МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»
ОБНИНСКИЙ ИНСТИТУТ АТОМНОЙ ЭНЕРГЕТИКИ
Отделение интеллектуальных кибернетических систем

Курсовая работа

«Теория автоматов»

Тема работы:

Спроектировать автомат Мили, управляющий выполнением деления двоичных чисел, представленных в дополнительном коде, в формате с плавающей точкой нормальным алгоритмом, со сдвигом множителя, рассмотрение с младших разрядов. Память автомата реализовать на Т - триггерах.

Выполнил студент 3 курса
группы ИВТ-Б18
Кабак Р. А.

Проверил:
Лискунов Р. Г.

Принял:
Перегуда А. И.

Обнинск, 2020

Содержание работы

1. Введение

1.1. Развернутая постановка задачи

Тема: проектирование цифрового автомата с жесткой логикой, управляющего выполнением деления двоичных чисел, представленных в дополнительном коде, в формате с плавающей точкой нормальным алгоритмом, со сдвигом множителя, рассмотрение с младших разрядов.

Целевая установка: в универсальном базисе логических элементов (И-ИЛИ-НЕ) спроектировать автомат Мили. При реализации памяти автомата используются (Т) триггеры.

Исходные данные: тип структуры операционного автомата, базис логических элементов, тип операции и метод её выполнения, код операндов и результата, тип управляющего автомата, тип элементов памяти.

1.2 Что такое «автомат»?

Поведение любого технического устройства можно описать в терминах некоторых физических переменных:
1) Входные переменные – их значения задаются извне, они не определяются самим устройством, но влияют на его поведение;
2) Выходные переменные – для получения их значений построено само устройство, эти значения определяются как некоторые функции, зависящие от входных переменных;
3) Внутренние переменные – не являются ни входными, ни выходными, но необходимы для описания поведения устройства.

Будем рассматривать такие технические устройства, в которых значения переменных проквантованы. Это означает, что из области значений каждой переменной выделены непересекающиеся интервалы, а значения самой переменной учитываются с точностью до интервала. Подобные устройства называются дискретными устройствами. Проквантованность переменных позволяет изучать дискретные устройства с единой точки зрения. Их непосредственное рассмотрение заменяется анализом абстрактной модели, называемой дискретным (цифровым) автоматом.

Абстрактный автомат – это математическая абстракция, модель дискретного устройства, имеющего один вход, один выход и в каждый момент времени находящегося в одном состоянии из множества возможных. На вход этому устройству поступают символы одного алфавита, на выходе оно выдаёт символы (в общем случае) другого алфавита. Формально определяется как шестёрка
$M=(X ,Y ,Q ,\delta , \lambda , q0)$.
где $X=\{x1 , x2…xn \}$– множество входных значений, $Y=\{ y1 , y2… ym \}$ – множество выходных значений,
$Q=\{q1 , q2…qk \}$ – множество внутренних состояний; где m ,n , k – конечные значения.

Если m ,n , kконечны, то автомат называют конечным. Состояние цифрового автомата определяется состоянием элементов памяти δ – характеристическая функция перехода из одного состояния в другое.λ – характеристическая функции выхода цифрового автомата. q0 – начальное состояние цифрового автомата. Автомат так же называется конечным, если конечны множества X ,Y и Q.

2. Основные понятия

2.1. Структура цифрового автомата

Операционный автомат служит для выполнения собственно набора требуемых операций алгоритма. Управляющий автомат задает последовательность действий по алгоритму в зависимости от условий (которые также формируются операционным автоматом как логические сигналы), т.е. координирует действия узлов операционного автомата. Он вырабатывает в некоторой временной последовательности управляющие сигналы, под действием которых в узлах операционного автомата выполняются требуемые действия, например, установка регистра в некоторое состояние, инвертирование содержимого разрядов регистра, пересылка содержимого одного узла в другой, сдвиг содержимого узла влево, вправо, счет, при котором число в счетчике (регистре) возрастает или убывает на единицу, сложение и т. д.

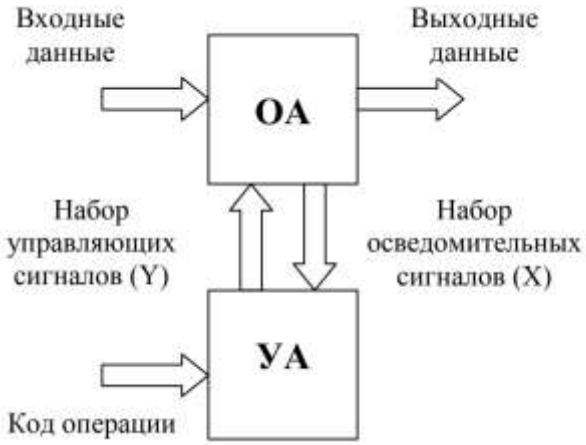


Рисунок 1 – Структура цифрового автомата

Работа автомата разбивается на такты (дискретные интервалы времени). Каждое такое элементарное действие, выполняемое в одном из узлов операционного автомата в течение одного тактового периода, называется микрооперацией. Совокупность микроопераций, которые могут выполняться в операционном автомате параллельно в одном такте, называется микрокомандой. Последовательность микрокоманд, реализующих алгоритм, называется микропрограммой. Таким образом, если в операционном автомате предусматривается возможность исполнения n различных микроопераций, то из управляющего автомата выходят n управляющих цепей, каждая из которых соответствует определенной микрооперации. И если необходимо в операционном автомате выполнить некоторую микрооперацию, достаточно из управляющего автомата по определенной управляющей цепи, соответствующей этой микрооперации, подать сигнал. В силу того, что управляющий автомат определяет микропрограмму, т.е. какие и в какой временной последовательности должны выполняться микрооперации, он получил название микропрограммного автомата.

Формирование управляющих сигналов для выполнения микрокоманд может происходить в зависимости от состояния узлов операционного автомата, определяемого сигналами , которые подаются с соответствующих выходов операционного автомата на входы управляющего автомата. Управляющие сигналы могут также зависеть от внешних сигналов .

Для сокращения числа управляющих цепей, выходящих из управляющего автомата (в тех случаях, когда оно конструктивно выполняется отдельно от операционного), микрокоманды могут кодироваться. Результаты обработки, выполненной в операционном автомате, снимаются с его выходов .

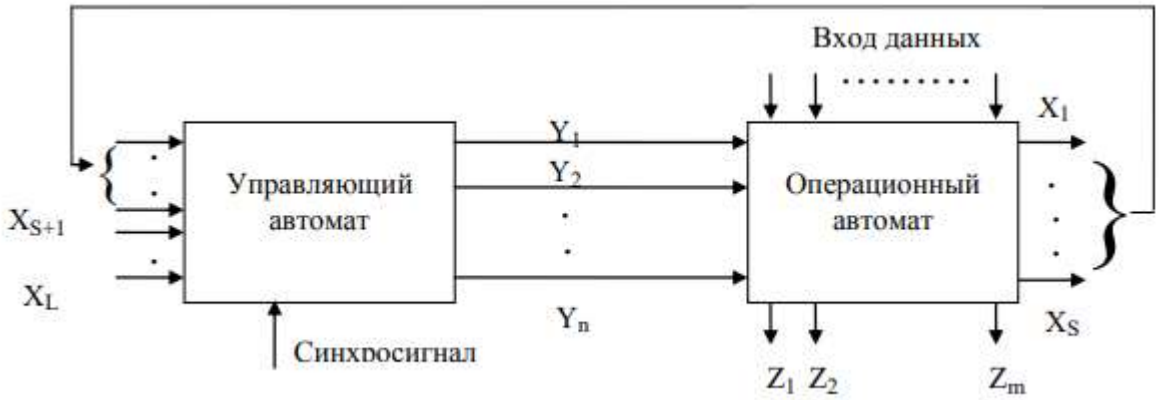


Рисунок 2 – Композиция операционного и управляющего автомата

Таким образом управляющий автомат предназначен для выдачи управляющих сигналов в каждом такте работы цифрового автомата, инициирующих выполнение определенных микроопераций (или микрокоманд) в операционном автомате в соответствии с выполняемым алгоритмом и в зависимости от поступающих на входы управляющего автомата информационных сигналов (условий). Фактически управляющий автомат реализует последовательность действий по алгоритму, при этом содержание этих действий зависит от управляемого объекта, в данном случае – от операционного автомата. Если операционный автомат "знает как" делать, то управляющий автомат "знает, что и когда", то есть в какой последовательности что делать. При этом для управляющего автомата "что делать" – это просто коды команд, про их содержание он не знает.

2.2 Автомат Мили

По закону функционирования или по виду выходной функции ЦА делятся на: автоматы 1-го рода (автоматы Мили) и автоматы 2-го рода (автоматы Мура).
Закон функционирования ЦА первого рода (автомата Мили) есть:

s (t)= δ (s (t-1), x (t)), y (t)= λ (s (t-1), x (t)), где
s (t) - состояние автомата в настоящий момент;
s (t-1)- состояние автомата в предыдущий момент. Если t=0, то s(t-1)=s0;
x (t) - входной сигнал в текущий момент;
δ - оператор формирования данного состояния s;
λ - оператор формирования данного выходного сигнала y.
Т.е., закон функционирования представляет собой совокупность двух функций: функции перехода δ и функции выхода λ, а также, что данное состояние s (t) зависит от предыдущего состояния s (t-1) и входного сигнала данный момент времени, что выходной сигнал в данный момент времени также определяется предыдущим состоянием и входным сигналом в данный
момент времени.

У ЦА Мили выходной сигнал имеется только тогда, когда есть входной сигнал, а у ЦА Мура выходной сигнал имеется всегда.

3. Понятие синтеза УА

3.1 Введение

При синтезе управляющего автомата с жесткой логикой выделяются этапы абстрактного и структурного синтеза. На этапе абстрактного синтеза по алгоритму, заданному на начальном языке строится таблица переходов, записываются системы канонических уравнений и системы выходных функций. На этапе структурного синтеза строится логическая схема управляющего автомата.

3.2 Абстрактный синтез

Получение граф-схемы алгоритма и содержательной граф-схемы алгоритма

Функция управляющего автомата задаётся содержательной граф-схемой алгоритма – ориентированный граф с вершинами четырех классов – начальные (1 исходящая, 0 входящих дуг), конечные (0 исходящих, 1 входящая дуга), операторные (любое число входящих, одна исходящая дуга), условные (любое число входящих, 2 исходящих дуги). Начальная и конечная вершина соответствуют началу и концу микропрограммы. Операторная вершина описывает действия микропрограммы в 1 такте работы устройства. Внутри операторной вершины могут быть указаны только функционально совместимые микрооперации. В условных вершинах указываются логические условия, две исходящих дуги условной вершины соответствуют значениям логического условия.

Кодированную граф-схему алгоритма получают путём замены в содержательной граф-схеме алгоритма микрооператоров (наборов совместимых микроопераций) на коды микрокоманд, а логических условий на их идентификаторы.

Проводится отметка внутренних состояний кодированной граф-схемы алгоритма. Отметка состояний должна соответствовать закону функционирования автомата Мили.

Будем полагать, что автомат начинает работу с состояния , в котором он не вырабатывает никаких выходных сигналов и после выполнения микропрограммы снова оказывается в этом же состоянии. Затем автомат переходит в состояния, предписанные законом функционирования, и формирует микрокоманды , соответствующие текущим значениям условий . Момент окончания выполнения микропрограммы отмечается возвратом автомата в начальное состояние .

Если для интерпретации закодированной ГСА используется автомат Мили, то отметка граф-схемы производится в следующем порядке:

- символом s0 отмечается выход начальной и вход конечной вершины;
- символами s1, s2, ... отмечаются входы вершин, следующие за операторными вершинами;
- входы двух различных вершин не могут быть отмечены одинаковыми символами;
- входы вершины могут отмечаться только одним символом состояния.

Приведенные правила означают, что если вершина имеет несколько входов, то символом состояния отмечается их подмножество, состоящее из входов, следующих только за начальной или за операторными вершинами. Если один из входов конечной вершины соединен с выходом операторной вершины, то между ними необходимо ввести пустую операторную вершину.

Построение графа

Внутренние состояния представляются вершинами графа. Внутренние переходы от одного состояния к другому изображаются направленными дугами. Значение входного сигнала, вызывающего этот переход из текущего состояния в последующее приписывается соответствующей дуге. Таким образом, на графах отображаются обе характеристические функции конечного автомата.

3.2 Структурный синтез управляющего автомата

При заданных типах элементов памяти структурный синтез управляющего автомата сводится к выполнению следующих проектных операций:

- кодирование внутренних состояний;
- формирование функций внешнего перехода;
- формирование и минимизация функций возбуждения элементов памяти и функций выходов;

- построение комбинационной схемы автомата в выбранном базисе логических элементов и функциональной схемы автомата.

3.2.1 Кодирование внутренних состояний

Структурный синтез начинается с двоичного кодирования внутренних состояний автомата - установления взаимно-однозначного соответствия между состояниями автомата и комбинациями состояний элементов памяти.

Анализ структурного синтеза автоматов показывает, что различные варианты кодирования состояний автомата приводят к различным выражениям функций возбуждения и функций выходов, в результате чего оказывается, что сложность комбинационной схемы автомата существенно зависит от выбранного кодирования. Как правило, нахождения вариантов кодирования состояний, которые обеспечивают ослабленную функциональную зависимость
для функций возбуждения, дает более экономичную схему, чем при других
типах кодирования.

При кодировании состояний каждому состоянию устройства должна быть поставлена в соответствие некоторая кодовая комбинация. Число разрядов кода выбирается из следующих соображений: если число состояний равно S, то для обеспечения s кодовых комбинаций требуется k-разрядный код, где k-минимальное целое число, при котором выполняется неравенство $s \leq 2\ k$.
При двоичном кодировании состояний автомата число триггеров в его схеме равно числу разрядов кода и вычисляется по формуле:

$$n = k = \lceil \log_2 S \rceil .$$

где S – число состояний автомата; $\lceil \rceil$ - округление в большую сторону.

Обычно выполняют экономичное кодирование состояний, которое обеспечивает наиболее простую реализацию комбинационной схемы (КС) автомата. Используется метод соседнего кодирования, основанный на поиске соседних состояний и назначении им соседних кодов.

Формирование функций внешнего перехода, возбуждения и выходов и построение функциональной схемы управляющего автомата

Функции внешнего перехода определяют изменение состояний каждого из элементов памяти в зависимости от изменения состояния всех элементов памяти и приходящих на автомат входных сигналов.

Чтобы каждый элемент памяти работал в соответствии со своей функцией внешних переходов, необходимо, чтобы на его входы приходили строго определенные управляющие сигналы. Эти сигналы формируются по логическим выражениям, которые называют функциями возбуждения элементов памяти. Функции возбуждения зависят не только от функции внешних переходов элемента памяти, но и от его собственного, внутреннего функционирования.

Табличные формы представления функций внешнего перехода, функций возбуждения и выходов можно получить непосредственно из таблицы переходов-выходов и таблицы кодов состояний. Для этого символы состояний необходимо заменить соответствующими кодами и установить порядок следования строк и столбцов. Для наглядности выполняемых преобразований строится структурная таблицу автомата, которая заполняется с учетом функционирования заданного элемента памяти.

В структурной таблице автомата отображаются значения функций возбуждения и выходов для всех рабочих наборов. С целью упрощения их аналитического представления выполняют минимизацию. Минимизацию функций целесообразно выполнять по критериям оптимальности совместной минимизации: минимум числа различных термов (конъюнкций или дизъюнкций), используемых для покрытия всех функций системы и минимум рангов этих термов. При этом один и тот же терм может входить в покрытие нескольких функций. На основе полученных функций возбуждения и функций выходов можно построить функциональную схему микропрограммного автомата. На практике чаще всего используют базисы Буля (элементы И, ИЛИ, НЕ), Шеффера (элементы И-НЕ) и Пирса (элементы ИЛИ-НЕ).

4.Неформальное описание алгоритма

Деление двоичных чисел во многом аналогично делению десятичных чисел. Процесс деления состоит в том, что последовательно, разряд за разрядом отыскиваются цифры частного путем подбора с последующим умножением этой цифры на делитель и вычитанием этого произведения из делимого. Деление в системе счисления с основанием 2 сводится к выполнению операций вычитания и сдвига.

Для получения частного от деления двух чисел, представленных в форме с плавающей запятой, необходимо определить не только мантиссу результата, но и его порядок, который вычисляется как разность порядков делимого и делителя.

4.1Алгоритм деления чисел с плавающей запятой

Для типичного алгоритма целочисленного деления делимым является двойное слово, а делителем – одинарное; частное и остаток получаются в виде одинарных слов. При выполнении деления необходимо исключить возможность деления на 0.

Если для представления частного потребуется более одного слова, то имеем переполнение. Поэтому перед выполнением деления необходимо проверить условие – делитель должен быть больше старшего слова делимого.

При делении целых чисел можно использовать алгоритм деления без восстановления остатка и алгоритм с восстановлением остатка, который здесь и рассматривается.

Алгоритм деления представляет собой итерационную процедуру. На каждой итерации производится удвоение делимого (на первой итерации) или остатка (на всех последующих) путем сдвига влево на один разряд, вычитание делителя и определение цифры частного по знаку разности. Если разность положительная, определяемая на данной итерации цифра частного Ci =1, если разность отрицательная – цифра частного Ci=0. Восстановление остатка выполняется путем сложения делителя с остатком после вычитания на текущей итерации деления. Деление выполняется до получения всех цифр частного.

Если исходные операнды заданы в прямых кодах, то путем сложения по модулю 2 знаковых разрядов можно определить знак частного. Модули делимого и делителя можно разделить, используя один из вышеописанных алгоритмов. Для выяснения переполнения необходимо выполнить пробное вычитание A – 2 (n-1)*B, резервируя один разряд n-разрядного частного для знака.

Для деления с плавающей запятой необходимо определить порядок частного Pc. Так как при операции деления порядки чисел вычитаются, то возможно переполнение разрядной сетки в сумматоре порядков. При переполнении в сторону отрицательных величин порядка мантисса результата превращается в машинный нуль, а порядку присваивается наибольшее отрицательное значение.

Алгоритм включает следующие шаги:

1.Операнды извлекаются из ОП с разделением на мантиссу (восстанавливаются скрытые биты) и порядок.

2.Порядок частного получается вычитанием из содержимого регистра порядка делимого содержимого регистра порядка делителя. Если порядок частного больше максимально допустимого, то фиксируется переполнение при делении. Если порядок меньше минимально допустимого порядка, то частное принимается равным 0, фиксируется потеря значимости.

3.Мантисса делимого делится мантиссу делители в дополнительном коде по алгоритму деления дробных чисел в формате с плавающей запятой.

4.При выполнении деления мантисс на шаге пробного вычитания может оказаться, что деление не состоится из-за возникновения переполнения. В этом случае выполняется масштабирование мантиссы делимого, для чего в РСМ восстанавливается исходное значение мантиссу делимого добавлением к РСМ делителя. Затем мантисса делимого уменьшается в два раза путем сдвига вправо на один разряд, при этом порядок делимого увеличивается на 1. Затем выполняется деление мантисс. Если при масштабировании мантиссы делимого и увеличении порядка на единицу оказывается, что $P_x > P_{max}$, то фиксируется переполнение.

5.После завершения деления мантисс производится нормализация частного.

6.При делении чисел с плавающей точкой в процессоре фиксируется только частное.

7.Полученная мантисса частного (со скрытием старших битов) и порядок частного объединяются в формат КВ и записываются в ОП.

5 Функция операционного устройства

Функция операционного устройства задаётся тройкой множеств {D , F , R }. Для записи микропрограммы используем язык функционального микропрограммирования. D и R (множество входных и выходных слов) определяем таблицей спецификации слов, F (множество операций) зададим содержательной граф-схемой алгоритма.

5.1 Спецификация слов микропрограммы

Р1 - регистр, в который записывается делитель;

РСМ - регистр сумматора, в который перед началом деления записываются старшие n разрядов делимого, а затем (в цикле деления) в нем находится остаток от делимого;

Р2 - регистр, в который пред началом деления записываются младшие разряды делимого, а затем (в цикле деления) при сдвигах влево в него последовательно заносятся цифры частного;

СМ - n-разрядный сумматор;

MS - n-разрядный мультиплексор, который используется или для прибавления делителя (MS:=¬P1 и СМ:=СМ+1);

СЧТ используется для подсчета количества циклов деления (работает на вычитание);

ШФ - шинный формирователь, который подключает ШД на передачу или прием к ОП;

дизъюнктор 1 определяет равенство 0 делителя;

дизъюнктор 2 определяет равенство 0 СЧТ;

СФФ - схема формирования флагов;

РФ - регистр флагов.

5.2 Содержательная граф-схема алгоритма микропрограммы

Для графического изображения микропрограммы применяется содержательная граф-схема алгоритма. Для указанной микропрограммы приведена в приложении А (рисунок 3).

6 Функция операционного автомата

Функция операционного автомата задаётся:
1 Спецификация слов микропрограммы (уже была определена ранее).
2 Списком микроопераций.
3 Списком логических условий.

6.1 Список логических условий

Идентефикатор	Логическое условие
x_1	P1=0
x_2	CF
x_3	СЧТ=0

Таблица 1 - логические условия

Идентификаторы логических условий x_i являются идентификаторами осведомительных сигналов.

6.2Список микроопераций

Идентефикатор	Микрооперация
y_1	P1:=ШД
y_2	СЧТ:=n
y_3	OF:=0
y_4	CF:=0
y_5	PCM:=PCM+-P1+1
y_6	(CF, P2):=RC((CF, P2), 1)
y_7	(CF, PCM):=RC((CF, PCM), 1)
y_8	PCM:=PCM+P1+1
y_9	PCM:=PCM-P1+1
y_{10}	СЧТ:=СЧТ-1
y_{11}	PCM:=PCM+P1
y_{12}	РФ:=СФФ
y_{13}	OF:=1

Таблица 2 - микрооперации

Идентефикаторы микроопераций y_i являются идентефикаторами управляющих сигналов.

7. Функция управляющего автомата

Функция управляющего автомата задаётся граф - схемой алгоритма микропрограммы. Она получается путём замены в содержательной граф-схеме алгоритма микроопераций и логических условий на их идентификаторы в соответствии со списком микроопераций и логических условий. Граф-схема для данного алгоритма приведена в приложении А (рисунок 4).

8 Синтез структуры управляющего автомата

8.1 Список переходов управляющего автомата

Множество состояний определим путём разметки граф-схемы в соответствии с введенными правилами. Далее составим список переходов управляющего автомата. Знаком (-) будем обозначать переход без выполнения каких-либо логических условий; x_i - выполнение соответствующего условия; $\ddot{x}_i$ - невыполнение данного условия; q_i - состояние с соответствующим номером.

q_0 (-) q_1
$q_1\ddot{x}_1q_8$,$q_1x_1q_2$
$q_2x_2q_8$, $q_2\ddot{x}_2q_3$
q_3 (-) q_4
$q_4x_2q_5$, $q_4\ddot{x}_2q_5$
q_5 (-) q_6
$q_6x_3q_7$, $q_6\ddot{x}_3q_3$
$q_7x_2q_8$, $q_7\ddot{x}_2q_8$
q_8 (-) q_0

8.2 Граф переходов управляющего автомата

По списку переходов строится граф переходов управляющего автомата. Приведён в приложении А (рисунок 5).

8.3Кодирование состояний автомата и определение сигналов возбуждения

Произведем кодирование состояний. Воспользуемся естественной системой кодирования состояний. Состояния нумеруются с целыми числами, начиная с нуля. Кодом состояния является его номер, записанный в двоичной системе счисления. Число разрядов кода состояния соответствует числу элементов памяти и определяется по формуле: $k = \lceil \log_2 S \rceil$. В нашем случае число состояний S=9. Количество разрядов кода состояния (число элементов памяти): k=4.

q_0	0000
q_1	0001
q_2	0010
q_3	0011
q_4	0100
q_5	0101
q_6	0110
q_7	0111
q_8	1000

Таблица 3 - кодирование состояний

Таким образом, для построения памяти автомата понадобится 4 триггера. По отмеченной граф-схеме алгоритма или графу функционирования автомата, таблицы кодировки состояний автомата и таблицы переходов триггеров строим таблицу функций перехода, возбуждения и выходов устройства (таблица 5).

Количество строк в таблице равно количеству переходов в графе. В столбце «Исходное состояние» записываются состояния, из которых начинается переход, в столбце «Следующее состояние» –состояния, в которые перешел автомат из исходных. В столбце «Выходные сигналы» записываются Y i – вырабатываемые автоматом в следующем состоянии. В столбце «Условия» записываются логические условия (их конъюнкция), обеспечивающие переход из

исходного состояния в следующее. В столбце «Сигналы возбуждения» - элементы триггеров, которые нужны для перехода в следующее состояние.

Q(t)->Q(t+1)	T
0->0	0
0->1	1
1->0	1
1->1	0

Таблица 4 - переходы Т-триггера

Составим структурную таблицу управляющего автомата:

Исходное состояние					Условия перехода	Последующее состояние					Выходные функции	Функции возбуждения			
Метка	код					Метка	код								
	Q_1	Q_2	Q_3	Q_4			Q_1	Q_2	Q_3	Q_4		T_1	T_2	T_3	T_4
q_0	0	0	0	0	-	q_1	0	0	0	1	y_1, y_2, y_3, y_4	0	0	0	1
q_1	0	0	0	1	x_1	q_8	1	0	0	0	y_{13}	1	0	0	1
					$\ddot{x}_1$	q_2	0	0	1	0	y_5	0	0	1	1
q_2	0	0	1	0	x_2	q_8	1	0	0	0	y_{13}	1	0	1	0
					$\ddot{x}_2$	q_3	0	0	1	1	y_6	0	0	0	1
q_3	0	0	1	1	-	q_4	0	1	0	0	y_7	0	1	1	1
q_4	0	1	0	0	x_2	q_5	0	1	0	1	y_8	0	0	0	1
					$\ddot{x}_2$						y_9				
q_5	0	1	0	1	-	q_6	0	1	1	0	y_{10}	0	0	1	1
q_6	0	1	1	0	x_3	q_7	0	1	1	1	y_6	0	0	0	1
					$\ddot{x}_3$	q_3	0	0	1	1	y_6	0	1	0	1
q_7	0	1	1	1	x_2	q_8	1	0	0	0	-	1	1	1	1
					$\ddot{x}_2$						y_{11}				
q_8	1	0	0	0	-	q_0	0	0	0	0	y_{12}	1	0	0	0

Таблица 5 - структура управляющего автомата

8.4 Каноническая система логических уравнений

Из таблицы функций перехода, возбуждения и выходов автомата Мили получаем каноническую систему логических уравнений для цифрового автомата:

$T_1 = Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_1 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_2 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4\ddot{x}_2 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_2 \vee \bar{Q}_1\bar{Q}_2\bar{Q}_3\bar{Q}_4$
$T_2 = \bar{Q}_1Q_2\bar{Q}_3\bar{Q}_4 \vee \bar{Q}_1\bar{Q}_2Q_3Q_4\ddot{x}_3 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_2 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4\ddot{x}_2$
$T_3 = \bar{Q}_1\bar{Q}_2Q_3\bar{Q}_4\ddot{x}_1 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_2 \vee \bar{Q}_1Q_2\bar{Q}_3\bar{Q}_4 \vee \bar{Q}_1Q_2Q_3\bar{Q}_4 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4\ddot{x}_2 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_2$
$T_4=\bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_1 \vee \bar{Q}_1\bar{Q}_2Q_3\bar{Q}_4\ddot{x}_1 \vee \bar{Q}_1\bar{Q}_2Q_3Q_4\ddot{x}_2 \vee \bar{Q}_1Q_2\bar{Q}_3\bar{Q}_4 \vee \bar{Q}_1Q_2\bar{Q}_3Q_4x_2 \vee \bar{Q}_1Q_2\bar{Q}_3Q_4\ddot{x}_2 \vee \bar{Q}_1Q_2Q_3\bar{Q}_4 \vee \bar{Q}_1Q_2Q_3Q_4x_3 \vee \bar{Q}_1\bar{Q}_2Q_3Q_4\ddot{x}_3 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4\ddot{x}_2 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_2$
$y_1 = \bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4$
$y_2 = \bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4$
$y_3 = \bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4$
$y_4 = \bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4$
$y_5 = \bar{Q}_1\bar{Q}_2Q_3\bar{Q}_4\ddot{x}_1$
$y_6 = \bar{Q}_1\bar{Q}_2Q_3Q_4\ddot{x}_2 \vee \bar{Q}_1Q_2Q_3Q_4x_3 \vee \bar{Q}_1\bar{Q}_2Q_3Q_4\ddot{x}_3$
$y_7 = \bar{Q}_1Q_2\bar{Q}_3\bar{Q}_4$
$y_8 = \bar{Q}_1Q_2\bar{Q}_3Q_4x_2$
$y_9 = \bar{Q}_1Q_2\bar{Q}_3Q_4\ddot{x}_2$
$y_{10} = \bar{Q}_1Q_2Q_3\bar{Q}_4$
$y_{11} = Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4\ddot{x}_2$
$y_{12} = \bar{Q}_1\bar{Q}_2\bar{Q}_3\bar{Q}_4$
$y_{13} = Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_1 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_2$

8.5 Минимизация системы канонических уравнений

После минимизации получим следующие уравнения для сигналов возбуждения:
$T_1 = Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_1 \vee \bar{Q}_2\bar{Q}_3\bar{Q}_4$
$T_2 = \bar{Q}_1Q_2\bar{Q}_3\bar{Q}_4 \vee \bar{Q}_1\bar{Q}_2Q_3Q_4\ddot{x}_3 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4$
$T_3 = \bar{Q}_1\bar{Q}_2Q_3\bar{Q}_4\ddot{x}_1 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_2 \vee \bar{Q}_1Q_2\bar{Q}_4 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4$
$T_4 = \bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4x_1 \vee \bar{Q}_1\bar{Q}_2Q_3\bar{Q}_4\ddot{x}_1 \vee \bar{Q}_1\bar{Q}_2Q_3Q_4\ddot{x}_2 \vee \bar{Q}_1Q_2\bar{Q}_3 \vee \bar{Q}_1Q_2Q_3\bar{Q}_4 \vee \bar{Q}_1Q_2Q_3Q_4x_3 \vee \bar{Q}_1\bar{Q}_2Q_3Q_4\ddot{x}_3 \vee Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4$
$y_1 = \bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4$
$y_2 = \bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4$
$y_3 = \bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4$
$y_4 = \bar{Q}_1\bar{Q}_2\bar{Q}_3Q_4$
$y_5 = \bar{Q}_1\bar{Q}_2Q_3\bar{Q}_4\ddot{x}_1$
$y_6 = \bar{Q}_1\bar{Q}_2Q_3Q_4\ddot{x}_2 \vee \bar{Q}_1Q_2Q_3Q_4$
$y_7 = \bar{Q}_1Q_2\bar{Q}_3\bar{Q}_4$
$y_8 = \bar{Q}_1Q_2\bar{Q}_3Q_4x_2$
$y_9 = \bar{Q}_1Q_2\bar{Q}_3Q_4\ddot{x}_2$
$y_{10} = \bar{Q}_1Q_2Q_3\bar{Q}_4$
$y_{11} = Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4\ddot{x}_2$
$y_{12} = \bar{Q}_1\bar{Q}_2\bar{Q}_3\bar{Q}_4$
$y_{13} = Q_1\bar{Q}_2\bar{Q}_3\bar{Q}_4(x_1 \vee x_2)$

8.6 Функционально-логическая схема управляющего автомата

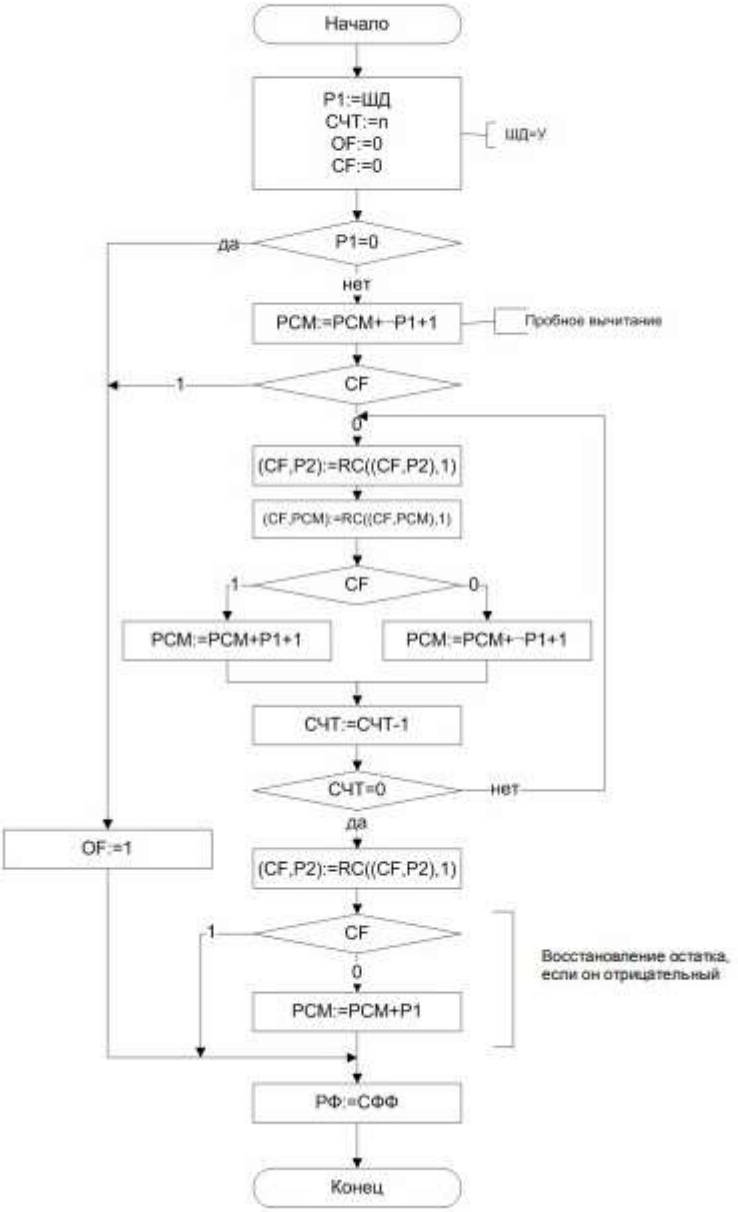
По упрощенной системе канонических уравнений строится функционально-логическая схема управляющего автомата. Представлена в приложении Б (рисунок 6). Временная диаграмма для данной схемы отображена в приложении Б (рисунок 7).

Литература:

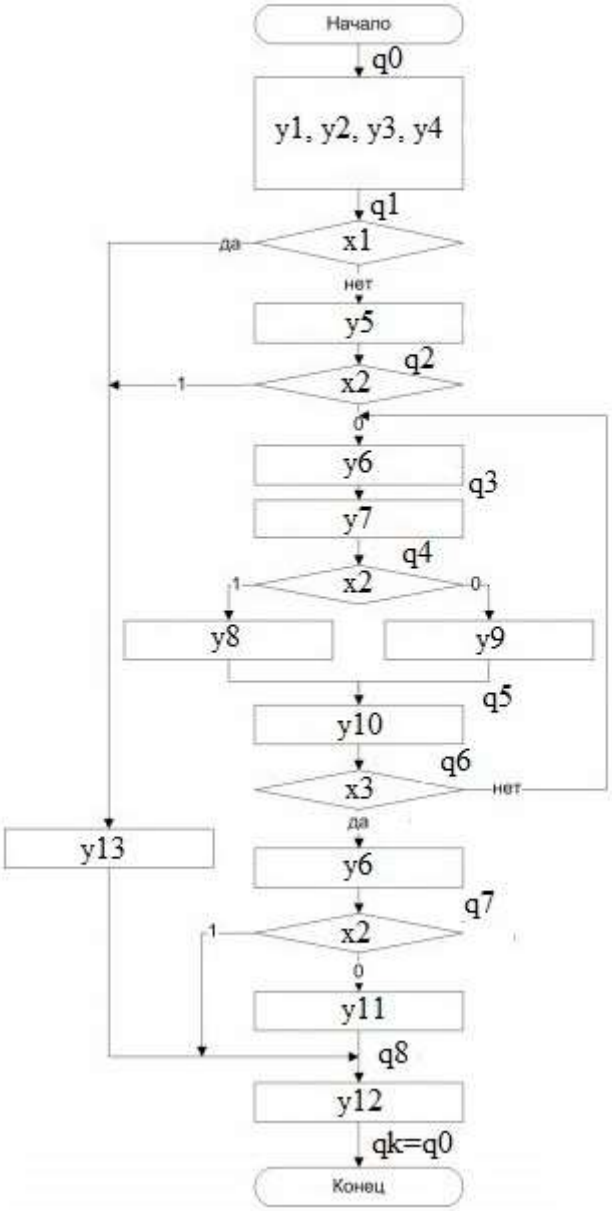
1. Савельев А. Я. Прикладная теория цифровых автоматов. Учебник для вузов по специальности ЭВМ. – 1-е издание. – М., Высшая школа, 1987.
2. Майоров С. А., Новиков Г. И. Принципы организации цифровых машин. – 1-е издание. – Л., Машиностроение, 1974.
3. Методические указания к выполнению курсового проектирования по курсу “Прикладная теория цифровых автоматов”. – 1-е издание. – Обнинск, ИАТЭ, 1990.
4. И.В. Жукалина. Теория автоматов. – 2-е издание. – Оренбург: ГОУ ОГУ, 2008.
5. Глушков, В.М. Синтез цифровых автоматов. – 1-е издание. – М: Государственное издательство физико-математической литературы, 1962.

Приложение А.

Рисунок 3 - содержательная граф-схема алгоритма микропрограммы

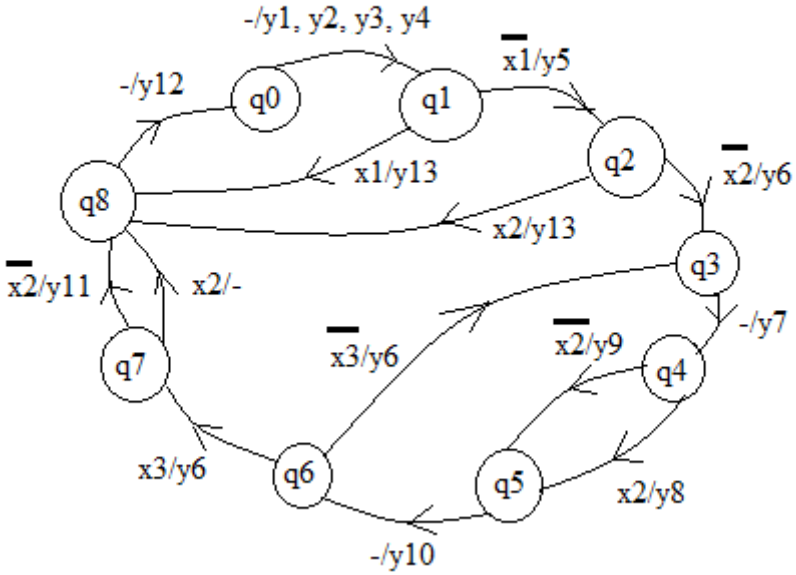


Приложение А.
Рисунок 4 - граф-схема алгоритма.



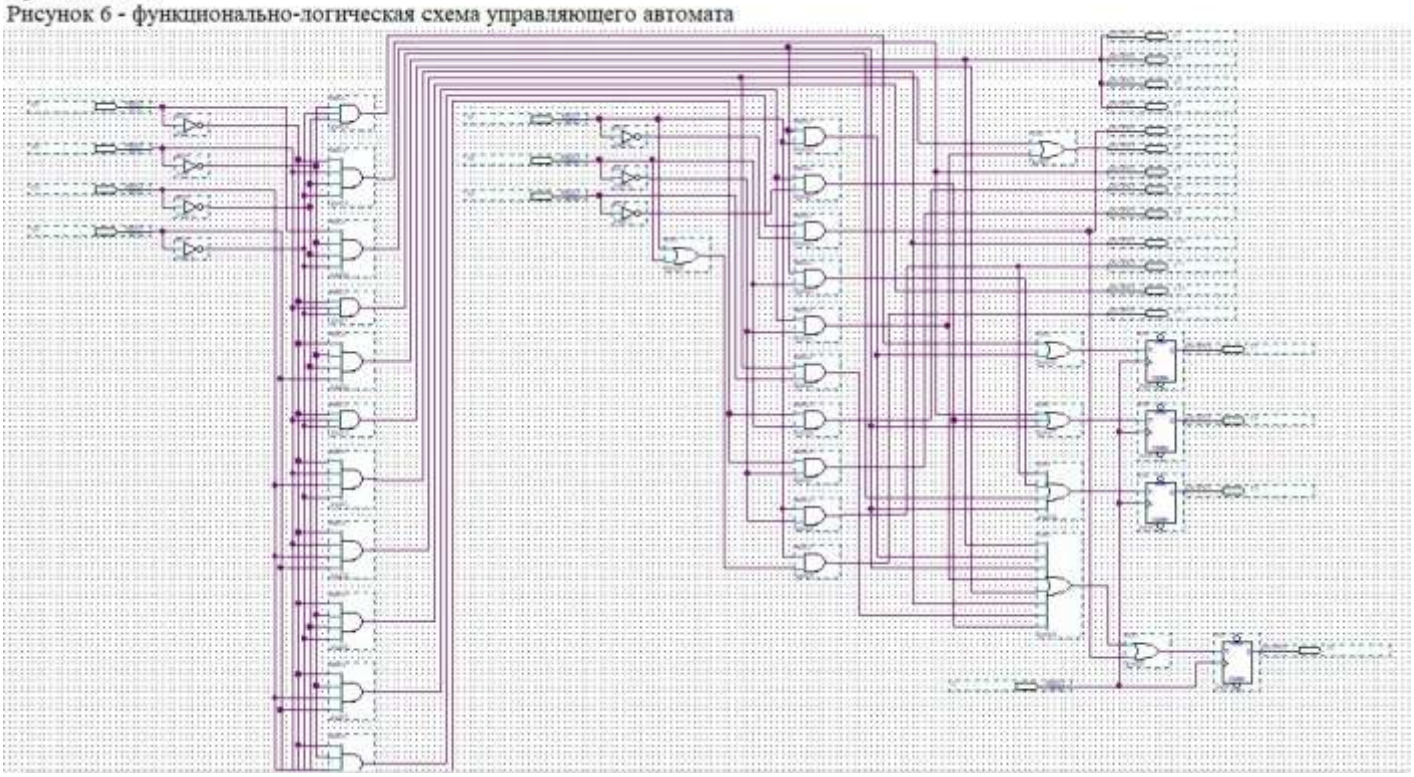
Приложение А.

Рисунок 5 - граф переходов управляющего автомата



Приложение Б.

Рисунок 6 - функционально-логическая схема управляющего автомата



Приложение Б.

Рисунок 7 - временная диаграмма для функционально-логической схемы управляющего автомата