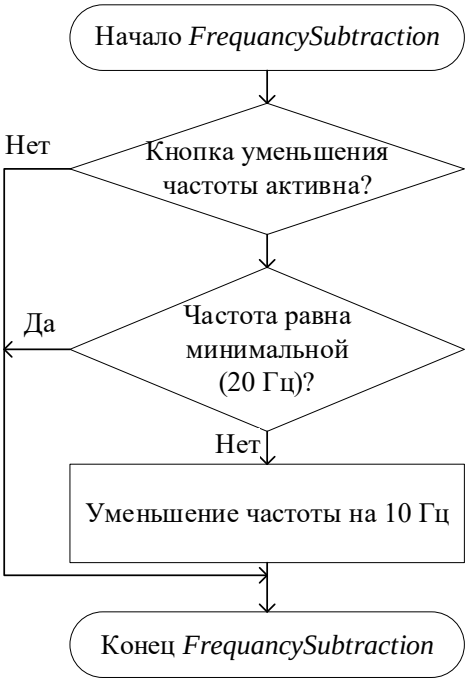


Рисунок 4.8 – Алгоритм подмодуля «Уменьшение частоты»

Подмодуль «Увеличение амплитуды» (AmplitudeAddition)

Этот подмодуль служит для увеличения параметра амплитуды импульсов непосредственно в ячейке памяти, то есть происходит инкремент ячейки памяти.

ГСА этого модуля приведена на рисунке 4.9.

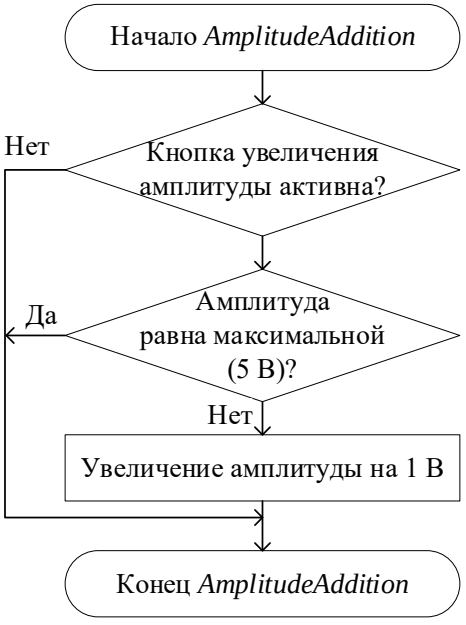


Рисунок 4.9 – Алгоритм подмодуля «Увеличение амплитуды»

Подмодуль «Уменьшение амплитуды» (AmplitudeSubtraction)

Этот подмодуль служит для уменьшения параметра амплитуды импульсов непосредственно в ячейке памяти, то есть происходит декремент ячейки памяти.

ГСА этого модуля приведена на рисунке 4.10.

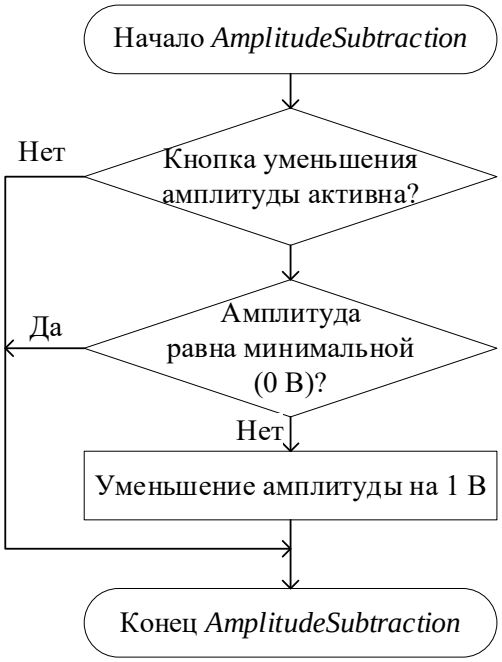


Рисунок 4.10 – Алгоритм подмодуля «Уменьшение амплитуды»

Подмодуль «Увеличение длительности импульсов» (PulseDurationAddition)

Этот подмодуль служит для увеличения параметра длительности импульсов непосредственно в ячейке памяти, то есть происходит инкремент ячейки памяти.

ГСА этого модуля приведена на рисунке 4.11.

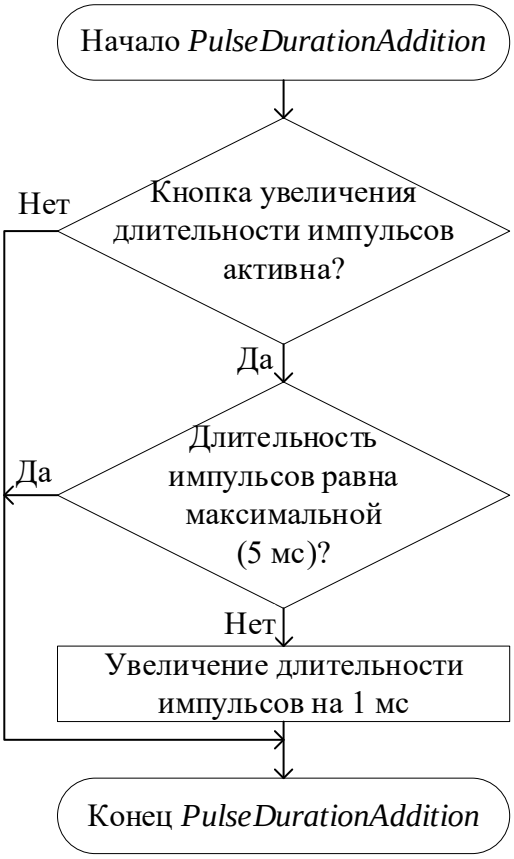


Рисунок 4.11 – Алгоритм подмодуля «Увеличение длительности импульсов»

Подмодуль «Уменьшение длительности импульсов»

(PulseDurationSubtraction)

Этот подмодуль служит для уменьшения параметра длительности импульсов непосредственно в ячейке памяти, то есть происходит декремент ячейки памяти.

ГСА этого модуля приведена на рисунке 4.12.

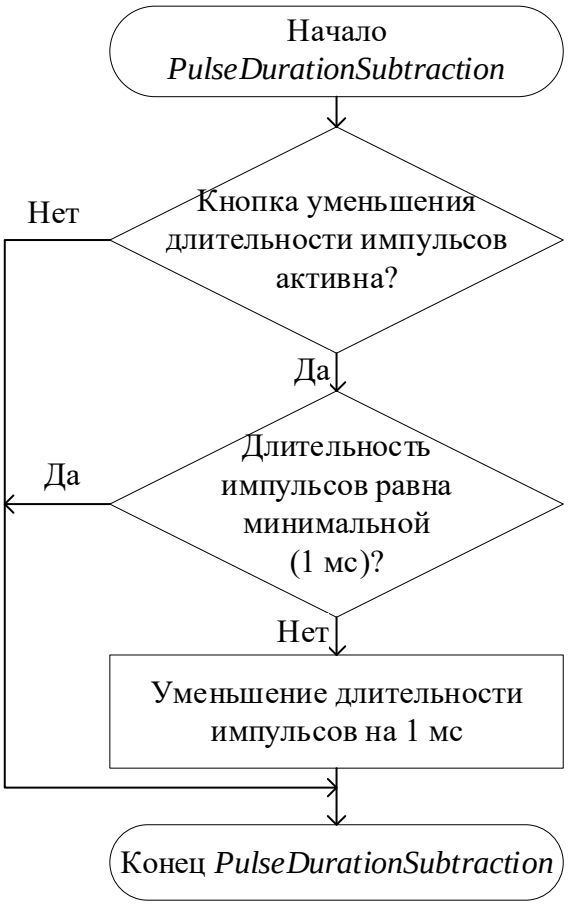


Рисунок 4.12 – Алгоритм подмодуля «Уменьшение длительности импульсов»

Подмодуль «Установка полярности» (PolaritySetting)

Этот подмодуль служит для изменения параметра полярности импульсов. Параметр реализован флаговым байтом, который инвертируется по нажатию кнопки «Полярность».

ГСА этого модуля приведена на рисунке 4.13.

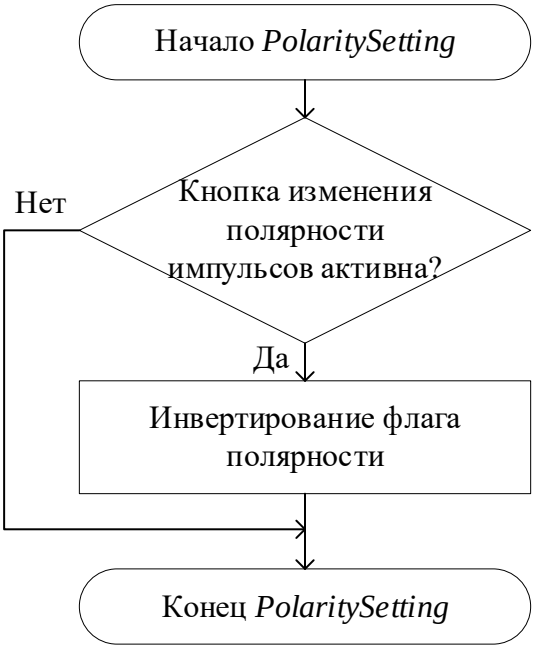


Рисунок 4.13– Алгоритм подмодуля «Установка полярности»

Подмодуль «Вычисление периода следования импульсов и паузы между импульсами» (PulsePeriodAndPauseCalculation)

Этот подмодуль служит для вычисления параметра периода следования импульсов исходя из заданной частоты и на основе уже известных периода и длительности импульсов вычисляется длительность паузы между импульсами. Вычисляется период путём деления значения частоты на константу времени. Длительность паузы между импульсами вычисляется путём вычитания длительности импульса из периода.

ГСА этого модуля приведена на рисунке 4.14.

Рисунок 4.14– Алгоритм подмодуля «Вычисление периода следования импульсов и паузы между импульсами»

Подмодуль «Вычисление количества полных импульсов»

(PulsesCountCalculation)

Этот подмодуль служит для вычисления количества полных импульсов, которые могут поместиться на матричном индикаторе, для последующего формирования изображения импульсов. Вычисляется путём деления общего количество колонок всех матриц на период следования импульсов.

ГСА этого модуля приведена на рисунке 4.15.

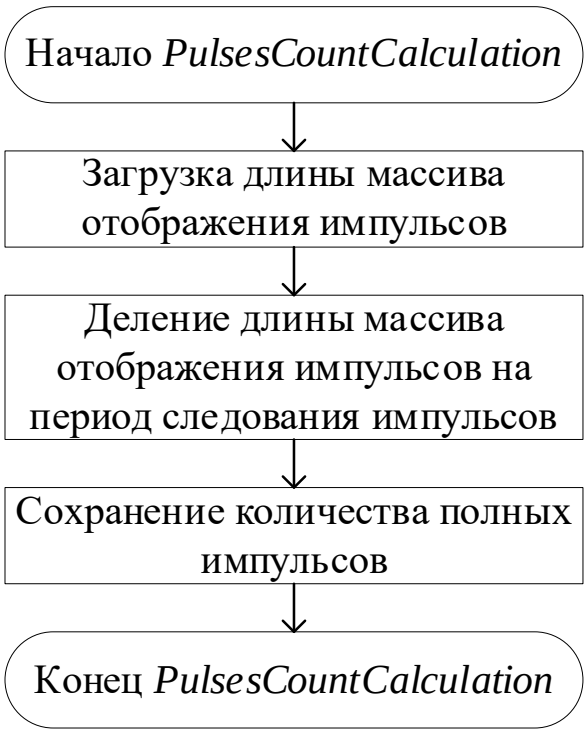


Рисунок 4.15 – Алгоритм подмодуля «Вычисление количества полных импульсов»

Подмодуль «Установка начального положения» (StartPositionSetting)

Этот подмодуль служит для установки начального положения указателя исходя из заданной полярности импульсов, который используется для последующего формирования изображения импульсов.

ГСА этого модуля приведена на рисунке 4.16.

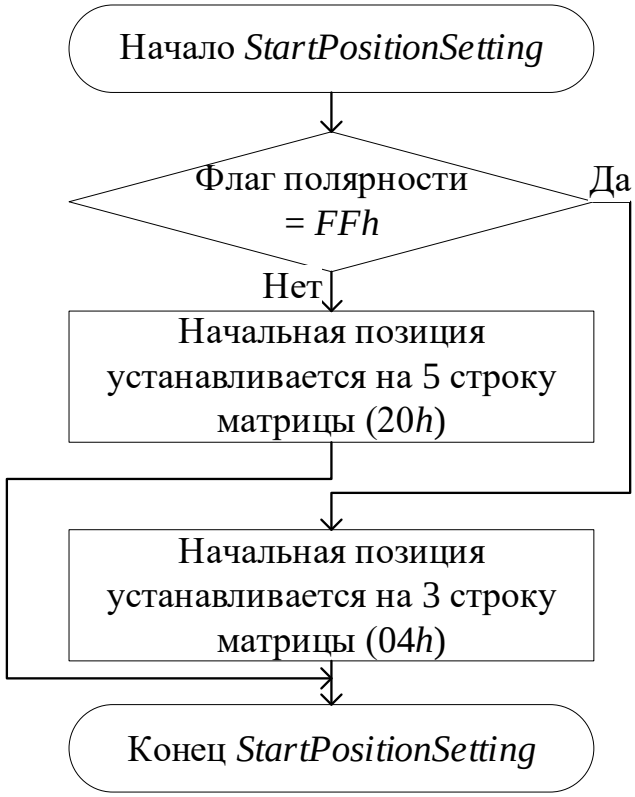


Рисунок 4.16 – Алгоритм подмодуля «Установка начального положения»

Модуль «Преобразование частоты из двоичного в десятичный код» (BinaryToBCD)

Этот модуль служит для перевода частоты из двоичного кода в десятичный упакованный *BCD* код. Десятичный код с последующей распаковкой необходим для табличного преобразования и последующего вывода значения частоты на дисплей.

ГСА этого модуля приведена на рисунке 4.17.

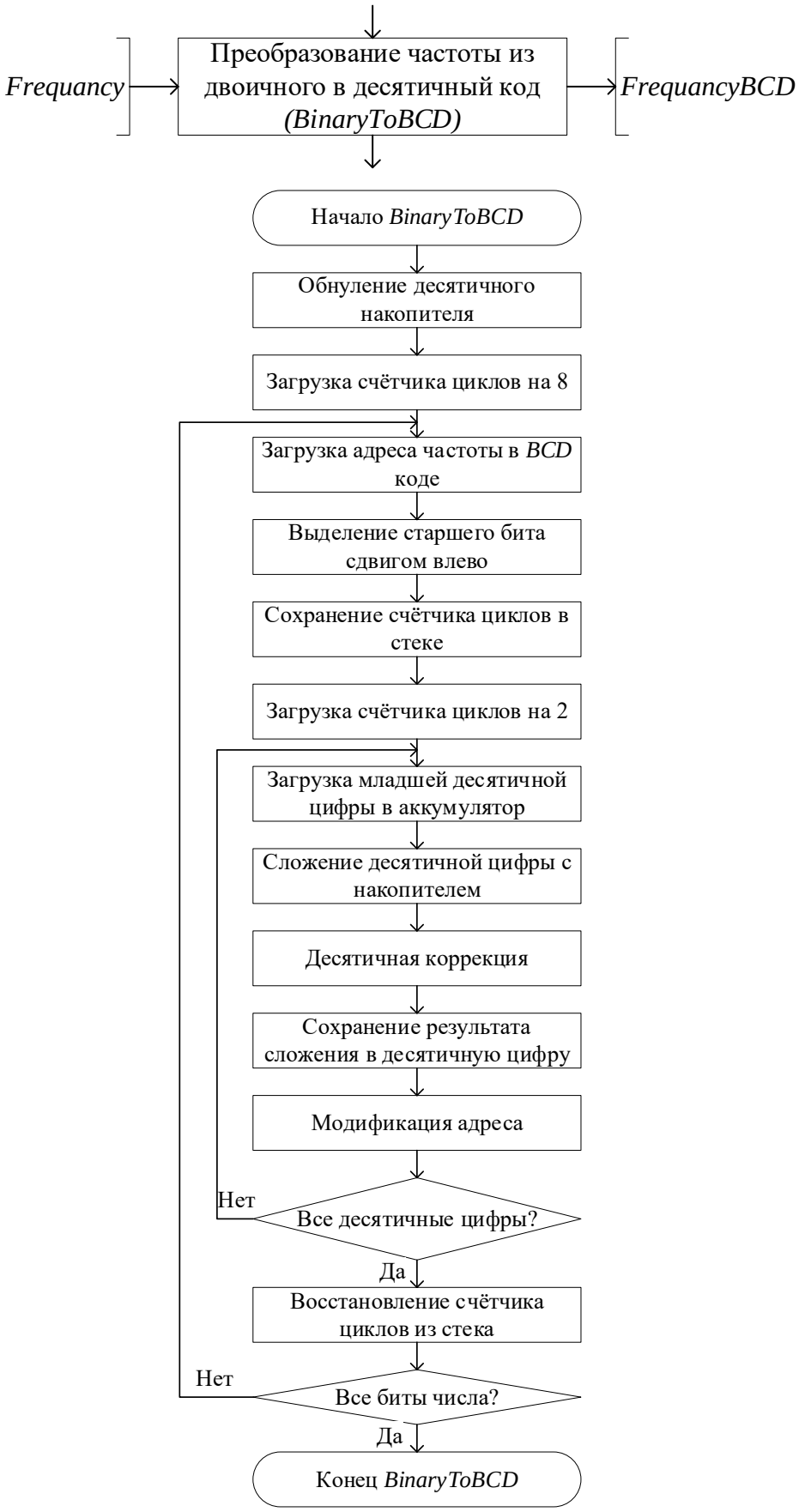


Рисунок 4.17 – Алгоритм модуля «Преобразование частоты из двоичного в десятичный код»

Модуль «Формирование массива отображения» (UnpackFrequancyBCD)

Этот модуль служит для распаковки значения частоты в десятичном *BCD* коде. Распакованный формат можно непосредственно использовать в качестве массива отображения для табличного преобразования и вывода значения частоты на дисплей.

ГСА этого модуля приведена на рисунке 4.18.

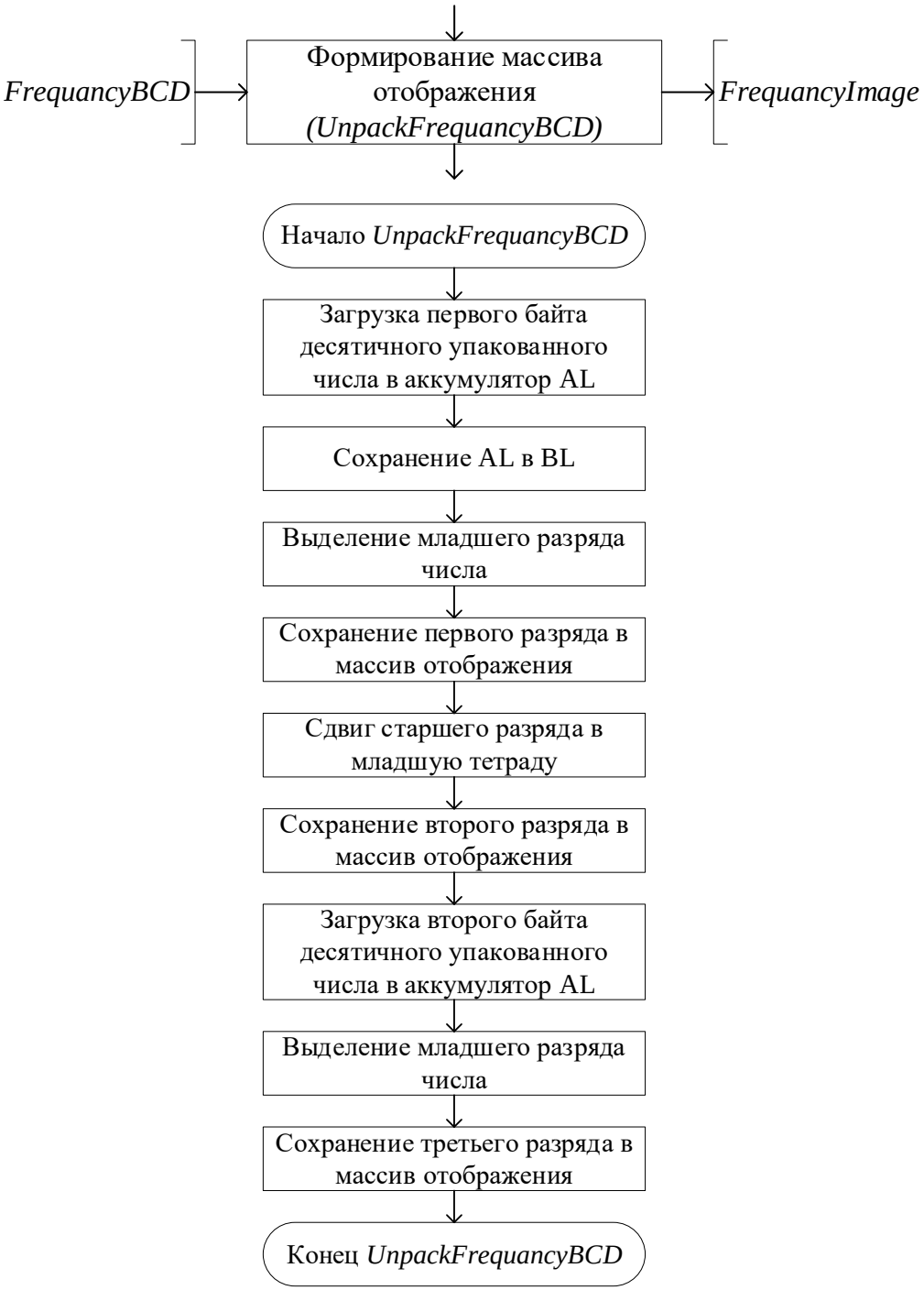


Рисунок 4.18 – Алгоритм модуля «Формирование массива отображения»

Модуль «Формирование изображения импульсов» (PulseImageForming)

Этот модуль служит для формирования изображения импульсов из заданных ранее параметров генерации. Для удобства этот модуль разбит на ряд подмодулей.

Для отображения сгенерированных импульсов используется совокупность матричных индикаторов, которые имитируют осциллограф, что накладывает особенность на формирование изображения импульсов. Выделяется специальный указатель, который будет перемещаться по ячейкам матрицы за счёт сдвигов и модификации адреса массива отображения импульсов с последующим логическим сложением с этим массивом. Таким образом складывается изображение импульсов из заданных параметров генерации.

ГСА этого модуля приведена на рисунке 4.19.

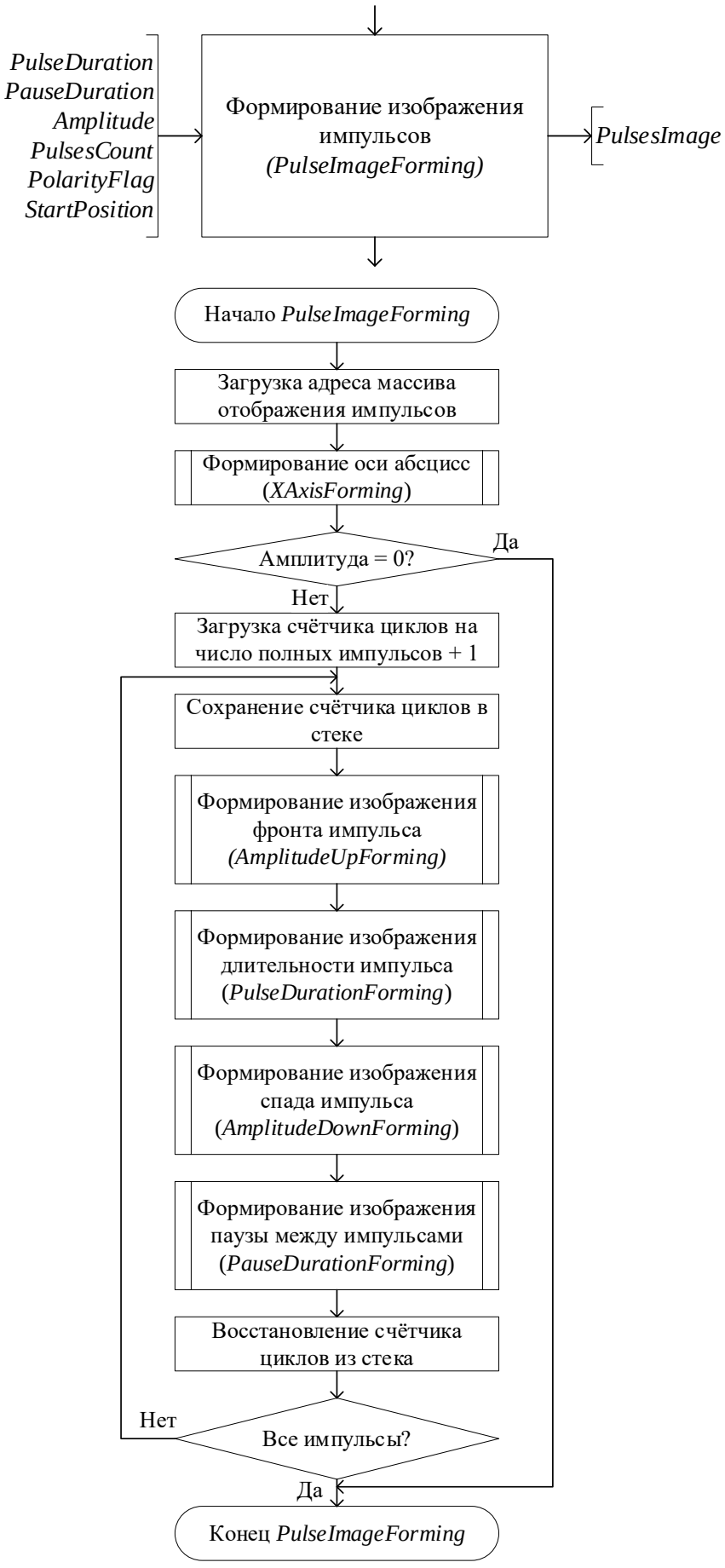


Рисунок 4.19 – Алгоритм модуля «Формирование изображения импульсов»

Подмодуль «Формирование оси абсцисс» (XAxisForming)

Этот подмодуль служит для формирования оси абсцисс в зависимости от параметров полярности импульсов и амплитуды.

ГСА этого модуля приведена на рисунке 4.20.

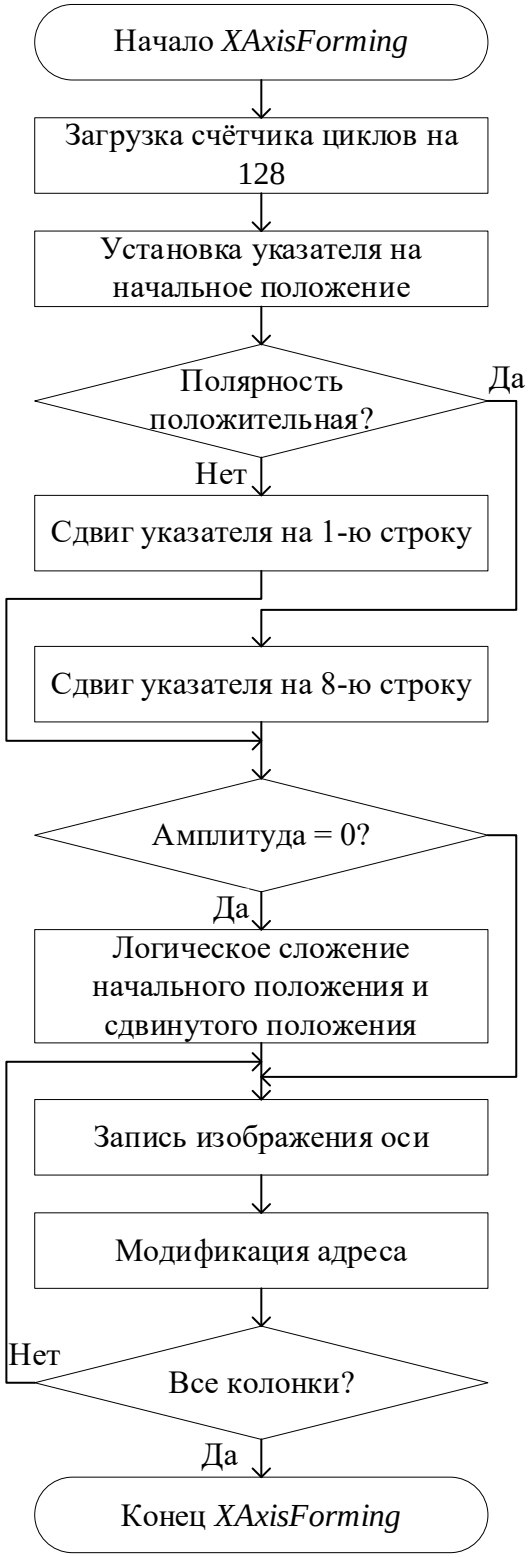


Рисунок 4.20 – Алгоритм подмодуля «Формирование оси абсцисс»

Подмодуль «Формирование изображения фронта импульса»

(AmplitudeUpForming)

Этот подмодуль служит для формирования изображения переднего фронта импульса в зависимости от параметров полярности импульсов и амплитуды.

ГСА этого модуля приведена на рисунке 4.21.

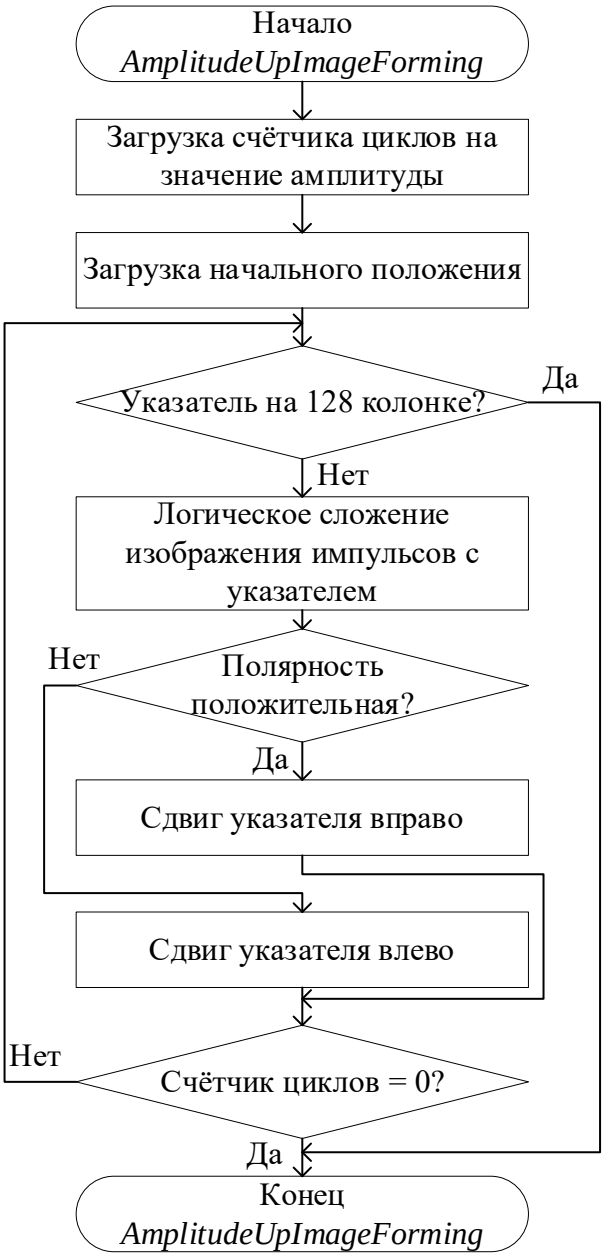


Рисунок 4.21 – Алгоритм подмодуля «Формирование изображения фронта импульса»

Подмодуль «Формирование изображения длительности импульса»

(PulseDurationForming)

Этот подмодуль служит для формирования изображения длительности импульса в зависимости от параметра длительности импульса.

ГСА этого модуля приведена на рисунке 4.22.

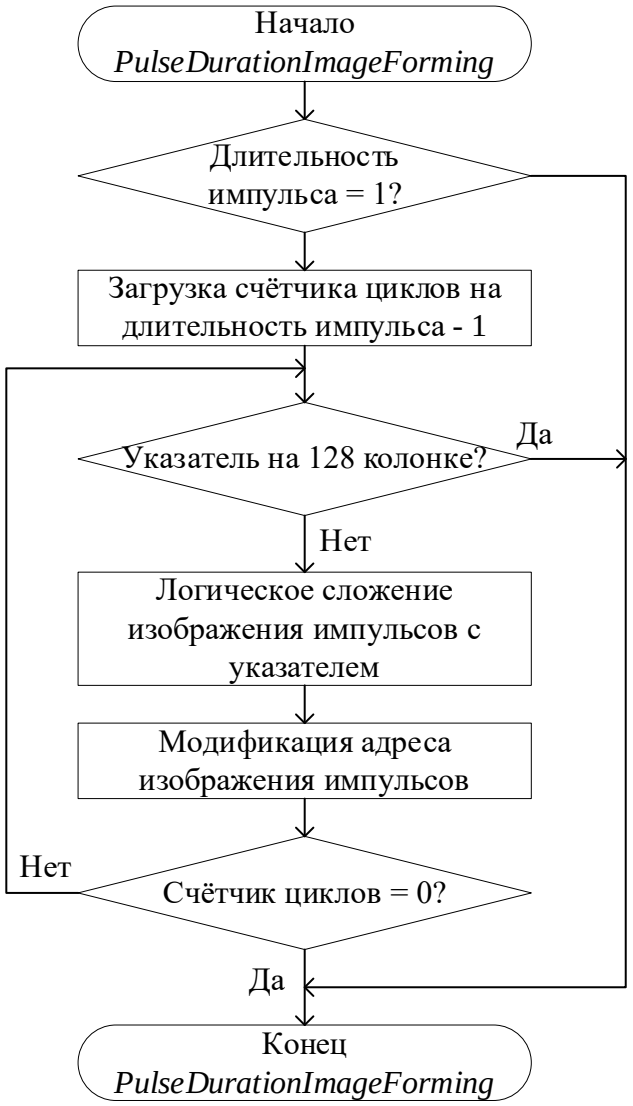


Рисунок 4.22 – Алгоритм подмодуля «Формирование изображения длительности импульса»

Подмодуль «Формирование изображения среза импульса»

(AmplitudeDownForming)

Этот подмодуль служит для формирования изображения среза импульса в зависимости от параметров полярности импульсов и амплитуды.

ГСА этого модуля приведена на рисунке 4.23.

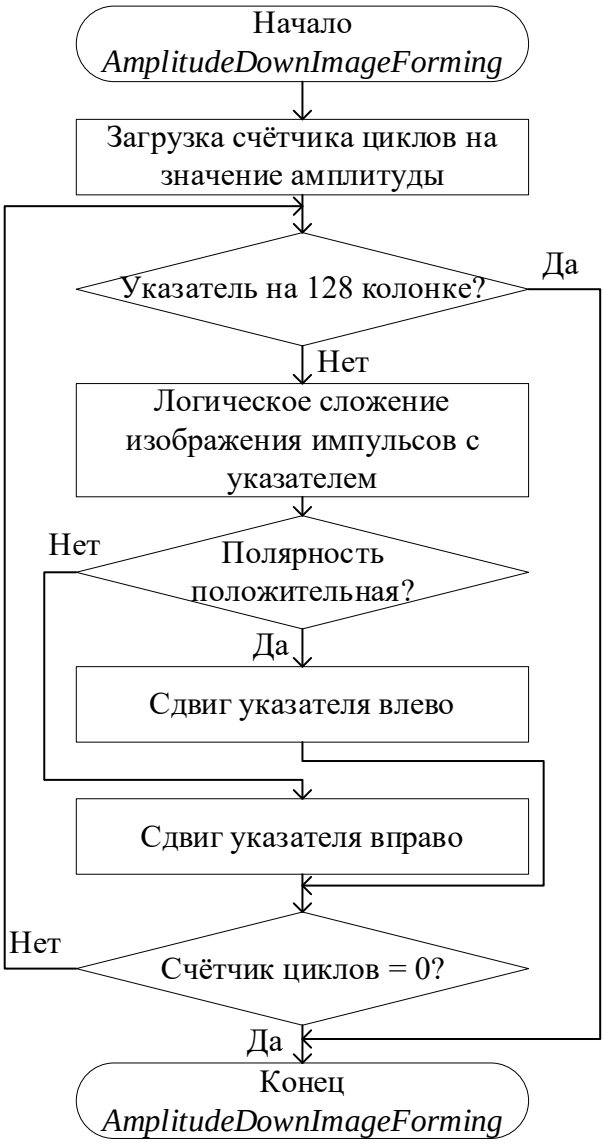


Рисунок 4.23 – Алгоритм подмодуля «Формирование изображения среза импульса»

Подмодуль «Формирование изображения паузы между импульсами» (PauseDurationForming)

Этот подмодуль служит для формирования изображения паузы между импульса в зависимости от параметра паузы между импульсами.

ГСА этого модуля приведена на рисунке 4.24.

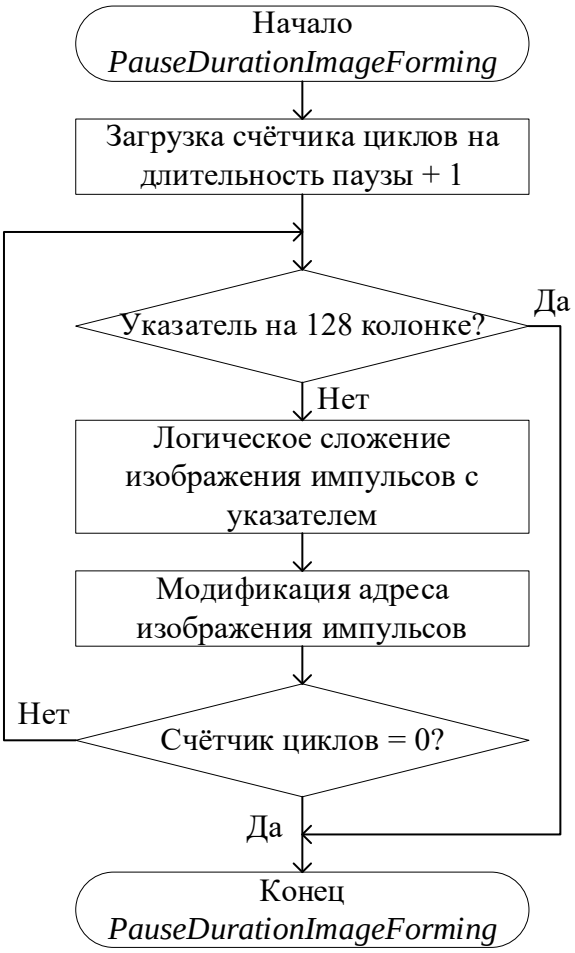


Рисунок 4.24 – Алгоритм подмодуля «Формирование изображения паузы между импульсами»

Модуль «Вывод числовой информации» (DisplayData)

Этот модуль служит для вывода заданных параметров генерации на соответствующие дисплеи. Для удобства этот модуль разбит на ряд подмодулей.

ГСА этого модуля приведена на рисунке 4.25.

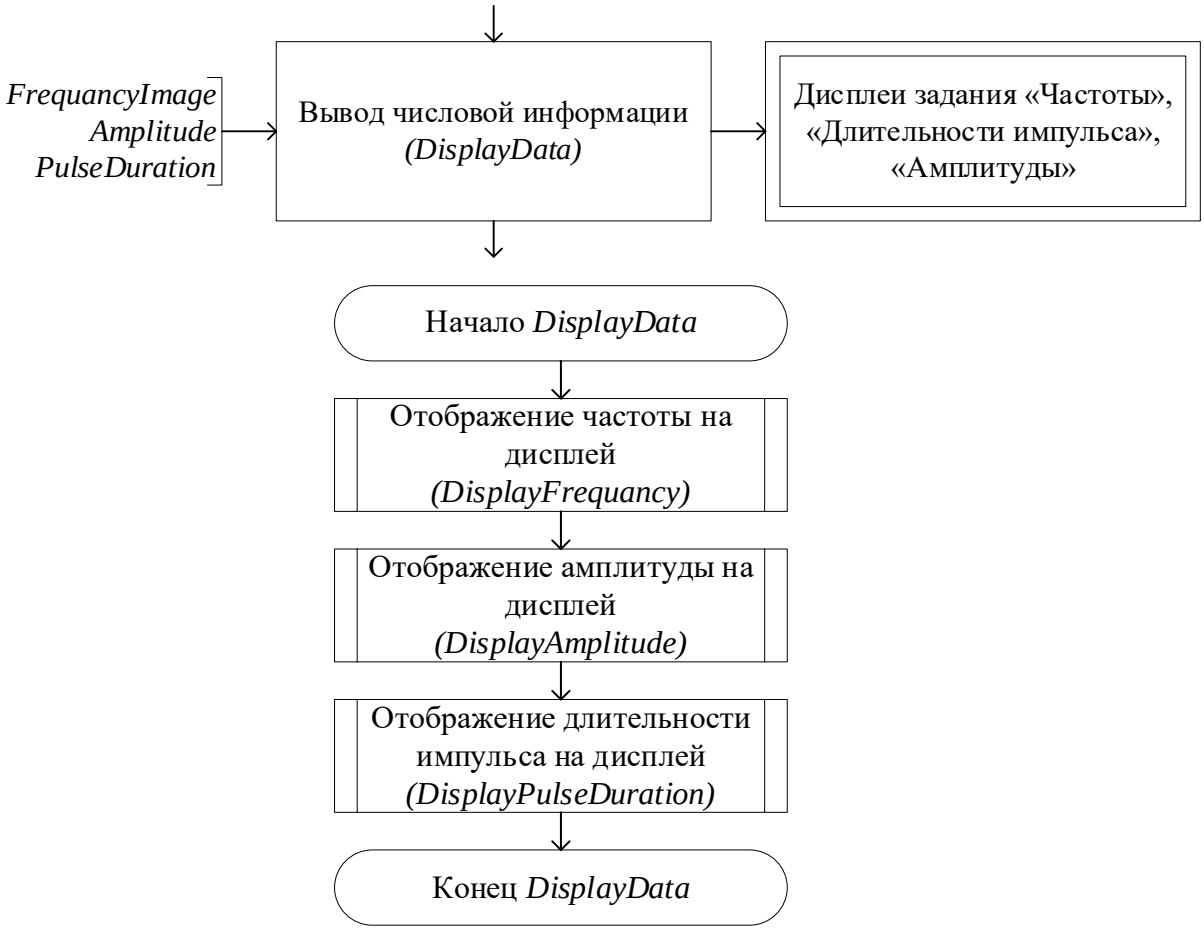


Рисунок 4.25 – Алгоритм модуля «Вывод числовой информации»

Подмодуль «Отображение частоты на дисплей» (DisplayFrequancy)

Этот подмодуль служит для отображения заданной частоты импульсов на соответствующем дисплее. Отображение осуществляется подачей питания на соответствующий индикатор с выводом образа значения параметра на данный индикатор.

ГСА этого модуля приведена на рисунке 4.26.

Рисунок 4.26 – Алгоритм подмодуля «Отображение частоты на дисплей»

Подмодуль «Отображение амплитуды на дисплей» (DisplayAmplitude)

Этот подмодуль служит для отображения заданной амплитуды импульсов на соответствующем дисплее. Отображение осуществляется подачей питания на соответствующий индикатор с выводом образа значения параметра на данный индикатор.

ГСА этого модуля приведена на рисунке 4.27.

Рисунок 4.27 – Алгоритм подмодуля «Отображение амплитуды на дисплей»

Подмодуль «Отображение длительности импульсов на дисплей» (DisplayPulseDuration)

Этот подмодуль служит для отображения заданной длительности импульсов на соответствующем дисплее. Отображение осуществляется подачей питания на соответствующий индикатор с выводом образа значения параметра на данный индикатор.

ГСА этого модуля приведена на рисунке 4.28.

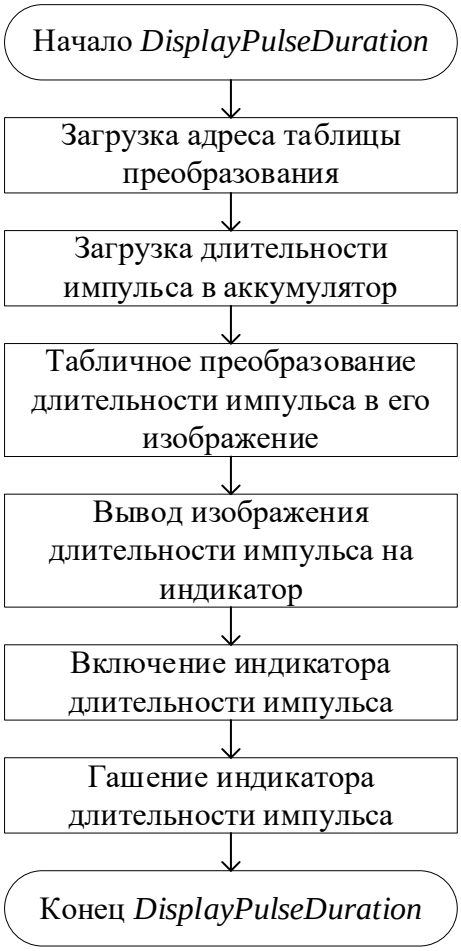


Рисунок 4.28 – Алгоритм подмодуля «Отображение длительности импульсов на дисплей»

Модуль «Вывод изображения импульсов на матрицу» (MatrixOutput)

Этот модуль служит для вывода сформированных ранее изображений импульсов соответствующий дисплей, который имитирует осциллограф. Отображение осуществляется активацией строк образом импульса и активацией колонки, соответствующей данному образу, с подачей питания на соответствующую матрицу.

ГСА этого модуля приведена на рисунке 4.29.

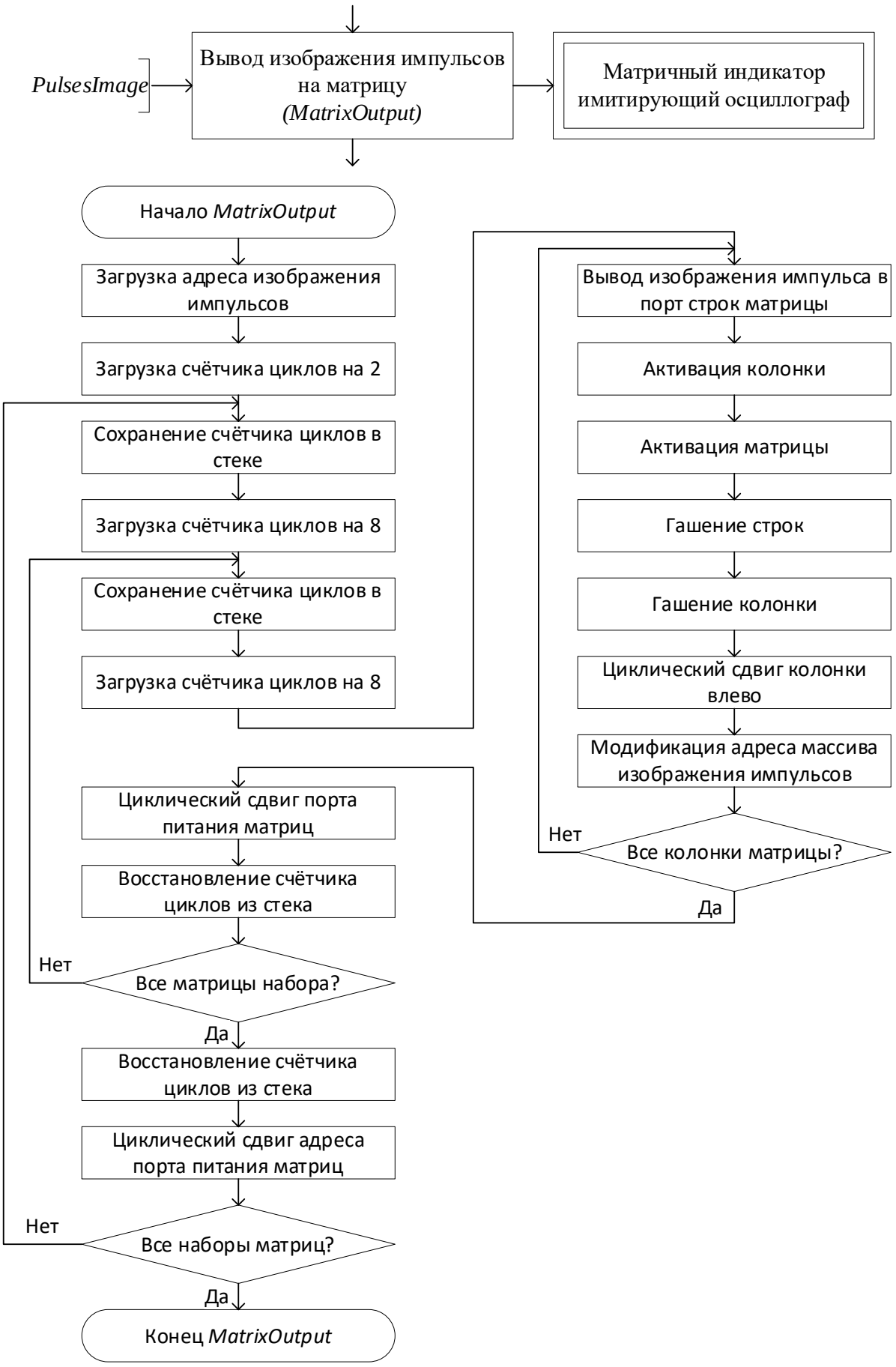


Рисунок 4.29 – Алгоритм модуля «Вывод изображения импульсов на матрицу»

Модуль «Функциональная подготовка» (Initialization)

Этот модуль предназначен для начальной установки некоторых наборов данных или служебных ячеек, которые служат для хранения вспомогательной информации. Для удобства выделен один подмодуль.

ГСА этого модуля приведена на рисунке 4.30.

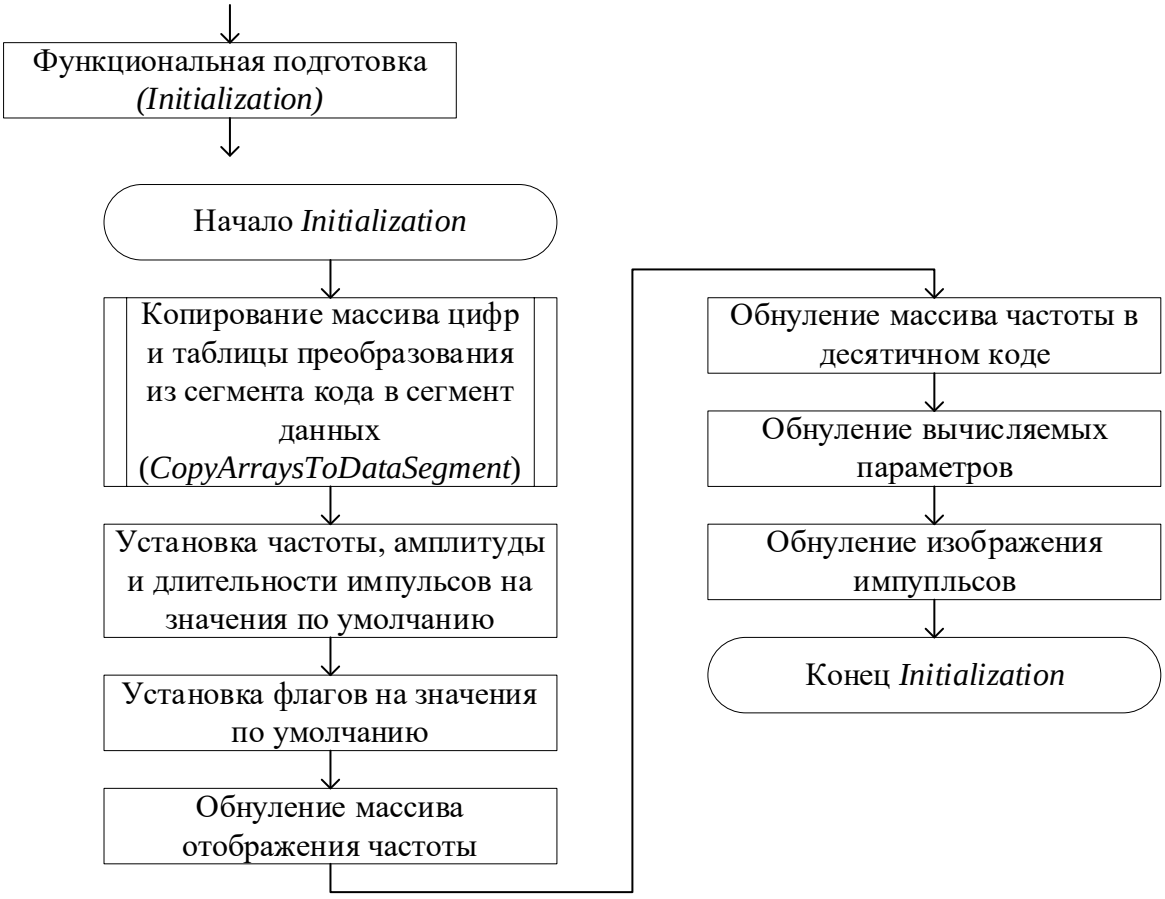


Рисунок 4.30 – Алгоритм модуля «Функциональная подготовка»

Подмодуль «Копирование массива цифр и таблицы преобразования из сегмента кода в сегмент данных» (CopyArraysToDataSegment)

Этот подмодуль служит для копирования данных, необходимых для табличного преобразования значений частоты, амплитуды и длительности импульса в их изображения, которые нужны для вывода на дисплеи.

ГСА этого модуля приведена на рисунке 4.31.

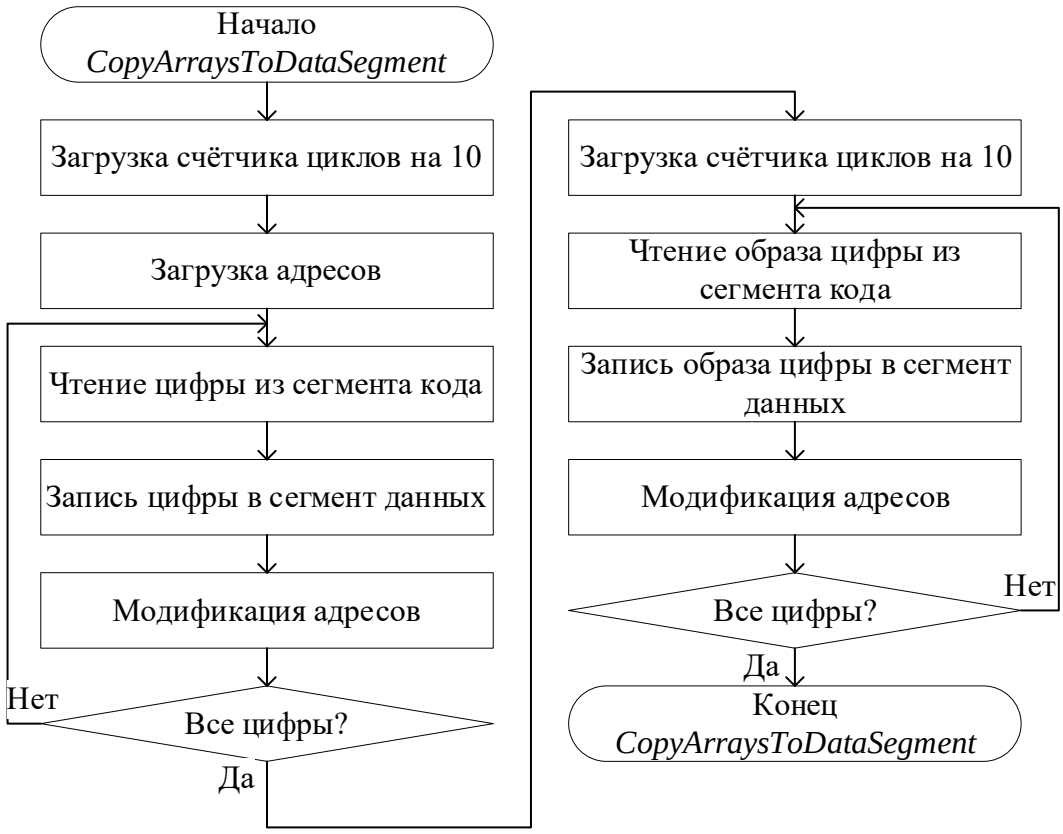


Рисунок 4.31 – Алгоритм подмодуля «Копирование массива цифр и таблицы преобразования из сегмента кода в сегмент данных»

5 Кодирование программы

Кодирование программы представляет собой запись алгоритмов проектируемой программы на языке ассемблер.

При кодировании программы используются ранее разработанные структура данных (таблица 3.1) и ГСА (рисунки 4.1 – 4.31).

Для полного кодирования программы остается лишь определить таблицу преобразования кодов десятичных цифр из массива отображения в семисегментные коды, которые выводятся на семисегментные знакосинтезирующие индикаторы для отображения соответствующей цифры.

Эта таблица зависит от выбранного варианта подключения сегментов индикатора к разрядам управляющего порта вывода. Для принятого варианта зажигания индикаторов нулём преобразование должно выполняться по таблице 5.1.

Исходный текст проектируемой программы приведен в Приложении А.

Таблица 5.1 – Кодирования кодов

Графический знак	Двоично-десятичный код				Семисегментный код								Hex
					D7	*D*6	*D*5	*D*4	*D*3	*D*2	*D*1	*D*0	
	*D*3	*D*2	*D*1	*D*0	*H*	*G*	*F*	*E*	*D*	*C*	*B*	*A*	
“0”	0	0	0	0	1	1	0	0	0	0	0	0	C0h
“1”	0	0	0	1	1	1	1	1	0	0	1	1	F3h
“2”	0	0	1	0	1	0	0	0	1	0	0	1	89h
“3”	0	0	1	1	1	0	1	0	0	0	0	1	A1h
“4”	0	1	0	0	1	0	1	1	0	0	1	0	B2h
“5”	0	1	0	1	1	0	1	0	0	1	0	0	A4h
“6”	0	1	1	0	1	0	0	0	0	1	0	0	84h
“7”	0	1	1	1	1	1	1	1	0	0	1	1	F1h
“8”	1	0	0	0	1	0	0	0	0	0	0	0	80h
“9”	1	0	0	1	1	0	1	0	0	0	0	0	A0h

6 Тестирование и отладка программы

Для тестирования и отладки разработанной программы будет использоваться интегрированная программная среда *Design Microsystem*. Эта среда позволяет на архитектурном уровне представить проектируемое устройство и работать с ним как с реальным устройством.

В состав архитектуры интерфейса устройства входят 1 порт ввода, 7 кнопок без фиксации для задания параметров генерации, 1 порт вывода для активации строк матриц, 1 порт вывода для активации колонок матриц, 2 порта вывода для подачи питания на матрицы, 1 порт вывода для подачи питания на знакосинтезирующие семисегментные индикаторы, 1 порт вывода для активации сегментов индикаторов, 16 матричных индикаторов для вывода на них изображения импульсов.

Для составления архитектуры устройства выберем необходимые интерфейсные элементы и разместим их на экране, соединяя в соответствии с архитектурой. После этого установим интерфейсным элементам необходимые свойства:

– для портов ввода/вывода зададим адреса в соответствии с исходным текстом программы;

– для знакосинтезирующих семисегментных индикаторов выберем уровень зажигания, равный «Зажигается нулём»;

– для кнопок определим исходное состояние, соответствующее ненажатой кнопке, как «нормально незамкнута».

Полученная в среде *Design Microsystem* архитектура генератора импульсов приведена в Приложении Б.

От архитектурного представления устройства легко перейти к его лицевой панели, которая представлена в Приложении В.

Далее с помощью текстового редактора занесем исходный текст программы.

После занесения программы выполним ее ассемблирование и компоновку, то есть сборку программы, а затем выполним ее.

7 Руководство пользователя

При включении устройства происходит первый расчёт всех параметров, формирование выходной информации и непосредственный её вывод на визуальные элементы.

На основе параметров генерации, заданных по умолчанию, происходит формирование импульсов и их отображение на дисплее выше подписи «Матричный индикатор, имитирующий осциллограф».

Задание частоты осуществляется нажатиями на соответствующие кнопки. Они располагаются выше надписи «Частота».

Нажимая на кнопку «+ 10 Гц», частота увеличивается на 10 Гц вплоть до максимального значения, составляющего 150 Гц. После достижения максимального значения реакции на нажатия данной кнопки не будет.

Нажимая на кнопку «– 10 Гц», частота уменьшается на 10 Гц вплоть до минимального значения, составляющего 20 Гц. После достижения минимального значения реакции на нажатия данной кнопки не будет.

Текущее состояние заданной частоты отображается на соответствующем дисплее над кнопками, что находятся выше надписи «Частота».

Задание амплитуды осуществляется нажатиями на соответствующие кнопки. Они располагаются выше надписи «Амплитуда».

Нажимая на кнопку «+ 1 В», частота увеличивается на 1 В вплоть до максимального значения, составляющего 5 В. После достижения максимального значения реакции на нажатия данной кнопки не будет.

Нажимая на кнопку «– 1 В», частота уменьшается на 1 В вплоть до минимального значения, составляющего 0 В. После достижения минимального значения реакции на нажатия данной кнопки не будет.

Текущее состояние заданной амплитуды отображается на соответствующем дисплее над кнопками, что находятся выше надписи «Амплитуда».

Задание длительности импульсов осуществляется нажатиями на соответствующие кнопки. Они располагаются выше надписи «Длительность импульса».

Нажимая на кнопку «+ 1 мс», частота увеличивается на 1 мс вплоть до максимального значения, составляющего 5 мс. После достижения максимального значения реакции на нажатия данной кнопки не будет.

Нажимая на кнопку «– 1 мс», частота уменьшается на 1 мс вплоть до минимального значения, составляющего 1 мс. После достижения минимального значения реакции на нажатия данной кнопки не будет.

Текущее состояние заданной длительности импульсов отображается на соответствующем дисплее над кнопками, что находятся выше надписи «Длительность импульса».

Задание полярности импульсов осуществляется нажатием на соответствующую кнопку. Она располагаются выше надписи «Положительная/отрицательная полярность».

Нажимая на кнопку изменения полярности, полярность импульсов изменится на противоположную. При этом произойдёт зеркальное отражение изображения импульсов на соответствующем дисплее, что находится выше надписи «Матричный индикатор, имитирующий осциллограф».

При изменении любого из параметров происходит формирование импульсов в соответствии с заданными значениями и эти импульсы отображаются на дисплее, что находится выше надписи «Матричный индикатор, имитирующий осциллограф».

Если параметры не изменяются, то формирование новых импульсов не произойдёт и никаких изменений на дисплее, что находится выше надписи «Матричный индикатор, имитирующий осциллограф», не будет.

Заключение

В ходе выполнения курсового проекта было разработано устройство «Генератор импульсов». Из анализа технического задания была произведена постановка задачи. Далее была проделана декомпозиция поставленной задачи и разработана структура данных. После этого выполнена алгоритмизация программы и ее отдельных частей, написан текст разрабатываемой программы.

При выполнении поставленной задачи были соблюдены все требования технического задания.

Список использованных источников

1. Комаров В. М. Микропроцессорные системы: Выполнение курсового проекта. Учебное пособие /РГАТУ.-Рыбинск, 2015. – 93 с.

2 Комаров В. М. Микропроцессорные системы. Проектирование аппаратного и программного обеспечения: Учебное пособие. – 2 изд. перераб. и доп. – Рыбинск, РГАТА, 2004. – 176 с.

Приложение А

Исходный текст программы «Генератор импульсов»

```
.386
;Задайте объём ПЗУ в байтах
RomSize    EQU   4096
                              MatrixPowerPortL = 0FBh; Матрицы 0-7
                              MatrixPowerPortH = 0F7h; Матрицы 8-15
                              MatrixColumnPort = 0FEh;
                              MatrixRowPort = 0FDh;
                              DisplayPowerPort = 0EFh ;0-2 Частота, 3 Амплитуда, 4 Длительность импульса
                              DisplaySegmentsPort = 0DFh
                              KeyboardPort = 0FEh
                              MS = 1000
                              NMax = 50
IntTable   SEGMENT use16 AT 0
;Здесь размещаются адреса обработчиков прерываний
IntTable   ENDS

Data       SEGMENT use16 AT 40h
;Здесь размещаются описания переменных

                              FrequancyBCD db 2 dup(?) ;Частота в десятичном коде
                              FrequancyImage db 3 dup(?) ;Частота в десятичном коде в распакованном
формате
                              Frequancy db ? ;Частота
                              Amplitude db ? ;Амплитуда
                              PulseDuration db ? ;Длительность импульса
                              PauseDuration db ? ;Длительность паузы
                              PulsePeriod db ? ;Период
                              NoInputErrorFlag db ? ;Фгал неактивных кнопок
                              PolarityFlag db ? ;Флаг полярности
                              DataHexArr db 10 dup(?)
                              DataHexTabl db 10 dup(?)
                              KeyImage db ? ; FF: ничего, FE: + Freq, FD: - Freq, FB: + Ampl, F7: - Ampl, EF: +
PulDur, DF: - PulDur, BF: Generation, 7F: Polarity
                              OldButton db ? ; Предыдущее состояние кнопки
                              PulsesImage db 128 dup(?) ;Массив отображения импульсов
                              PulsesCount db ? ;Количесвто импульсов
                              StartPosition db ? ;Начальная позиция указателя
Data       ENDS
;Задайте необходимый адрес стека
Stk                           SEGMENT use16  AT 00FFh
;Задайте необходимый размер стека
          dw    16 dup (?)
StackTop     Label Word
Stk        ENDS
InitData   SEGMENT use16
InitDataStart:
;Здесь размещаются описания констант
InitDataEnd:
InitData   ENDS

Code       SEGMENT use16
;Здесь размещаются описания констант

                              ASSUME cs:Code,ds:Data,es:Data, ss:Stk
                              HexArr DB 00h,01h,02h,03h,04h,05h,06h,07h,08h,09h
                              HexTabl DB 0C0h,0F3h,89h,0A1h,0B2h,0A4h,84h,0F1h,80h,0A0h
Initialization PROC ;Функциональная подготовка
                              CALL CopyArraysToDataSegment
```

```
                        XOR AX, AX
                        MOV Frequancy, 20
                        MOV Amplitude, 5;
                        MOV PulseDuration, 5
                        MOV KeyImage, 0FFh
                        MOV NoInputErrorFlag, 0FFh
                        MOV PolarityFlag, AH
                        MOV FrequancyImage+0, AH
                        MOV FrequancyImage+1, AH
                        MOV FrequancyImage+2, AH
                        MOV FrequancyBCD+0, AH
                        MOV FrequancyBCD+1, AH
                        MOV FrequancyBCD+2, AH
                        MOV PulsePeriod, AH
                        MOV PulsesCount, AH
                        MOV PauseDuration, AH
                        MOV OldButton, AH
                        LEA DI, PulsesImage
                        MOV CX, 128
        M1:             MOV [DI], AL
                        INC DI
                        LOOP M1

                        RET
        Initialization ENDP
        CopyArraysToDataSegment PROC
                        MOV CX, 10 ;Загрузка счётчика циклов
                        LEA BX, HexArr ;Загрузка адреса массива цифр
                        LEA BP, HexTabl ;Загрузка адреса таблицы преобразования
                        LEA DI, DataHexArr ;Загрузка адреса массива цифр в сегменте данных
                        LEA SI, DataHexTabl ;Загрузка адреса таблицы преобразования в сегменте
данных
        M0:             MOV AL, CS:[BX] ;Чтение цифры из массива в аккумулятор
                        MOV [DI], AL ;Запись цифры в сегмент данных/DataHexArr
                        INC BX ;Модификация адреса HexArr
                        INC DI ;Модификация адреса DataHexArr
                        LOOP M0
                        MOV CX, 10 ;Загрузка счётчика циклов
        M1:             MOV AH, CS:[BP] ;Чтение графического образа из таблицы преобразования
                        MOV [SI], AH ;Запись графического образа в сегмент данных/DataHexTabl
                        INC BP ;Модификация адреса HexTabl
                        INC SI ;Модификация адреса DataHexTabl
                        LOOP M1
                        XOR BP,BP
                        RET
        CopyArraysToDataSegment ENDP

        KeyRead   PROC  ;Чтение кнопок
                        MOV DX, KeyboardPort
                         IN AL, KeyboardPort
                        CALL VibrDestr
                        MOV AH, AL
                        XOR AL, OldButton
                        AND AL, AH
                        NOT AL
                        MOV OldButton, AH
                        MOV KeyImage, AL
                        RET
        KeyRead   ENDP

        VibrDestr PROC
        VD1:             mov  ah,al     ;Сохранение исходного состояния
```

```
                    mov   bh,0        ;Сброс счётчика повторений
VD2:                in    al,dx       ;Ввод текущего состояния
                    cmp   ah,al       ;Текущее состояние=исходному?
                    jne   VD1         ;Переход, если нет
                  inc   bh          ;Инкремент счётчика повторений
                  cmp   bh,NMax     ;Конец дребезга?
                  jne   VD2         ;Переход, если нет
                  mov   al,ah       ;Восстановление местоположения данных
                  ret
VibrDestr  ENDP

KeyCheck PROC
                    CMP KeyImage, 0FFh
                    JNZ M1
                    MOV NoInputErrorFlag, 0FFh
                    JMP M2
M1:                 MOV NoInputErrorFlag, 00h
M2:                 RET
KeyCheck ENDP

DataSetting PROC
                    CMP NoInputErrorFlag, 0FFh
                    JZ M1
                    CALL FrequancyAddition
                    CALL FrequancySubtraction
                    CALL AmplitudeAddition
                    CALL AmplitudeSubtraction
                    CALL PulseDurationAddition
                    CALL PulseDurationSubtraction
                    CALL PolaritySetting
                    CALL PulsePeriodAndPauseCalculation
                    CALL PulsesCountCalculation
                    CALL StartPositionSetting
M1:                 RET
DataSetting ENDP

FrequancyAddition PROC
                    CMP KeyImage, 0FEh
                    JNZ M1
                    CMP Frequancy, 150
                    JZ M1
                    ADD Frequancy, 10
M1:                 RET
FrequancyAddition ENDP

FrequancySubtraction PROC
                    CMP KeyImage, 0FDh
                    JNZ M1
                    CMP Frequancy, 20
                    JZ M1
                    SUB Frequancy, 10
M1:                 RET
FrequancySubtraction ENDP

AmplitudeAddition PROC
                    CMP KeyImage, 0FBh
                    JNZ M1
                    CMP Amplitude, 5
                    JZ M1
                    INC Amplitude
M1:                 RET
AmplitudeAddition ENDP
```

```
AmplitudeSubtraction PROC
                    CMP KeyImage, 0F7h
                    JNZ M1
                    CMP Amplitude, 0
                    JZ M1
                    DEC Amplitude
M1:                 RET
AmplitudeSubtraction ENDP

PulseDurationAddition PROC
                    CMP KeyImage, 0EFh
                    JNZ M1
                    CMP PulseDuration, 5
                    JZ M1
                    INC PulseDuration
M1:                 RET
PulseDurationAddition ENDP

PulseDurationSubtraction PROC
                    CMP KeyImage, 0DFh
                    JNZ M1
                    CMP PulseDuration, 1
                    JZ M1
                    DEC PulseDuration
M1:                 RET
PulseDurationSubtraction ENDP

PolaritySetting PROC
                    CMP KeyImage, 0BFh
                    JNZ M1
                    NOT PolarityFlag
M1:                 RET
PolaritySetting ENDP

PulsePeriodAndPauseCalculation PROC
                    MOV AX, MS
                    DIV Frequancy
                    MOV PulsePeriod, AL
                    SUB AL, PulseDuration
                    MOV PauseDuration, AL
                    RET
PulsePeriodAndPauseCalculation ENDP

PulsesCountCalculation PROC
                    XOR AX, AX
                    MOV AL, LENGTH PulsesImage
                    DIV PulsePeriod
                    MOV PulsesCount, AL
                    RET
PulsesCountCalculation ENDP

StartPositionSetting PROC
                    CMP PolarityFlag, 00h
                    JNZ M1
                    MOV StartPosition, 20h
                    JMP M2
M1:                 MOV StartPosition, 04h
M2:                 RET
StartPositionSetting ENDP

BinaryToBCD PROC
```

```
                XOR BX, BX
                MOV FrequancyBCD+0, BL
                MOV FrequancyBCD+1, BL
                MOV FrequancyBCD+2, BL
                MOV BL, Frequancy
                MOV CX, 8
M2:             LEA DI, FrequancyBCD
                SHL BL, 1
                PUSH CX
                MOV CX, 2
M1:             MOV AL, [DI]
                ADC AL, [DI]
                DAA
                MOV [DI], AL
                INC DI
                LOOP M1
                POP CX
                LOOP M2
                RET
BinaryToBCD ENDP

UnpackFrequancyBCD PROC
                MOV AL, FrequancyBCD+0
                MOV BL, AL
                AND AL, 0Fh
                MOV FrequancyImage+0, AL
                SHR BL,4
                MOV FrequancyImage+1, BL
                MOV AL, FrequancyBCD+1
                AND AL, 0Fh
                MOV FrequancyImage+2, AL
                RET
UnpackFrequancyBCD ENDP

PulseImageForming PROC
                LEA BX, PulsesImage
                XOR DI, DI

                CALL XAxisForming

                CMP Amplitude, 00h
                JZ M2
                XOR DI, DI

                MOV CL, PulsesCount
                INC CL

M1:             PUSH CX

                CALL AmplitudeUpImageForming
                CALL PulseDurationImageForming
                CALL AmplitudeDownImageForming
                CALL PauseDurationImageForming
                POP CX
                LOOP M1
M2:             RET
PulseImageForming ENDP

XAxisForming PROC
                MOV CL, 128
                MOV AH, StartPosition
                MOV AL, AH
```

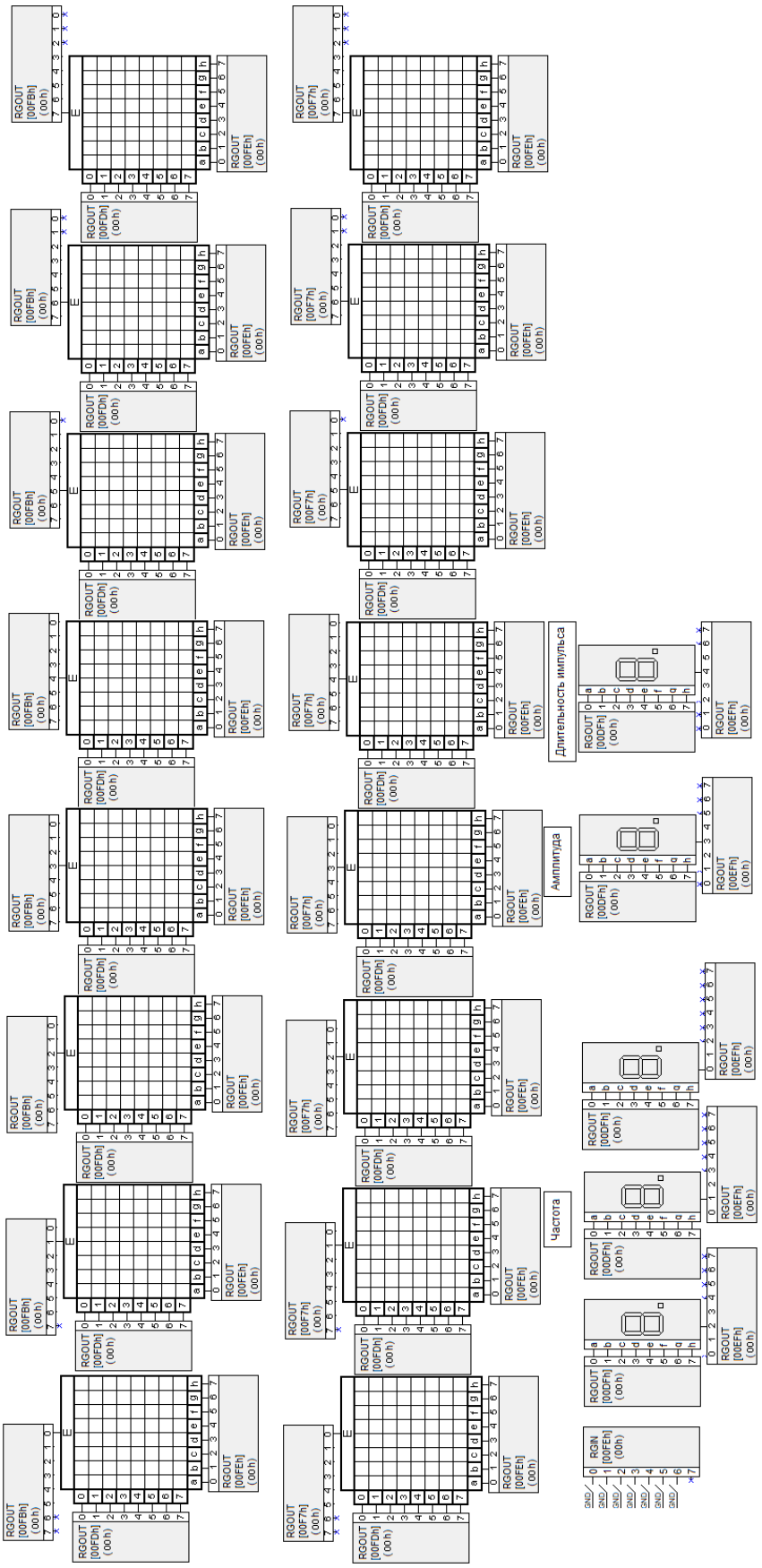
```
                    CMP PolarityFlag, 0FFh
                    JNZ M1
                    SHR AH, 2
M3:                 CMP Amplitude, 00h
                    JNZ M2
                    OR AH, AL
                    JMP M2
M1:                 SHL AH, 2
                    JMP M3
                    ;CMP Amplitude, 00h
                    ;JNZ M2
                    OR AH, AL
M2:                 OR [BX+DI], AH
                    AND [BX+DI], AH
                    INC DI
                    LOOP M2
                    RET
XAxisForming ENDP

AmplitudeUpImageForming PROC
                    MOV CL, Amplitude
                    MOV AH, StartPosition
M1:                 CMP DI, 128
                    JZ M7
                    OR [BX+DI], AH
                    CMP PolarityFlag, 00h
                    JNZ M2
                    SHR AH, 1
                    JMP M3
M2:                 SHL AH, 1
M3:                 LOOP M1
M7:                 RET
AmplitudeUpImageForming ENDP

PulseDurationImageForming PROC
                    CMP PulseDuration, 1
                    JZ M7
                    MOV CL, PulseDuration; Длительность импульса - 1
                    DEC CL
M2:                 CMP DI, 128
                    JZ M7
                    OR [BX+DI], AH
                    INC DI
                    LOOP M2
M7:                 RET
PulseDurationImageForming ENDP

AmplitudeDownImageForming PROC
                    MOV CL, Amplitude
M1:                 CMP DI, 128
                    JZ M7
                    OR [BX+DI], AH
                    CMP PolarityFlag, 00h
                    JNZ M2
                    SHL AH, 1
                    JMP M3
M2:                 SHR AH, 1
M3:                 LOOP M1
M7:                 RET
AmplitudeDownImageForming ENDP

PauseDurationImageForming PROC
```

```
                        MOV CL, PauseDuration
                        INC CL
M4:                     CMP DI, 128
                        JZ M7
                        OR [BX+DI], AH
                        INC DI
                        LOOP M4
M7:                     RET
PauseDurationImageForming ENDP

DisplayData PROC
                        CALL DisplayFrequancy
                        CALL DisplayAmplitude
                        CALL DisplayPulseDuration
                        RET
DisplayData ENDP

DisplayFrequancy    PROC  ;Вывод частоты на дисплей
                        LEA BX, DataHexTabl
                       MOV AH, FrequancyImage+0
                       MOV AL, AH            ;теперь в al старшая цифра
                       XLAT                        ;табличное преобразование старшей цифры
                       OUT DisplaySegmentsPort, AL    ;выводим на страший индикатор
                       MOV AL, 1
                       OUT DisplayPowerPort, AL    ;зажигаем старший индикатор
                       MOV AL,00h
                       OUT DisplayPowerPort, AL    ;гасим индикатор
                         MOV AH, FrequancyImage+1      ;загружаем в регистры
                       MOV AL, AH            ;текущее значение суммы
                       XLAT                        ;табличное преобразование младшей цифры
                       OUT DisplaySegmentsPort, AL    ;Выводим на младший индикатор
                       MOV AL, 2
                       OUT DisplayPowerPort, AL    ;зажигаем младший индикатор
                       MOV AL,00h
                       OUT DisplayPowerPort, AL    ;гасим индикатор
                         MOV AH, FrequancyImage+2      ;загружаем в регистры
                       MOV AL, AH            ;текущее значение суммы
                       XLAT                        ;табличное преобразование младшей цифры
                       OUT DisplaySegmentsPort, AL    ;Выводим на младший индикатор
                       MOV AL, 4
                       OUT DisplayPowerPort, AL    ;зажигаем младший индикатор
                       MOV AL,00h
                       OUT DisplayPowerPort, AL    ;гасим индикатор
        RET
DisplayFrequancy    ENDP

DisplayAmplitude    PROC  ;Вывод амплитуды на дисплей
                        LEA BX, DataHexTabl
                       MOV AL, Amplitude
                       XLAT                              ;табличное преобразование старшей цифры
                       OUT DisplaySegmentsPort, AL    ;выводим на страший индикатор
                       MOV AL, 8
                       OUT DisplayPowerPort, AL    ;зажигаем старший индикатор
                       MOV AL,00h
                       OUT DisplayPowerPort, AL    ;гасим индикатор
                        RET
DisplayAmplitude    ENDP

DisplayPulseDuration    PROC  ;Вывод длительности импульса на дисплей
                        LEA BX, DataHexTabl
                        MOV AL, PulseDuration
                        XLAT                ;табличное преобразование старшей цифры
```

```
                        OUT DisplaySegmentsPort, AL    ;выводим на страший индикатор
                        MOV AL, 16
                        OUT DisplayPowerPort, AL    ;зажигаем старший индикатор
                        MOV AL,00h
                        OUT DisplayPowerPort, AL    ;гасим индикатор
                        RET
DisplayPulseDuration    ENDP

MatrixOutput PROC
                        LEA SI, PulsesImage
                        MOV DL, MatrixPowerPortL
                        MOV BL, 1
                        MOV AH, 1
                        MOV CX, 2
M3:                     PUSH CX
                        MOV CX, 8
M2:                     PUSH CX
                        MOV CX, 8
M1:                     MOV AL, [SI]
                        OUT MatrixRowPort, AL
                        MOV AL, AH
                        OUT MatrixColumnPort, AL
                        MOV AL, BL
                        OUT DX, AL
                        MOV AL, 0
                        OUT MatrixRowPort, AL
                        OUT MatrixColumnPort, AL
                        OUT DX, AL
                        ROL AH, 1
                        INC SI
                        LOOP M1
                        ROL BL, 1
                        POP CX
                        LOOP M2
                        POP CX
                        ROL DL, 1
                        LOOP M3
                        RET
MatrixOutput ENDP
Start:                  mov   ax,Data
                        mov   ds,ax
                        mov   es,ax
                        mov   ax,Stk
                        mov   ss,ax
                        lea   sp,StackTop
;Здесь размещается код программы
                        CALL Initialization
ILOOP:                  CALL KeyRead
                        CALL KeyCheck
                        CALL DataSetting
                        CALL BinaryToBCD
                        CALL UnpackFrequancyBCD
                        CALL PulseImageForming
                        CALL DisplayData
                        CALL MatrixOutput
                        JMP ILOOP
;В следующей строке необходимо указать смещение стартовой точки
        org   RomSize-16-((InitDataEnd-InitDataStart+15) AND 0FFF0h)
        ASSUME cs:NOTHING
        jmp   Far Ptr Start
Code    ENDS
END             Start
```

Приложение Б

Архитектура генератора импульсов в среде Demis



Приложение В

Лицевая панель генератора импульсов в среде Demis