

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

В. Ю. МЕЛЬЦОВ
Т. Р. ФАДЕЕВА

СИНТЕЗ МИКРОПРОГРАММНЫХ УПРАВЛЯЮЩИХ АВТОМАТОВ

Учебное пособие

Киров 2010

УДК 004.312.466(07)
М486

Учебное пособие для дисциплины "Теория автоматов".
/Вятский государственный технический университет.
Киров, 2010, 56с.

В учебном пособии даются пояснения по выполнению основных этапов курсового проектирования: выбору функциональной схемы операционного автомата, разработке и разметке содержательной ГСА, построению графа автомата и структурной таблицы переходов и выходов, выбору функциональной схемы управляющего автомата, кодированию состояний управляющего автомата, формированию логических выражений для функций возбуждения и функций выходов, построению принципиальной схемы микропрограммного управляющего автомата.

Методические указания предназначены для студентов очного обучения специальности 230101 - "Вычислительные машины, комплексы, системы и сети".

Составитель: доцент каф. ЭВМ, к.т.н. В.Ю. Мельцов.

Рецензент: профессор каф. АТ, к.т.н. В.В. Куклин.

© Вятский государственный технический университет, 2010
© В.Ю. Мельцов, Т.Р. Фадеева

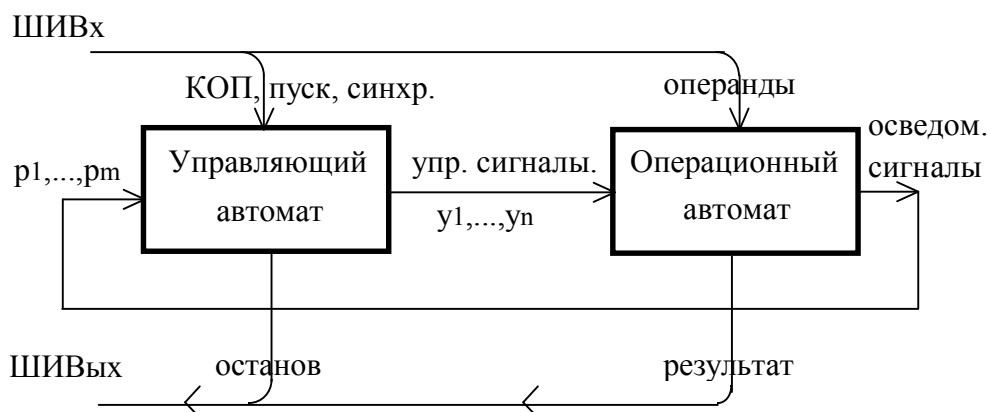
Оглавление

1. Рекомендации по выполнению курсовой работы	4
1.1. Функциональная схема операционного автомата.....	6
1.2. Разработка содержательной граф-схемы алгоритма	18
1.3. Разметка содержательной граф-схемы алгоритма.....	19
1.4. Построение графа автомата и структурной таблицы переходов и выходов	20
1.5. Выбор и обоснование функциональной схемы управляющего автомата.....	23
1.6. Кодирование состояний управляющего автомата	26
1.7. Формирование логических выражений для функций возбуждения и функций выходов	29
1.8. Построение функциональной схемы управляющего МПА	30
2. Пример синтеза МПА с жёсткой логикой, управляющего операцией сложения двоичных чисел в форме с плавающей запятой	33
2.1. Выбор и описание функциональной схемы ОА.....	33
2.2. Реализация содержательной ГСА.....	36
2.3. Построение отмеченной ГСА	39
2.4. Построение графа автомата и структурной таблицы переходов и выходов	42
2.5. Построение функциональной схемы управляющего МПА	49
3. Тематика курсовых работ. Варианты заданий	49
4. Требования к оформлению и защите курсовой работы	56
5. Календарный план выполнения курсовой работы.....	59
Рекомендуемая литература	60
Приложение	61

Основной целью курсовой работы является приобретение практических навыков синтеза управляющих микропрограммных автоматов с жесткой логикой на основе разработки машинных алгоритмов одной из заданных арифметических операций.

1. Рекомендации по выполнению курсовой работы

Любое вычислительное устройство может быть представлено композицией взаимодействующих пар автоматов - операционного автомата (ОА) и управляющего автомата (УА) (рис.1).



Операционный автомат содержит операционные устройства - регистры, сумматоры, счётчики, дешифраторы, мультиплексоры и др., на которых выполняется преобразование информации. В операционный автомат из других устройств ЭВМ поступают операнды по входной шине (ШИВ_Х), а после выполнения предписанной операции результат по выходной шине (ШИВ_{Вых}) передается в другие устройства ЭВМ.

Управляющий автомат в соответствии с кодом операции (КОП) и внешними сигналами (пуск, синхронизация) вырабатывает множество управляющих сигналов, которые поступают в операционный автомат и изменяют состояние операционных устройств в соответствии с реализуемой микропрограммой. Порядок следования управляющих сигналов определяется специальными осведомительными сигналами, называемыми логическими условиями (ЛУ), которые формируются на устройствах операционного автомата и значения которых проверяются в каждом такте работы управляющего автомата. После завершения выполнения операции управляющий автомат посылает на ШИВых сигнал останова.

Далее следует четко определить несколько понятий, которые широко используются при синтезе вычислительных устройств.

Микрооперация (МО) - это элементарный акт обработки информации в операционном автомате на одном устройстве за один такт машинного времени под воздействием одного управляющего сигнала.

Микрокоманда (МК) - это совокупность микроопераций, выполняемых на нескольких устройствах одновременно за один такт машинного времени под воздействием нескольких управляющих сигналов.

Микропрограмма - это совокупность микрокоманд и функций перехода (зависящих от логических условий) реализуемая за несколько тактов машинного времени.

Управляющий автомат, реализующий микропрограмму работы дискретного устройства, называется микропрограммным автоматом (МПА). Существует несколько способов проектирования управляющего МПА, среди которых наиболее известны:

- управляющие автоматы с "жесткой" или схемной логикой;
- управляющие автоматы с хранимой в памяти или программируемой логикой;

- управляющие автоматы на программируемых БИС с матричной структурой.

Задачей данной курсовой работы является синтез управляющего МПА с жесткой логикой, при проектировании которого можно выделить следующие основные этапы:

1. Выбор функциональной схемы ОА, определение списка МО и ЛУ.
2. Разработка содержательной граф-схемы алгоритма (ГСА) в соответствии со словесным описанием алгоритма заданной операции и выбранной структурой ОА.
3. Разметка содержательной ГСА и формирование отмеченной ГСА для модели Мили и (или) модели Мура.
4. Построение графа автомата и структурной таблицы переходов и выходов.
5. Выбор и обоснование функциональной схемы УА.
6. Выбор способа кодирования внутренних состояний УА, типа элементов памяти (ЭП) и завершение формирования структурной таблицы.
7. Запись логических выражений для функции возбуждения ЭП, функций выходов и их совместная минимизация.
8. Построение логической (принципиальной) схемы управляющего МПА, цепей начальной установки, синхронизации и запуска.

Ниже будут даны некоторые рекомендации по реализации перечисленных этапов синтеза МПА в курсовой работе.

1.1. Функциональная схема операционного автомата

Начинать проектирование управляющего МПА следует с определения структуры операционного автомата, состава операционных устройств, фиксации множества МО, необходимых для реализации алгоритма заданной операции, и множества ЛУ, определяемых в процессе выполнения алгоритма.

Этот этап проектирования выполняется параллельно с разработкой содержательной ГСА по словесному описанию алгоритма.

Далее дано описание основных операционных устройств, используемых в ОА, приведены правила их изображения в функциональных схемах и перечислены основные МО, реализуемые на каждом из устройств.

Шина с функциональной точки зрения представляет линию связи между любыми устройствами. Шина может иметь различное количество каналов в зависимости от разрядности связываемых устройств. Шины разделяют на информационные и управляющие и по-разному изображают в функциональных схемах ОА.

Любое операционное устройство (регистры, сумматоры, счетчики и др.) изображают в функциональных схемах ОА в виде условного графического обозначения (УГО) в соответствии с ГОСТ 2.708-81, ГОСТ 2.743-72 (рис.2).

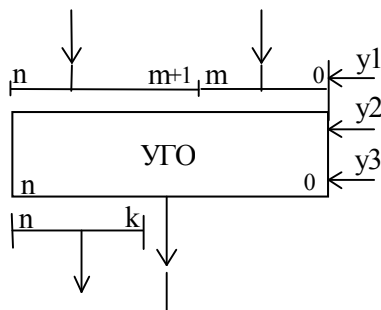


Рис.2

Размер Н выбирается из ряда 10, 15 мм и далее через 5 мм, а размер L=1,5 Н.

Допускается совмещать УГО функциональных узлов, если выходы одного полностью соответствуют входам другого. Внутри УГО функционального узла указывают его наименование и (или) условное обозначение.

Информационные линии связи следует подводить к большей стороне УГО сверху, а отводить от противоположной стороны УГО снизу. Если информация снимается с части разрядов УГО, то следует изображать параллельную линию с указанием над ней цифрами граничных значений разрядов, как показано на рис.2.

Управляющие линии связи подводят к любой меньшей (обычно правой!) стороне УГО или к линии продолжения этой меньшей стороны.

Р е г и с т р - упорядоченная совокупность запоминающих элементов (триггеров) со схемами управления, предназначенная для записи, хранения и выдачи информации, а также для выполнения некоторых микроопераций над этой информацией. По изображению в функциональных схемах ОА регистры разделяют на несдвиговые, изображаемые в виде прямоугольника (рис.3а). и сдвиговые, на которых может быть реализована МО сдвига содержимого (они изображаются в виде параллелограмма - рис.3б и 3в). В поле изображения необходимо указать номер регистра и его разрядность.

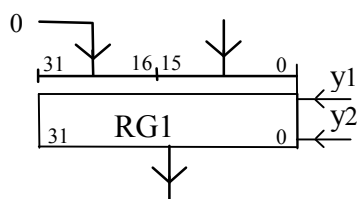


Рис. 3.а

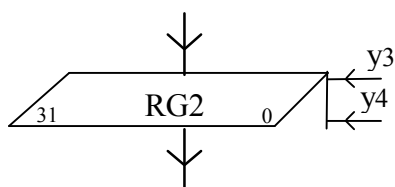


Рис. 3.б

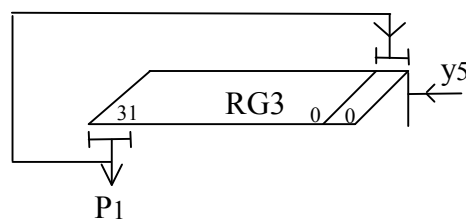


Рис.3в

На регистрах могут быть реализованы следующие МО.

y1 - занесение в RG1.

y2 - $RG1 := 0$ - сброс всех разрядов в ноль.

y3 - занесение в RG2.

y4 - $RG2 := R1 (RG2)$ - сдвиг содержимого вправо на 1 разряд.

y5 - $RG3 := L1 (RG3)$, $RG3[0] := RG3[31]$ - циклический сдвиг влево.

Сигнал y1 реализует МО занесения информации в регистр RG1. При этом внешние данные заносятся в младшие разряды регистра ($RG1[0,15]$), а в старшие разряды ($RG1[16,31]$) **всегда заносятся нули** (рис.3а).

Сдвиги вправо изображают заглавной латинской буквой R, влево - L с указанием рядом количества разрядов, на которое выполняется сдвиг. Например, сдвиг на четыре разряда вправо следует записать $RG2 := R4 (RG2)$. Необходимо помнить, что в стандартных микросхемах реализованы сдвиги только на один разряд.

При сдвигах освобождающиеся разряды регистра заполняются необходимой информацией (0 или 1). По умолчанию записывается случайная информация! При реализации операций сдвига в ОК или ДК иногда

требуется заполнять освобождающиеся разряды регистра единицами. В этих случаях МО сдвига следует записать так: $RG2 := 1.R1 (RG2)$ или $RG3 := L1 (RG3).0$.

МО циклического сдвига y5 следует использовать в тех алгоритмах, когда необходимо сохранять операнды до конца операции (например, в алгоритмах умножения с простой коррекцией). В

приведенной на рис.3в схеме в нулевой разряд $RG3 [0]$ после сдвига запишется информация, "выталкиваемая" из $RG3 [31]$.

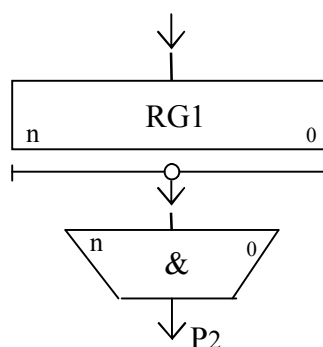


Рис.4

Нередко при выполнении алгоритмов арифметических операций необходимо проверить, не записан ли в регистре 0 (например, проверка делителя на нуль в операциях деления). В этих случаях используют схему логического "И", на n входов которой подают информацию с инверсных выходов триггеров регистра, а с единственного выхода схемы "И" можно снять логическое условие p_2 . При этом, если $p_2=1$, в регистр занесен 0. На

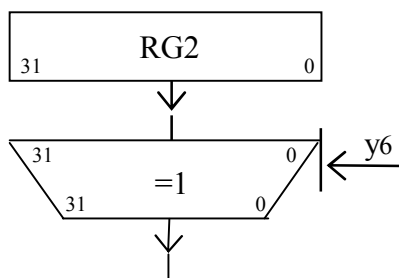


Рис.5

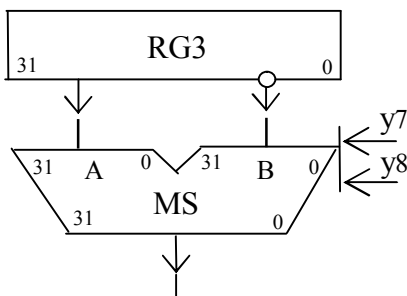


Рис.6

рис.3в показано, как можно снять другое логическое условие $P1 = RG3 [31]$, анализ которого позволит определить, например, знак операнда в RG3.

При выполнении машинных алгоритмов операций содержимое регистров необходимо передать в другие операционные устройства, чаще всего в сумматор. Если выходы триггеров

регистра (прямые или инверсные) соединены шиной со входом сумматора без

каких-либо промежуточных устройств, то такой

съем информации называют неуправляемым.

При этом содержимое регистра постоянно

находится на соответствующих входах

сумматора. Для управления съемом информации

могут быть использованы различные

операционные устройства, включаемые между

выходом регистра и соответствующим входом сумматора.

Чаще всего при выполнении алгоритма в одних тактах необходимо передать в сумматор содержимое RG без инверсии, в других - с инверсией. На рис.5 представлен один из возможных вариантов реализации с использованием совокупности схем сложения по модулю 2. Пока не подан управляющий сигнал y_6 , на выход совокупности схем поступает неинвертированная информация с выходов регистра.

Для реализации управления съемом информации с регистра может быть использована комбинационная схема специального вида - мультиплексор MS, о котором более подробная информация будет дана позже. На рис.6 представлен мультиплексор, на плечо А которого поступает информация с прямых выходов триггеров регистра, на плечо В - с инверсных выходов. Под управлением сигнала y_7 к выходу MS будет подключено плечо В. Когда же y_7 не подается, на выход MS поступает информация с плеча А. Сигнал y_8 позволяет получить на выходе MS нули.

Замечание. Следует обратить внимание на то, что МО y_6 и y_7 , управляющие съёмом информации с регистра, реализуются не на регистре, а на других устройствах операционного автомата.

С ч е т ч и к (двоичный) предназначен для подсчета числа выполненных тактов (например, в циклах умножения и деления) и для хранения результата счета. С точки зрения структуры счетчик (СТ) это регистр, дополненный операциями $+1$ и -1 . В силу этой особенности в функциональных схемах устройств, реализующих операции с ПЗ, удобно порядок (или характеристику) результата операции заносить в счетчик для упрощения операции нормализации мантиссы результата. Счетчики бывают суммирующие, вычитающие и реверсивные, то есть настраиваемые на сложение или вычитание. Если N - максимальное число, которое необходимо занести в счетчик, то его разрядность определяется формулой $n \geq \lceil \log_2 N \rceil$.

Счетчик имеет цепи установки в нулевое состояние всех триггеров, в него можно занести любое число по установочным входам триггеров, а также выполнять операции счета. Поэтому основные МО на СТ:

y9 - СТ := 0 - сброс.

y10 - занесение информации.

y11 - СТ := СТ+1 или СТ := СТ-1.

Обычно содержимое счетчика необходимо анализировать в цикле умножения (или деления), чтобы, выполнив нужное количество действий, завершить цикл. Для этого выходы триггеров счетчика следует подать на схему анализа, с выхода которой снять логическое условие. Пусть СТ работает на вычитание, и закончить цикл надо, когда содержимое счетчика равно нулю. Это можно сделать, если инверсные выходы триггеров СТ подать на схему "И", с выхода которой снять логическое условие p3 (рис.7).

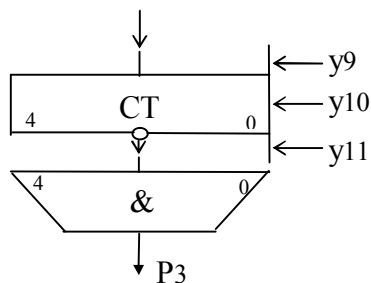


Рис.7

Цикл завершится, когда $p3 = 1$, т.е. когда во всех разрядах счётчика - нули. Обнаружить это же условие можно, используя вместо схемы "И" элемент Пирса с нужным количеством входов. Аналогичным образом можно обнаружить, когда во всех разрядах счетчика "записаны" единицы.

С у м м а т о р - это комбинационная схема для сложения двух двоичных чисел. В функциональных схемах ОА сумматор изображают так, как показано на рис.8, выделяя два входных плеча А и В, на которые поступают операнды, и выходное плечо S, с которого снимают результат. Кроме того, сумматор имеет вход переноса CRP и выход переноса CR, которые выделяют при изображении в тех случаях, когда они используются.

На SM чаще всего реализуется МО подачи единицы на вход переноса: $y_{12} - SMp = 1$, если выполняется сложение операндов в ДК, что равносильно прибавлению единицы к младшему разряду суммы. При выполнении некоторых операций может возникнуть необходимость в занесении единиц в часть разрядов SM (рис.8).

Главное, о чем следует помнить при реализации такта сложения:

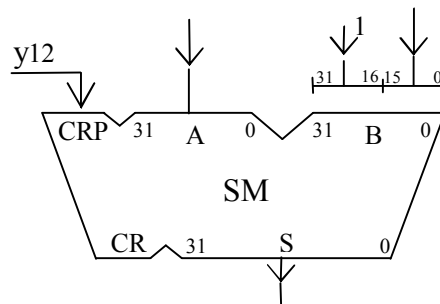


Рис.8

сумматор - комбинационная схема, которая формирует результат на выходе S, пока поданы операнды на входы A и B, но не хранит результат операции. Поэтому результат операции сложения должен быть передан в том же такте в регистр для запоминания. Например, если операнды занесены в RG1 и RG2, причем операнд в RG2 отрицательный, и нужно выполнить операцию сложения с использованием ДК, то такт сложения должен быть записан так:

$$\begin{aligned} SMa &= \underline{RG1} \\ SMb &= RG2 \\ SMp &= 1 \\ RG3 &:= SMs \end{aligned}$$

Все МО, реализующие перечисленные выше действия, должны быть включены в одну МК.

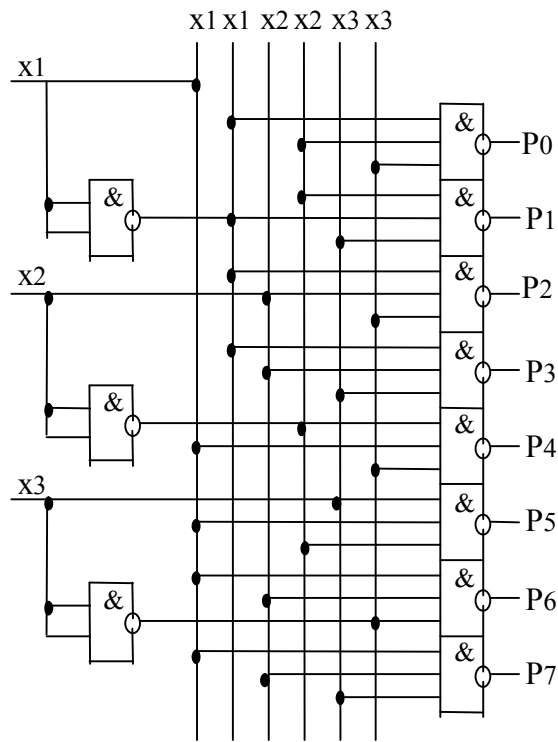


Рис.9

Дешифратор (DC) - это комбинационная схема, которая преобразует код, подаваемый на его входы, в единичный сигнал на одном из выходов, то есть преобразует n -разрядное входное слово в 2^n -разрядный унитарный код. На рис.9 приведена функциональная схема дешифратора на 3 входа, выполненная на элементах И-НЕ. В функциональных схемах дешифратор условно изображают так, как показано на рис.10.

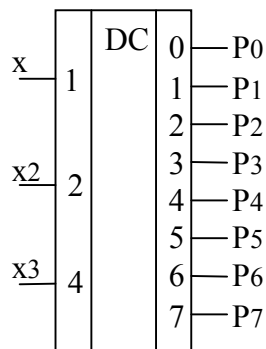


Рис.10

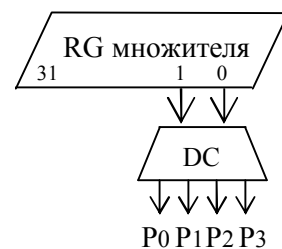


Рис.11

В функциональных схемах ОА используют дешифраторы при реализации алгоритмов умножения с ускорением. На входы дешифратора подают группу анализируемых разрядов множителя, а с выходов снимают логические условия, анализ которых позволяет выявлять двоичные комбинации на входе.

На рис.11 приведен пример использования дешифратора при выполнении алгоритма умножения с ускорением 2-го порядка. 4 логических условия, снимаемых с дешифратора: p_0 , p_1 , p_2 , p_3 позволяют выявить всевозможные двухразрядные комбинации в двух младших разрядах регистра множителя.

Мультиплексор (MS) - это комбинационная схема, осуществляющая передачу сигналов с одной из входных линий в выходную. Выбор входной информационной линии производится кодом, поступающим на управляющие входы мультиплексора. Мультиплексор с k управляющими входами может управлять 2^k входными информационными линиями.

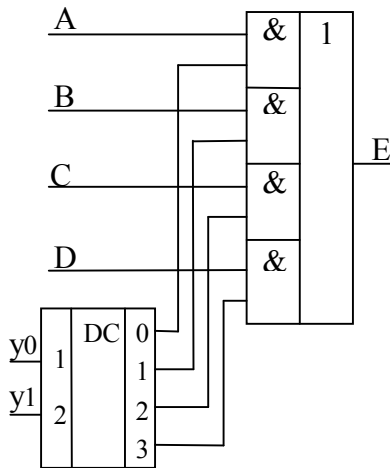


Рис.12

На рис.12 приведена функциональная схема мультиплексора с двумя управляющими входами y_0 и y_1 , подача сигналов на которые позволяет подключать на выход E одну из четырех входных информационных линий A , B , C , D .

В схеме использован дешифратор на 2 входа, позволяющий распознавать, какие управляющие сигналы поданы на входы y_0 , y_1 для подключения одной из линий на выход мультиплексора.

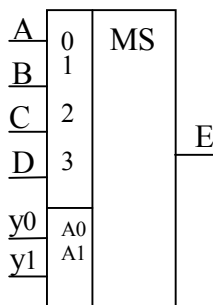


Рис.13а

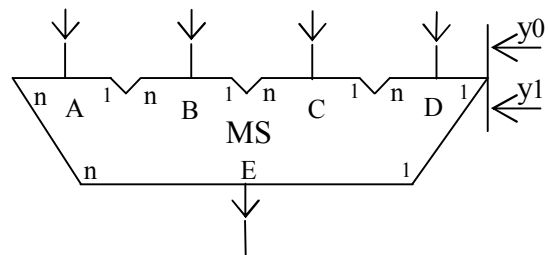


Рис.13б

На рис.13а дано условное обозначение мультиплексора на принципиальных схемах, а на рис.13б - изображение, которое используют на функциональных схемах ОА. Мультиплексор используют, например, в случаях, когда на одно плечо сумматора надо в разных тактах подавать содержимое различных регистров.

Управление выбором плеча MS, подключаемого к выходу (рис.13б), выполняется в соответствии с табл.1.

Таблица 1

На выход E	A	B	C	D
Управляющие сигналы	-	Y0	Y1	Y0, Y1

Из приведенной таблицы видно, что когда ни один из управляющих сигналов не подан, к выходу E подключено плечо A, подача управляющего сигнала y_0 подключает плечо B, сигнала y_1 - плечо C, подача же обоих сигналов y_0, y_1 обеспечивает подключение к выходу E плеча D.

Таким образом, управляющие сигналы y_0 и y_1 осуществляют выбор одной из 4 входных информационных линий. Если управляющих сигналов больше, то и количество входных линий значительно возрастает. Например, при $K = 3$ возможен выбор одной из 8 линий, при $K = 4$ выбор из 16 линий и т.д. Существуют варианты схем MS, в которых имеется дополнительный управляющий сигнал разрешения подключения к выходу, без подачи которого информация ни с одной из информационных линий не поступает на выход.

Таковы основные операционные устройства, используемые в функциональных схемах ОА. В них, кроме того, могут использоваться отдельные триггеры (например, для фиксации ПРС), сумматоры по модулю 2 (для определения знака результата), инверторы (для формирования цифр частного) и другие.

1.2. Разработка содержательной граф-схемы алгоритма

Микропрограмма работы проектируемого устройства может быть описана в виде ориентированного связного графа, содержащего одну начальную вершину, одну конечную вершину и конечное число операторных и условных вершин. В каждой операторной вершине записывается одна МК, содержащая одну или несколько МО, а в каждой условной вершине - одно из логических условий. Допускается запись одинаковых МК в различных операторных вершинах и одинаковых ЛУ в различных условных вершинах. Такую запись микропрограммы в виде графа называют содержательной ГСА.

При записи микропрограммы на языке ГСА следует руководствоваться следующими правилами.

1. Начальная вершина должна иметь один выход, конечная - один вход, операторная - один вход и один выход, условная - один вход и не менее двух выходов. В логической вершине допустимо вывести один из выходов на собственный вход, реализовав таким образом режим ожидания.

2. Из любой вершины ГСА должен существовать хотя бы один путь в конечную вершину.

3. К входной стрелке операторной вершины может подходить несколько выходных стрелок от других вершин ГСА, при этом каждый выход вершины соединен точно с одним входом.

Содержательная ГСА составляется в полном соответствии со словесным описанием алгоритма выполнения заданной операции. Далее в соответствии с выбранной структурой она составляется список МО и ЛУ. Теперь каждой операторной вершине содержательной ГСА можно сопоставить набор МО, образующих МК.

1.3. Разметка содержательной граф-схемы алгоритма

Для синтеза микропрограммного автомата необходимо выполнить переход от содержательной ГСА к отмеченной ГСА путем выделения состояний автомата (разметки) в соответствии с моделью Мили или моделью Мура.

Предварительно в каждой условной вершине проставляются символы из множества входных сигналов УА - x_1, x_2, \dots, x_M , каждый из которых эквивалентен одному из осведомительных сигналов ОА. Во всех операторных вершинах ГСА проставляют символы из множества выходных сигналов управляющего автомата - y_1, y_2, \dots, y_N , эквивалентных одной из МК, поступающих в ОА в соответствующие такты машинного времени. Удобно в каждой операторной вершине ГСА вслед за символом МК указать в скобках набор МО, образующих каждую МК. Допускается запись одинаковых символов X_i и Y_j в различных условных и операторных вершинах.

Выделение состояний автомата Мили, то есть разметка ГСА в соответствии с моделью Мили, выполняется по следующим правилам:

1. Вход вершины, следующей за начальной, и вход конечной вершины отмечаются символом начального состояния автомата a_0 .
2. Входы всех вершин, следующих за операторными, отмечаются символами a_1, \dots, a_K .
3. Если вход вершины отмечается, то только одним символом.
4. Входы различных вершин за исключением конечной отмечаются различными символами.

Выделение состояний автомата Мура, то есть разметка ГСА в соответствии с моделью Мура, выполняется по следующим правилам.

1. Символом начального состояния a_0 отмечаются начальная и конечная вершины.

2. Различные операторные вершины отмечаются различными символами a_1, \dots, a_k .

3. Все операторные вершины должны быть отмечены, то есть каждой МК, отдельно представленной в ГСА ставится в соответствие отдельное состояние автомата Мура.

4. В логических вершинах ГСА, реализующих режим ожидания, существует возвратная дуга, когда один из выходов вершины подан на ее вход. На этой дуге необходимо вводить дополнительное фиктивное состояние автомата Мура.

Для одной и той же ГСА количество состояний для модели Мура, как правило, больше, чем для модели Мили. Однако при проектировании управляющего МПА трудно заранее определить, какая из моделей - Мили или Мура - даст комбинационную схему УА меньшей сложности. Поэтому, чаще всего, на начальном этапе проектирования студентам предлагается исследовать обе модели.

1.4. Построение графа автомата и структурной таблицы переходов и выходов

Имея отмеченную ГСА проектируемого управляющего микропрограммного автомата, следует описать его работу известными способами - графическим и табличным. Если количество состояний автомата и переходов между ними невелико, то задание его в виде графа позволяет наглядно представить работу МПА.

Граф автомата есть ориентированный связный граф, вершины которого соответствуют состояниям, а дуги - переходам между ними. Причем, две вершины графа a_m и a_s - соединены дугой, направленной от a_m (исходное состояние) к a_s (состояние перехода) если в ГСА существует этот переход.

Для автомата Мили каждой дуге приписываются входные и выходные сигналы, если они определены. Для автомата Мура дугам приписаны только входные сигналы; выходные сигналы приписаны вершинам графа.

Замечания.

1. В графе автомата необходимо указывать все возможные переходы между состояниями, "проходя" встречающиеся на пути из a_m в a_s условные вершины по всем исходящим из них дугам.

2. При построении графа автомата Мили необходимо избегать "пустых" переходов, то есть переходов, на которых не вырабатываются управляющие сигналы (или на которых не встречается операторной вершины, что эквивалентно). Чтобы не снижать быстродействия МПА, надо в графе показать переход в следующее состояние, если это возможно. Если автомат имеет большое число состояний и переходов между ними, то наглядность графа теряется. Тогда удобно использовать табличный способ задания автомата. При синтезе МПА строят прямые (или инверсные) структурные таблицы переходов и выходов.

Таблица 2.

Исх. Состояни е a_m	Код исх. состояния $K(a_m)$	Состояние перехода a_s	Код сос- тояния перехода $K(a_s)$	Входные сигналы $X(a_m, a_s)$	Выходные сигналы $Y(a_m, a_s)$	Функции возбужде- ния ЭП $F(a_m, a_s)$
1	2	3	4	5	6	7

В табл.2 дана прямая структурная таблица для автомата Мили. Для автомата Мура столбец 6 таблицы (входные сигналы) следует располагать вслед за первым столбцом.

Состояния, перечисляемые в первом столбце таблицы, должны быть упорядочены, то есть сначала следует указать все переходы из a_0 , затем из a_1 и т.д. Аналогично и для инверсной таблицы переходов (в a_0 , в a_1 и т.д.). При формировании столбца 5 (входные сигналы) следует указывать конъюнкцию всех входных сигналов, записанных в логических вершинах ГСА на данном переходе. Причем X_i берут без отрицания, если переход выполняется по единичному значению сигнала, и с отрицанием, если по нулевому значению сигнала.

Первоначально столбцы 2 и 4 таблицы (коды состояний) не могут быть заполнены, так как еще не выполнено кодирование состояний автомата. По этой же причине не могут быть определены и функции возбуждения элементов памяти (столбец 7). Формирование структурной таблицы будет завершено после выполнения пятого и шестого этапов синтеза МПА.

1.5. Выбор и обоснование функциональной схемы управляющего автомата

Этот этап структурного синтеза выполняется параллельно с выбором способа кодирования внутренних состояний и типа элементов памяти для управляющего автомата.

Вариант 1. Классическая структура УА - это совокупность взаимосвязанных элементов памяти (ЭП) и комбинационной схемы (КС), реализующей функции возбуждения ЭП и функции выходов Y (рис.14).

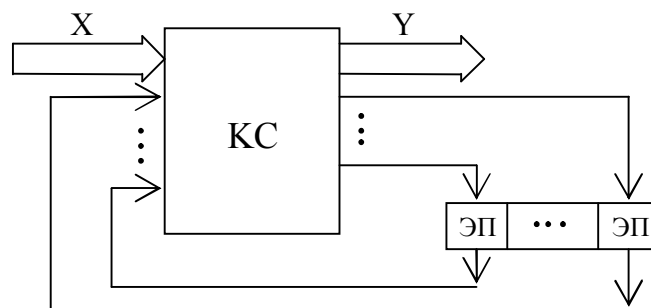


Рис.14

Набор ЭП фиксирует состояния, в которые переходит автомат в процессе выполнения заданной микропрограммы. Синтез УА сводится к синтезу его КС, на вход которой поступает множество входных сигналов X из ОА и сигналы, фиксирующие коды состояний автомата.

Естественным требованием при синтезе КС является минимизация ее цены, что обеспечивается правильным выбором способа кодирования состояний и типа ЭП, а также совместной минимизацией функций возбуждения ЭП и функций выходов.

Вариант 2. Цена КС может быть снижена, если сигналы с выходов ЭП подать на дешифратор ДС, что приводит к структуре, изображенной на рис.15.

В такой структуре УА необходимо стремиться к полному использованию выходов дешифратора. Если дешифратор, подключаемый к выходам всех ЭП, недоиспользуется, целесообразно продумать вариант установки дешифратора, подключаемого к части выходов ЭП.

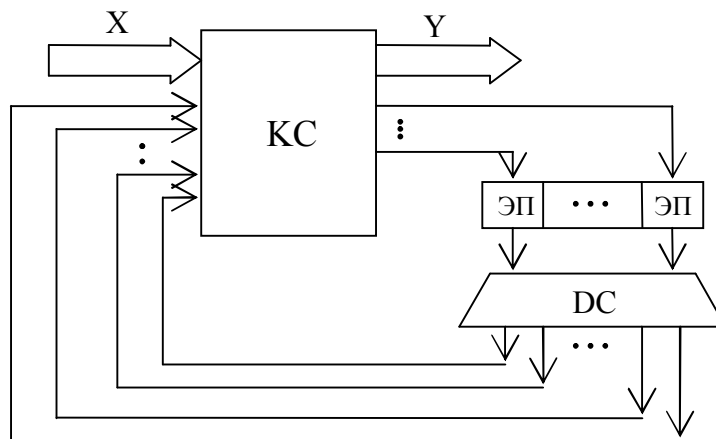


Рис.15

Вариант 3. В случае выбора унитарного способа кодирования внутренних состояний автомата, элементы памяти объединяются в регистр сдви-

га, что приводит к структуре УА, изображенной на рис.16.

Тогда часть комбинационной схемы, реализующая функции возбуждения элементов памяти, значительно упрощается, так как в этом случае необходимо вырабатывать лишь сигнал первоначальной установки регистра и сигнал сдвига содержимого регистра (в качестве которого часто используют сигнал синхронизации).

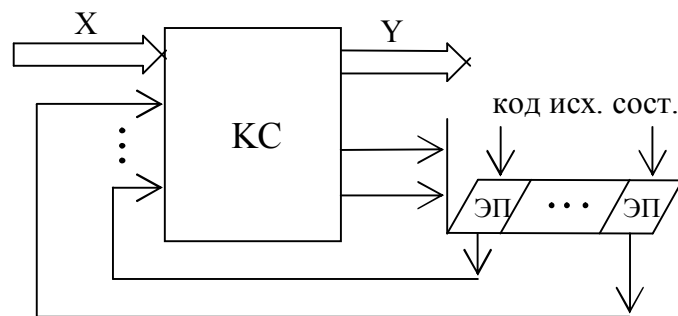


Рис.16

Вариант 4. Когда граф проектируемого автомата имеет большое количество последовательных (стандартных) переходов и незначительное количество нестандартных переходов, целесообразно закодировать состояния автомата последовательными двоичными числами. Это и диктует структуру

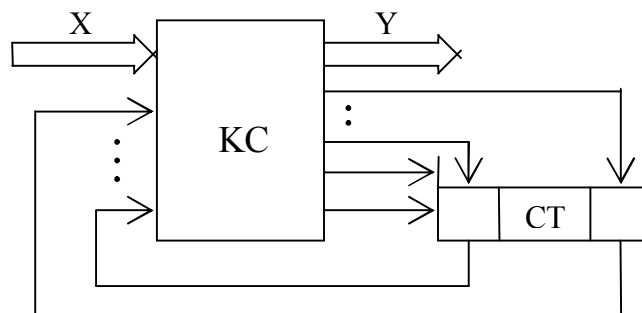


Рис.17

УА на основе двоичного счетчика, работающего в режиме сложения или вычитания. При этом на последовательных переходах необходимо подавать сигнал на счетный вход счетчика, что обеспечит стандартные переходы. На нестандартных переходах следует устанавливать нужный код по установочным входам счетчика. Если число нестандартных переходов невелико, то КС, реализующая функции возбуждения ЭП, будет проста.

Вариант 5. Структуру на основе двоичного счетчика можно модифицировать, используя дешифратор, подключаемый к выходам триггеров счетчика. Получим вариант структуры УА, аналогичный варианту 2 (рис.15) со счетчиком, используемым в качестве ЭП. В заключение этого параграфа следует отметить, что возможны и другие варианты структур УА. Однако, учитывая содержание заданий, предлагаемых студентам для выполнения курсовой работы, а также то, что данная работа имеет своей целью приобретение первых практических навыков проектирования МПА, которые при дальнейшем обучении будут расширены и углублены, можно на начальном этапе ограничиться предлагаемыми выше вариантами структур УА.

1.6. Кодирование состояний управляющего автомата

Выбор способа кодирования внутренних состояний автомата, типа используемых элементов памяти и варианты структуры МПА - взаимосвязанные вопросы, решаемые параллельно. От их правильного решения зависит сложность комбинационной схемы проектируемого МПА.

Основным ориентиром при выборе структуры и метода кодирования является граф автомата, наглядно представляющий переходы между состояниями при выполнении автоматом заданной микропрограммы.

Кодирование заключается в сопоставлении каждому состоянию МПА набора состояний элементов памяти одинаковой длины. Если N - число состояний автомата, n - число ЭП, используемых для кодирования состояний, то n определяется из интервала

$$\lceil \log_2 N \rceil \leq n \leq N,$$

то есть наименьшее число ЭП n есть ближайшее большее целое от $\log_2 N$, а наибольшее число ЭП равно числу состояний автомата N .

Формального способа выбора метода кодирования состояний автомата предложить нельзя, можно лишь дать некоторые рекомендации, а окончательный выбор остается за разработчиком проектируемого МПА, который он делает после сопоставления вариантов.

Наиболее простым является унитарный способ кодирования, при котором $n = N$. Этот способ кодирования в совокупности с вариантом 3 структуры УА целесообразен в тех случаях, когда $\lceil \log_2 N \rceil$ не намного меньше N , иначе будут значительными затраты на память автомата, которые поглотят выигрыш от уменьшения цены КС, формирующей функции возбуждения ЭП.

Другой простой способ кодирования связан с вариантами 4 и 5 структур УА на основе счетчика. Здесь состояния автомата кодируются последовательными двоичными числами, а формирование функций возбуждения ЭП на нестандартных переходах определяется типом элементов памяти, на которых построен счетчик. Этот способ кодирования и структуры УА целесообразен, как уже указывалось, для графов автомата, имеющих большое количество последовательных переходов.

Далее несколько слов о связи типа ЭП со способом кодирования. Среди различных типов триггеров особое место занимают синхронизируемые D-триггеры, имеющие вход данных D и вход синхронизации C . По сигналу C -входа триггер переключается в состояние, предписываемое D входом.

При использовании D-триггеров в качестве ЭП для получения смены состояний на каждом переходе ($a_m \rightarrow a_s$) сигналы возбуждения должны быть поданы на те триггеры, которые в коде состояния перехода a_s содержат "1". Отсюда основное требование к выбору кодов состояний: чем больше переходов в какое-либо состояние, тем меньше "1" должен содержать код этого состояния. Здесь удобно строить инверсные таблицы переходов. Этот способ кодирования позволит получить КС меньшей сложности.

Для триггеров, имеющих отдельные входы установки в "1" и в "0", (RS - триггеры, ТК - триггеры, Т - триггеры) целесообразно использовать метод кодирования, минимизирующий число переключений ЭП, в сочетании с методом соседнего кодирования. При этом сначала следует выделить пары соседей 1-го и 2-го рода и закодировать их соседними кодами, используя диаграмму Вейча-Карно. А затем применить метод, минимизирующий число переключений ЭП.

Замечание. При выборе числа элементов памяти следует учитывать, что в стандартных сериях логических элементов одна микросхема может содержать 4, 6 или 8 триггеров. Поэтому, если по расчетам, например, минимальное число ЭП получилось равным 3, то можно использовать 4 элемента памяти. Это позволит в большинстве случаев уменьшить цену КС для функций возбуждения ЭП.

Таким образом, после выбора типа ЭП и кодирования внутренних состояний автомата следует вернуться к структурной таблице переходов автомата и заполнить столбцы кодов состояний и столбец обязательных функций возбуждения ЭП.

При определении сигналов, которые должны быть поданы на входы триггеров для получения требуемой смены состояний на каждом переходе автомата, следует учитывать следующее.

При использовании D-триггеров сигналы возбуждения подаются на те триггеры, для которых в коде состояния перехода a_s записаны "1", то есть анализируется столбец кодов состояний перехода $K(a_s)$ структурной таблицы.

При использовании RS- и JK-триггеров сигнал возбуждения подают на S-вход (J-вход), если на переходе триггер требует смены состояний $0 \rightarrow 1$; сигнал возбуждения надо подать на R-вход (K-вход), если требуемая смена состояний триггера $1 \rightarrow 0$.

При использовании T-триггеров сигналы возбуждения подают на те триггеры, которые изменяют свое состояние на переходе ($0 \rightarrow 1$, $1 \rightarrow 0$).

1.7. Формирование логических выражений для функций возбуждения и функций выходов

По структурной таблице переходов и выходов можно построить логические выражения для всех выходных сигналов, формируемых КС.

Для формирования функций возбуждения ЭП из последнего столбца структурной таблицы выбираются строки, соответствующие определенной функции возбуждения. Для каждой строки составляются конъюнкции кодов исходных состояний и входных сигналов, записанных в данной строке. Образованные таким образом конъюнкции объединяют знаками дизъюнкций для всех строк, содержащих формируемую функцию возбуждения.

Аналогично записывают булевы выражения для функций выходов ориентируясь на столбец выходных сигналов $Y(a_m, a_s)$ структурной таблицы переходов и выходов. Для автомата Мили каждый управляющий сигнал содержит конъюнкции кодов состояний и входных сигналов, объединенных знаками дизъюнкций для всех строк, содержащих формируемый управляющий сигнал. Для автомата Мура управляющие сигналы есть

дизъюнкции состояний автомата, отмеченных соответствующими выходными сигналами.

После совместной минимизации полученной системы логических выражений для функций возбуждения ЭП и функций выходов можно перейти к построению КС в заданном логическом базисе.

1.8. Построение функциональной схемы управляющего МПА

Полученные на предыдущем этапе логические выражения для функций возбуждения ЭП и функций выходов позволяют построить комбинационную схему, реализующую эти функции. При этом может быть использован как основной логический базис И, ИЛИ, НЕ, так и любой другой базис по заданию преподавателя.

Построенная комбинационная схема в совокупности с набором ЭП и, быть может, другими элементами, устанавливаемыми в соответствии с выбранной структурой управляющего автомата, и дают функциональную схему управляющего микропрограммного автомата. При изображении схемы следует руководствоваться соответствующим ГОСТ.

Схема МПА должна, кроме того, иметь цепи начальной установки автомата в исходное состояние и цепи включения автомата на однократное выполнение алгоритма по запускающему сигналу. Кроме того в схему поступает сигнал синхронизации от генератора тактовых импульсов.

Для реализуемых в курсовой работе алгоритмов первой после начальной вершины ГСА является вершина ожидания поступления операндов с ШИВх. Единичный выход этой логической вершины и является фактически сигналом запуска автомата на однократное выполнение алгоритма. Так как при разметке ГСА начало и конец микропрограммы отмечены начальным состоянием a_0 , то автоматически происходит сброс в

начальное состояние после завершения микропрограммы. Таким образом обеспечивается многократное повторение алгоритма с поступлением следующих операндов в МПА.

Цепи начальной установки необходимы в связи с тем, что после включения питания состояния элементов памяти могут быть произвольными, а для правильного функционирования автомата его необходимо установить в начальное состояние сигналом b .

При формировании цепей начальной установки следует учитывать как код исходного состояния, так и тип триггеров, используемых в качестве ЭП.

Пусть, например, исходное состояние автомата имеет код 01. Тогда, если в качестве ЭП используются D-триггеры, то реализовать цепи начальной установки следует так, как показано на рис.18а, где D_1 и D_2 - соответствующие функции возбуждения D-триггеров. Если же в качестве

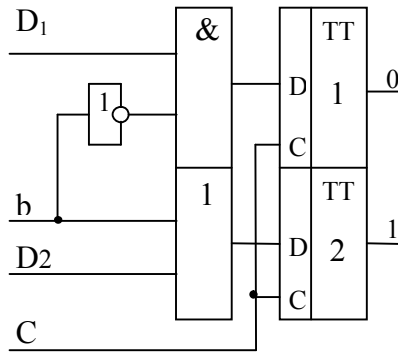


Рис.18а

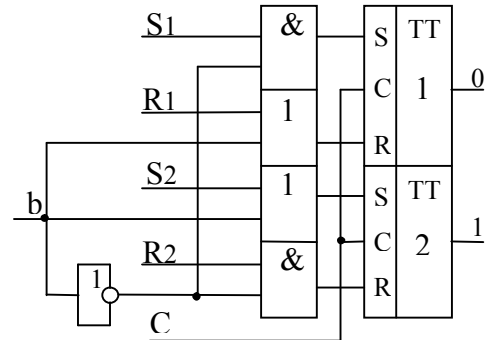


Рис.18б

ЭП используется RS-триггер, то цепи начальной установки показаны на рис.18б, где R_1S_1 , R_2S_2 - функции возбуждения для 1 и 2 триггеров.

Когда в схемах D-триггеров есть отдельные асинхронные входы установки и сброса, то их используют для подачи сигнала начальной установки, а функции возбуждения ЭП поступают на соответствующие синхронные входы триггеров.

2. Пример синтеза МПА с жёсткой логикой, управляющего операцией сложения двоичных чисел в форме с плавающей запятой

Внимание! Данный пример приведен с целью показа функционирования максимально широкого числа элементов схемотехники и не является оптимальным для реализации заданной операции. Основным требованием к курсовому проекту, выполняемого студентами, остается **минимизация аппаратных затрат** (как УА так и ОА) при приемлемом быстродействии!

Используемые коды. Операнды разрядностью 4 байта поступают по входной шине (ШИВх) в прямом коде (ПК), результат, также в ПК, выводится по выходной шине (ШИВых). При сложении использовать дополнительный код (ДК).

2.1. Выбор и описание функциональной схемы ОА

Операционный автомат должен содержать (рис.19):

- регистры RG1 и RG2 для приема операндов с ШИВх;
- регистр RG3 и триггер T1 для записи и хранения результата (T1 - знак результата);
- комбинационный сумматор SM;
- два мультиплексора MS1 и MS2 для передачи информации на плечи А и В сумматора;
- набор схем "сложение по модулю 2" для реализации инверсии;
- схема "сложение по модулю 2" для знаковых разрядов операндов;
- триггер T2 для фиксации ПРС;
- усилитель-формирователь для выдачи результата на ШИВых.

На плечо А сумматора информация поступает через мультиплексор MS1 либо с прямых, либо с инверсных выходов триггеров RG1, причем, содержимое знакового разряда подается непосредственно на сумматор под

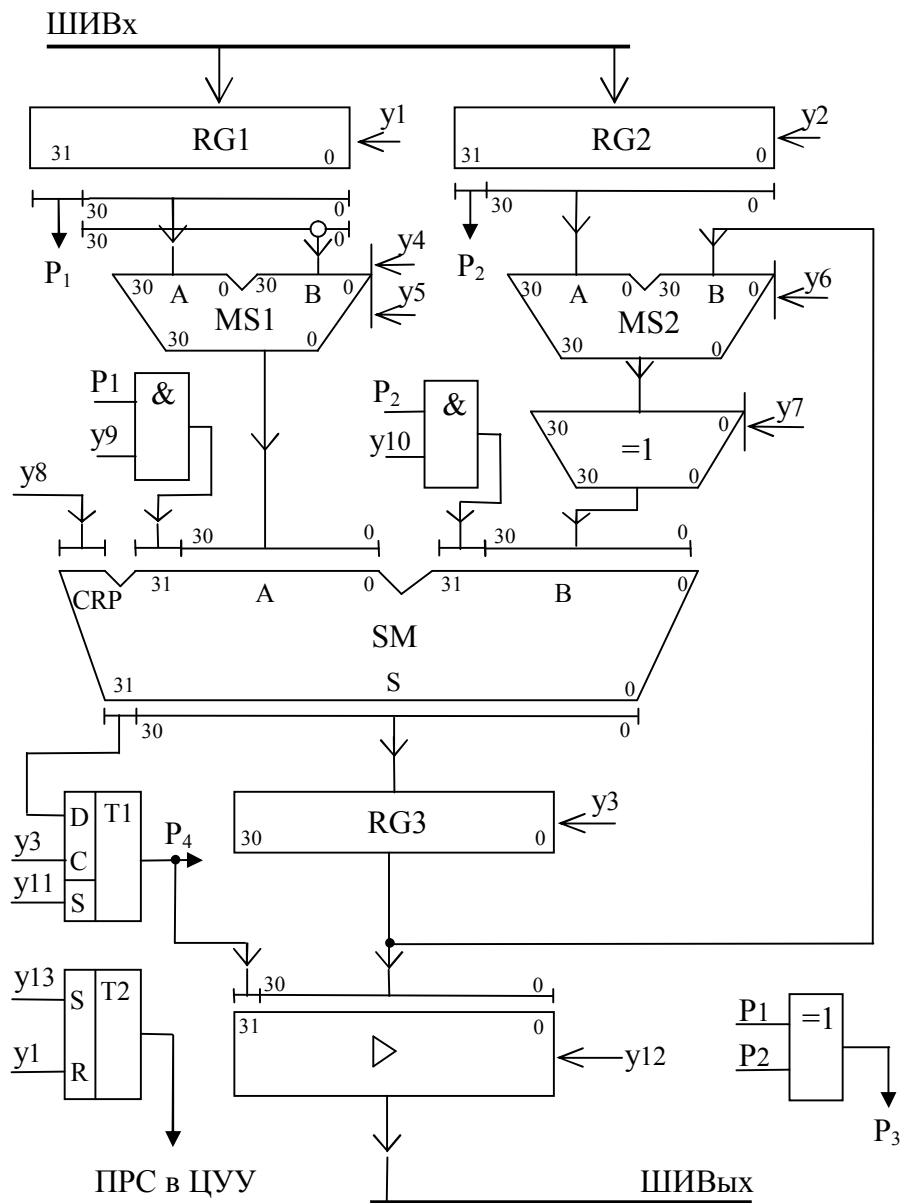


Рис.19 Функциональная схема операционного автомата

управлением сигнала y_9 . Такое решение принято в связи с тем, что, если поступают операнды одинакового знака, необходимо сложить их модули (без знаков). Если же содержимое регистра надо передать в сумматор с инверсией, то знаковые разряды не должны инвертироваться.

Следует обратить внимание на то, что мультиплексор на два входа MS1 управляется двумя сигналами - y_4 и y_5 , один из них (y_4) подключает на выход MS1 плечо В, а другой (y_5) разрешает передачу информации с выхода MS1 на плечо А сумматора. Таким образом, если сигнал y_5 не подан на MS1, то на плечо А сумматора поступают нули. Это условие ($SMA = 0$) необходимо обеспечить для правильного выполнения алгоритма (см. Вершины 9 и 15 содержательной ГСА).

На плечо В сумматора информация поступает через мультиплексор MS2: либо с RG2, либо с RG3, причем с плеча В мультиплексора информация поступает на его выход под управлением сигнала y_6 . Для реализации инверсии используется совокупность схем "сложения по модулю 2": подача сигнала y_7 позволяет передать на плечо В сумматора инверсное содержимое RG2 или RG3. Знаковый разряд RG2 передается непосредственно в сумматор под управлением сигнала y_{10} . Если управляющие сигналы y_6 и y_7 не поданы, на плечо В сумматора поступает содержимое RG2.

Для выполнения операции сложения из управляющего автомата (УА) в операционный автомат (ОА) необходимо подать управляющие сигналы, реализующие следующие микрооперации (МО).

- y_1 - запись в RG1 и сброс триггера ПРС;
- y_2 - запись в RG2;
- y_3 - запись результата в RG3 и триггер T1;
- y_4, y_5 - управление мультиплексором MS1;
- y_6 - управление мультиплексором MS2;
- y_7 - управление совокупностью схем "сложения по модулю 2";
- y_8 - $SMp=1$ - подача "1" на вход переноса сумматора;
- y_9 - $SMA[31] = RG1[31]$;
- y_{10} - $SMB[31] = RG2[31]$;

y11 - T1:= 1 - установка знака результата;

y12 - управление выдачей информации на ШИВых;

y13 - T2:=1 - установка триггера ПРС.

Из операционного автомата (ОА) в управляющий автомат (УА) необходимо передать осведомительные сигналы о состоянии устройств ОА, определяемые списком следующих логических условий.

X - проверка наличия операндов на ШИВх,

$P_1 = RG1 [31]$ - знак операнда в RG1;

$P_2 = RG2 [31]$ - знак операнда в RG2;

$P_3 = RG1 [31] \oplus RG2 [31]$ - условие на выходе сумматора "по мод. 2";

$P_4 = T1$ - знак результата;

Z - проверка возможности выдачи на ШИВых.

Таким образом, управляющий МПА должен вырабатывать 13 управляющих сигналов и посылать их в ОА в нужные такты машинного времени в соответствии с алгоритмом выполнения операции сложения, ориентируясь на 5 осведомительных сигналов, поступающих из ОА, функциональная схема которой представлена на рис.19.

2.2. Реализация содержательной ГСА

Содержательная граф-схема алгоритма представлена на рис.20. Выполнение алгоритма начинается с проверки наличия операндов на ШИВх (блоки 1 и 3). При поступлении операндов они последовательно заносятся в регистры RG1 (блок 2) и RG2 (блок 4).

Затем логическим условием $P_3 = P_1 \oplus P_2$ (блок 5) проверяется, одинаковые ($P_3=0$) или разные ($P_3=1$) знаки имеют операнды. Если знаки одинаковы, то в следующей операторной вершине (блок 6) складываются модули операндов и результат записывается в RG3. При сложении чисел одинакового знака возможно переполнение разрядной сетки (ПРС), которое

можно обнаружить в знаковом разряде $RG3$. Если $P_4=1$ (блок 7), возникло ПРС и триггер T необходимо установить в "1" подачей сигнала y_{13} на вход синхронизации D-триггера (блок 10). Если же $P_4=0$, в следующей логической вершине (блок 8) определяется знак результата: если $P_1=1$, то в

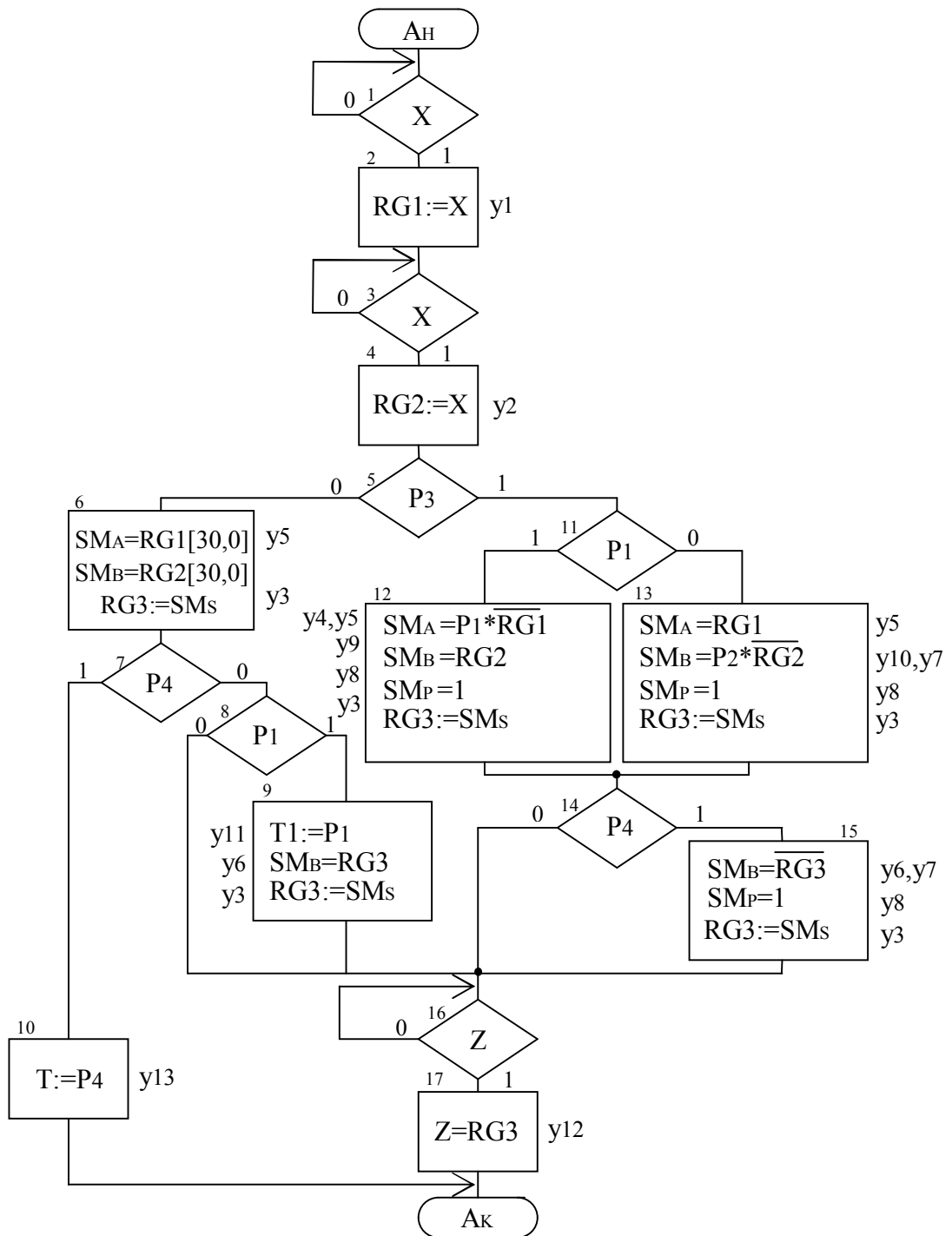


Рис.20 Содержательная граф-схема алгоритма

знаковый разряд $RG3$ заносится "1" (блок 9). И далее сформированный результат необходимо передать на ШИВых.

Когда знаки операндов различны ($P_3=1$), в логической вершине 11 проверяется знак слагаемого в $RG1$, и в зависимости от знака P_1 отрицательный операнд передается на сумматор в дополнительном коде (блоки 12 и 13).

Таблица 4

К	Совокупность МО
Y1	y1
Y2	y2
Y3	y3, y5
Y4	y3, y4, y5, y8, y9
Y5	y3, y5, y7, y8, y10
Y6	y11
Y7	y3, y6, y7, y8
Y8	y12
Y9	y13

Далее проверяется знак результата (блок 14), и если результат отрицательный ($P_4=1$), его переводят в прямой код (блок 15), и результат записывается в $RG3$.

Затем результат при $Z=1$ (блок 16) будет передан по ШИВых (блок 17) в другие устройства.

2.3. Построение отмеченной ГСА

Перед разметкой содержательной ГСА необходимо возле каждой операторной вершины проставить управляющие сигналы $y1, \dots, y13$, являющиеся выходными сигналами УА и обеспечивающие выполнение требуемых действий в соответствии со списком МО операционного автомата. Совокупности МО для каждой операторной вершины образуют микрокоманды (МК), список которых приведен в табл.3

Каждой условной вершине содержательной ГСА ставится в соответствие один из входных сигналов управляющего автомата X_1, \dots, X_5 список которых дан в табл.4.

Далее в полном соответствии с содержательной ГСА строится отмеченная ГСА (рис.21), условным вершинам которой приписывается один из входных сигналов УА (X_1, \dots, X_5), а операторным вершинам - одна из МК (в скобках указана совокупность МО для каждой МК). Выделение состояний управляющего МПА возможно в соответствии с моделью Мили или моделью Мура.

На рис.21 приведена разметка ГСА для модели Мили символами a_0, a_1, \dots, a_5 и для модели Мура - символами b_0, b_1, \dots, b_{11} . Таким образом, если строить управляющий МПА в соответствии с моделью Мили, то он будет иметь 6 состояний, а в соответствии с моделью Мура - 12 состояний.

Таблица 3

Входной сигнал УА	X_1	X_2	X_3	X_4	X_5
Логич. Условие ОА	X	P_3	P_1	P_4	Z

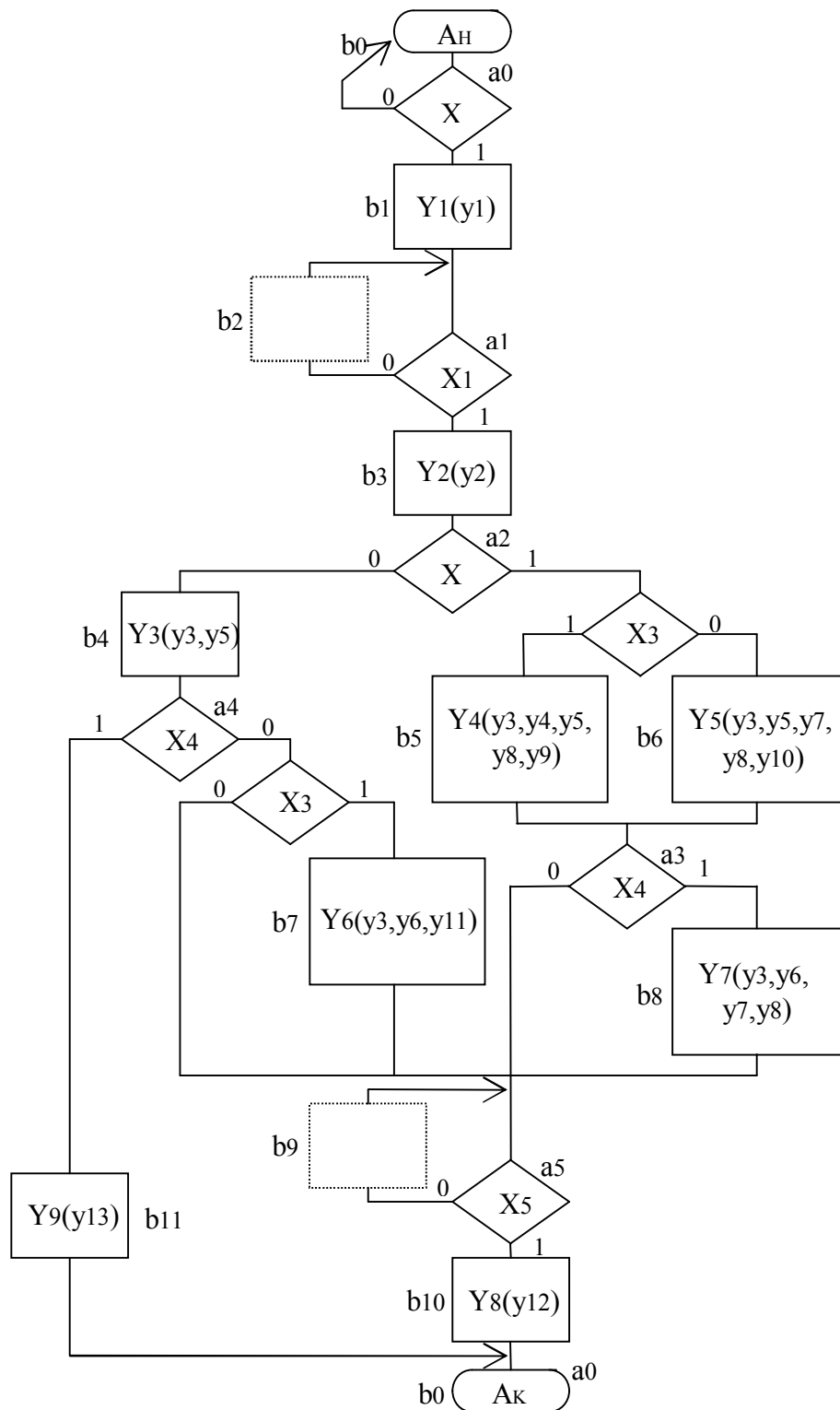


Рис.21 Отмеченная граф-схема алгоритма

Замечание. В двух вершинах ожидания (3 и 6) при разметке по Муру введены фиктивные состояния автомата b_2 и b_9 .

Явно большее число состояний для модели Мура по сравнению с моделью Мили не дает достаточных оснований для выбора модели Мили как более предпочтительной. Сравнение вариантов можно будет выполнить лишь на этапе построения функциональных схем УА, сравнив схемы по сложности и быстродействию. Поэтому далее будем вести проектирование УА параллельно для модели Мили и для модели Мура.

2.4. Построение графа автомата и структурной таблицы переходов и выходов

На основе отмеченной ГСА построены граф автомата Мура (рис.22) и граф автомата Мили (рис.23).

Граф автомата Мура имеет 12 вершин, соответствующих состояниям автомата b_0, b_1, \dots, b_{11} , каждое из которых определяет наборы выходных сигналов y_1, \dots, y_9 управляющего автомата, а дуги графа отмечены входными сигналами, действующими на данном переходе.

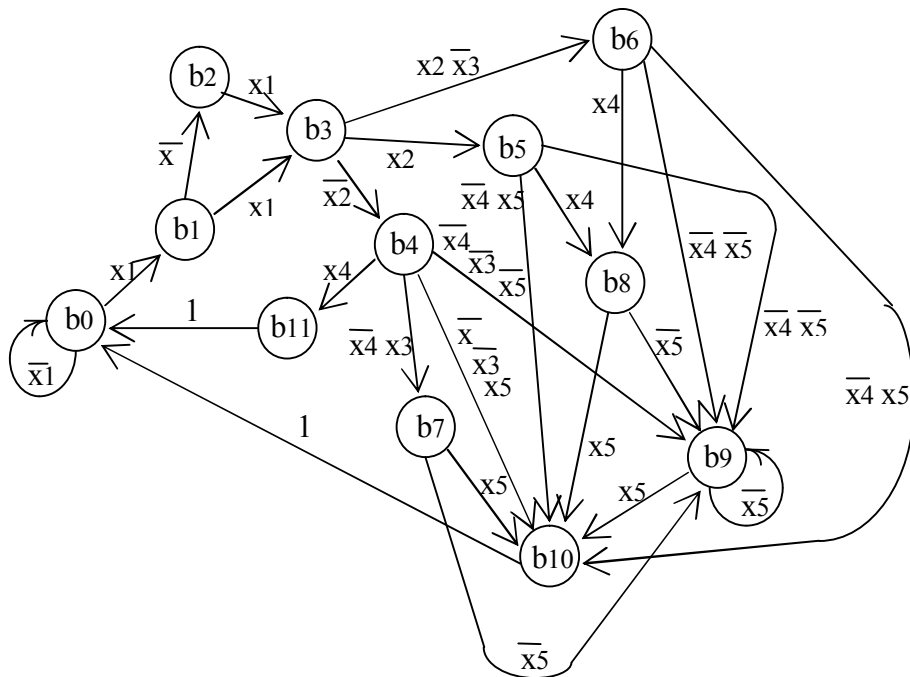


Рис.22.Граф автомата Мура.

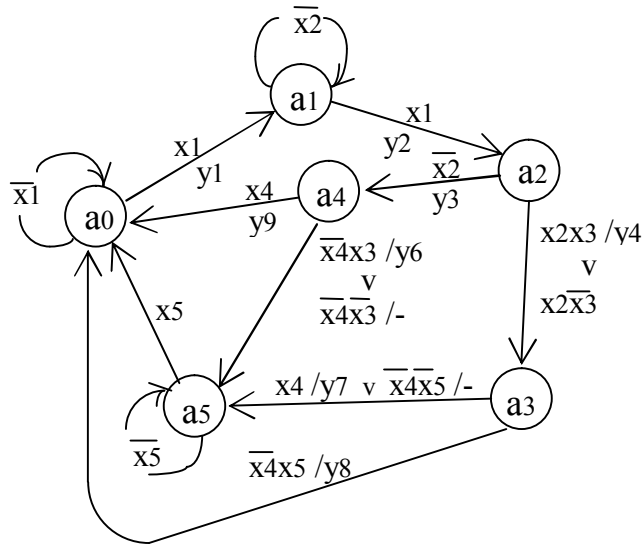


Рис.22. Граф автомата Мили

Таблица 5. Прямая структурная таблица переходов и выходов автомата Мили.

Исходное состояние e	Код a_m	Состояние перехода a_s	Код a_s	Входной сигнал $X(a_m, a_s)$	Выходные сигналы $Y(a_m, a_s)$	Функции возбуждения D-триггеров
a0	0 0 0	a0	0 0 0	$\sim X1$	-	D1
		a1	1 0 0	X1	y1	
a1	1 0 0	a1	1 0 0	$\sim X1$	-	D1
		a2	1 0 1	X1	y2	D1D3
a2	1 0 1	a3	0 0 1	X2X3	y3y4y5y8y	D3
		a3	0 0 1	X2 \sim X3	9	D3
		a4	0 1 1	\sim X2	y3y5y7y8y 10 y3y5	D2D3
a3	0 0 1	a0	0 0 0	\sim X4X5	y12	D2
		a5	0 1 0	X4	y3y6y7y8	
		a5	0 1 0	\sim X4 \sim X5	-	
a4	0 1 1	a0	0 0 0	X4	y13	D2
		a5	0 1 0	\sim X4X3	y3y6y11	
		a5	0 1 0	\sim X4 \sim X3	-	
a5	0 1 0	a0	0 0 0	X5 \sim X5	y12	D2
		a5	0 1 0	-	-	

Граф автомата Мили имеет шесть вершин, соответствующих состояниям автомата a_0, a_1, \dots, a_5 . Дуги его отмечены входными сигналами, действующими на каждом переходе (числитель), и набором выходных сигналов, вырабатываемых УА на данном переходе (знаменатель). Из приведенных рисунков видно, что с увеличением количества состояний автомата наглядность графа теряется и больше удобств представляет табличный способ задания автомата.

Логические выражения для каждой функции возбуждения D-триггера получают по таблице как конъюнкции соответствующих исходных состояний A_m и входных сигналов, которые объединены знаками дизъюнкции для всех строк, содержащих данную функцию возбуждения.

$$D1 = a_0x_1 \vee a_1,$$

$$D2 = a_2\sim x_2 \vee a_3x_4 \vee a_3\sim x_4\sim x_5 \vee a_4\sim x_4 \vee a_5\sim x_5,$$

$$D3 = a_1x_1 \vee a_2.$$

Аналогично составляются логические выражения для функций выходов.

$$y_1 = a_0x_1$$

$$y_5 = a_2,$$

$$y_9 = a_2x_2x_3$$

$$y_2 = a_1x_1$$

$$y_6 = a_3x_4 \vee a_4x_3\sim x_4,$$

$$y_{10} = a_2x_2\sim x_3$$

$$y_3 = a_2 \vee a_3x_4 \vee a_4x_3\sim x_4$$

$$y_7 = a_2x_2\sim x_3 \vee a_3x_4,$$

$$y_{11} = a_4x_3\sim x_4$$

$$y_4 = a_2x_2x_3$$

$$y_8 = a_2x_2 \vee a_3x_4,$$

$$y_{12} = a_5x_5,$$

$$y_{13} = a_4x_4,$$

После выделения общих частей в логических выражениях и некоторого их упрощения получаем логические уравнения для построения функциональной схемы управляющего автомата.

$$y_1 = a_0x_1$$

$$D1 = y_1 \vee a_1$$

$$\begin{array}{ll}
y_2 = a_1 x_1 & D_2 = a_2 \sim x_2 \vee m \vee a_3 \sim x_4 \sim x_5 \vee n \vee a_5 \sim x_5 \\
y_3 = a_2 \vee m \vee y_{11} & D_3 = y_2 \vee a_2 \\
y_4 = y_9 = k x_3 & \\
y_5 = a_2 & m = a_3 x_4 \\
y_6 = m \vee y_{11} & n = a_4 \sim x_4 \\
y_7 = m \vee y_{10} & k = a_2 x_2 \\
y_8 = m \vee k & y_{11} = n x_3 \\
y_9 = y_4 & y_{12} = a_5 x_5 \\
y_{10} = k \sim x_3 & y_{13} = a_4 x_4
\end{array}$$

Цена комбинационной схемы по Квайну для автомата Мили $C = 44$, причем в схеме предполагается использовать 3-входовой дешифратор.

В табл.6 представлена прямая структурная таблица переходов и выходов для автомата Мура. Так как каждому состоянию автомата Мура соответствует свой набор выходных сигналов, то столбец выходных сигналов в табл.6 помещен следом за столбцом исходных состояний автомата. Проанализируем вариант синтеза автомата Мура на D-триггерах, причем выберем структуру управляющего автомата на основе использования набора элементов памяти и дешифратора на 4 входа.

При кодировании состояний автомата, в качестве элементов памяти которого выбраны D-триггеры, следует стремиться использовать коды с меньшим числом "1" в кодовом слове. Для кодирования 12 состояний (b_0, b_1, \dots, b_{11}) необходимо 4 элемента памяти и из множества 4-разрядных двоичных слов надо выбрать код каждого состояния, ориентируясь на граф и таблицу переходов: чем чаще в какое-либо состояние происходят переходы из других состояний, то есть чем чаще оно встречается в столбце b_s таблицы, тем меньше в коде этого состояния следует иметь "1". Для этого

построим таблицу, в первой строке которой перечислены состояния, в которые есть более одного перехода, а во второй - состояния, из которых осуществляются эти переходы.

Таблица 6. Прямая структурная таблица переходов и выходов автомата Мура.

Исходное состояние b_m	Выходные сигналы	Код b_m	Состояние перехода b_s	Код b_s	Входной сигнал	Функции возбуждения триггеров
b0	-	1 0 0 0	b0 b1	1 0 0 0 1 1 0 1	$\sim X1$ X1	D1 D1D2D4
b1	y1	1 1 0 1	b2 b3	1 1 0 0 0 1 0 0	$\sim X1$ X1	D1D2 D2
b2	-	1 1 0 0	b3	0 1 0 0	X1	D2
b3	y2	0 1 0 0	b4 b5 b6	0 1 0 1 1 0 1 0 0 1 1 0	$\sim X2$ X2X3 X2 \sim X3	D2D4 D1D3 D2D3
b4	y3y5	0 1 0 1	b7 b9 b10 b11	0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1	$\sim X4X3$ $\sim X4\sim X3\sim X5$ $\sim X4\sim X3X5$ X4	D3D4 D4 - D1D4
b5	y3y4y5y8y9	1 0 1 0	b8 b9 b10	0 0 1 0 0 0 0 1 0 0 0 0	X4 $\sim X4\sim X5$ $\sim X4X5$	D3 D4 -
b6	y3y5y7y8y10	0 1 1 0	b8 b9 b10	0 0 1 0 0 0 0 1 0 0 0 0	X4 $\sim X4\sim X5$ $\sim X4X5$	D3 D4 -
b7	y3y6y11	0 0 1 1	b9 b10	0 0 0 1 0 0 0 0	$\sim X5$ X5	D4 -
b8	y3y6y7y8	0 0 1 0	b9 b10	0 0 0 1 0 0 0 0	$\sim X5$ X5	D4 -
b9	-	0 0 0 1	b10	0 0 0 0	X5	-
b10	y12	0 0 0 0	b0	1 0 0 0	1	D1
b11	y13	1 0 0 1	b0	1 0 0 0	1	D1

Таблица 7.

b_s	b_0	b_3	b_8	b_9	b_{10}
$\{b_m\}$	b_0, b_{10}, b_{11}	b_1, b_2	b_5, b_6	$b_4, b_5, b_6, b_7,$ b_8	$b_4, b_5, b_6, b_7, b_8,$ b_9

Наибольшее количество переходов в состояние b_{10} - закодируем его кодом $K(b_{10})=0000$. Для остальных состояний первой строки табл.7 назначим коды с одной "1": $K(b_9) = 0001$, $K(b_8) = 0010$, $K(b_0) = 1000$. Для кодирования других состояний будем использовать слова с двумя "1" в кодовом слове, стараясь, насколько возможно, использовать соседние с b_s коды для состояний, находящихся в одном столбце табл.7. Код одного из состояний имеет три "1": $K(b_1) = 1101$. Таким образом, таблица кодирования для D-триггеров такова.

Таблица 8.

b	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}
$K(b)$	1 0 0	1 1 0	1 1 0	0 1 0	0 1 0	1 0 1	0 1 1	0 0 1	0 0 1	0 0 0	0 0 0	1 0 0
)	0	1	1	0	1	0	0	1	0	1	0	1

Далее коды состояний заносят в соответствующие столбцы прямой таблицы переходов (табл.6) и по известному правилу формируют логические выражения для функции возбуждения.

$$D1 = b_0 \vee b_1 \sim x_1 \vee b_3 x_2 x_3 \vee b_4 x_4 \vee b_{10} \vee b_{11}$$

$$D2 = b_0 x_1 \vee b_1 \vee b_2 x_1 \vee b_3 \sim x_2 \vee b_3 x_2 \sim x_3$$

$$D3 = b_3 x_2 \vee b_4 x_3 \sim x_4 \vee b_5 x_4 \vee b_6 x_4$$

$$D4 = b_0 x_1 \vee b_3 \sim x_2 \vee b_4 x_3 \sim x_4 \vee b_4 \sim x_3 \sim x_4 \sim x_5 \vee b_4 x_4 \vee b_5 \sim x_4 \sim x_5 \vee b_6 \sim x_4 \sim x_5 \vee b_7 \sim x_5 \vee b_8 \sim x_5$$

Так как для автомата Мура функции выходов не зависят от входных сигналов, то в соответствии со вторым столбцом табл.6 записываем логические выражения для управляющих сигналов.

$$\begin{array}{ll}
 y_1 = b_1 & y_7 = b_8 \vee b_6 \\
 y_2 = b_3 & y_8 = b_5 \vee b_6 \vee b_8 \\
 y_3 = b_4 \vee b_5 \vee b_6 \vee b_7 \vee b_8 & y_{10} = b_6 \\
 y_4 = y_9 = b_5 & y_{11} = b_7 \\
 y_5 = b_5 \vee b_6 \vee b_4 & y_{12} = b_{10} \\
 y_6 = b_7 \vee b_8 & y_{13} = b_{11}
 \end{array}$$

После выделения общих частей в логических уравнениях и упрощений получаем окончательные выражения для построения функциональной схемы управляющего автомата.

$$\begin{array}{ll}
 n = b_4 x_3 \sim x_4 & D_1 = b_0 \vee b_1 \sim x_1 \vee p x_3 \vee k \vee b_{10} \vee b_{11} \\
 p = b_3 x_2 & D_2 = b_1 \vee b_2 x_1 \vee q \vee p \sim x_3 \\
 q = b_3 \sim x_2 & D_3 = p \vee n \vee r x_4 \\
 k = b_4 x_4 & D_4 = f \vee q \vee n \vee b_4 \sim x_3 s \vee k \vee sr \vee y_6 \\
 f = b_0 x_1 & \\
 s = \sim x_4 \sim x_5 & \\
 r = b_5 \vee b_6 & \\
 y_1 = b_1 & y_5 = b_4 \vee r & y_{10} = b_6 \\
 y_2 = b_3 & y_6 = b_7 \vee b_8 & y_{11} = b_7 \\
 y_3 = y_5 \vee y_6 & y_7 = b_6 \vee b_8 & y_{12} = b_{10} \\
 y_4 = y_9 = b_5 & y_8 = b_5 \vee y_7 & y_{13} = b_{11}
 \end{array}$$

Цена комбинационной схемы по Квайну для автомата Мура, построенного на D-триг-герах, $C = 60$.

Сравнение вариантов построения управляющего автомата по модели Мили и модели Мура показывает, что модель Мура дает комбинационную схему несколько большей сложности и требует включения дополнительного элемента памяти. Однако следует обратить внимание на то, что комбинационная схема, реализующая функции выходов автомата Мура, чрезвычайно проста (ее цена $C = 11$).

2.5. Построение функциональной схемы управляющего МПА

На рис.24 приведена функциональная схема проектируемого микропрограммного автомата, управляющего операцией сложения двоичных чисел с ФЗ.

Функциональная схема построена в основном логическом базисе И, ИЛИ, НЕ в полном соответствии с приведенной для модели Мили системой логических уравнений для функций возбуждения D-триггеров и функций выходов. В схему поступают сигналы синхронизации c и начальной установки b . Появление на одной выходной линии двух управляющих сигналов (y_4 и y_9) говорит о том, что эти сигналы идентичны и введение сигнала y_9 для ОА было излишне !

3. Тематика курсовых работ. Варианты заданий

Тематика курсовых работ должна соответствовать учебным задачам курса и основным разделам рабочей программы дисциплины "Дискретная математика и цифровые автоматы" для специальности 22.01. Задания на курсовую работу должны быть индивидуальными. В отдельных случаях возможны темы исследовательского характера.

Каждому студенту предлагается синтезировать микропрограммный

автомат, управляющий выполнением одной из арифметических или логических операций.

Исходные данные к курсовой работе выдаются руководителем на специальных бланках заданий, которые включают:

- название выполняемой операции,
 - систему счисления и форматы данных,
 - особенности алгоритма выполнения операции,
- логический базис для синтеза комбинационной схемы. Ниже в табл.9 приведены варианты типовых заданий для курсовых работ. При записи вариантов введены следующие сокращения: СС - система счисления, ПК - прямой код, ДК - дополнительный код, ОК - обратный код, ФЗ - формат с фиксированной запятой, ПЗ - формат с плавающей запятой, ВО - восстановление остатков. Способы умножения и деления пронумерованы в соответствии с нумерацией, приведенной в методических указаниях [7].

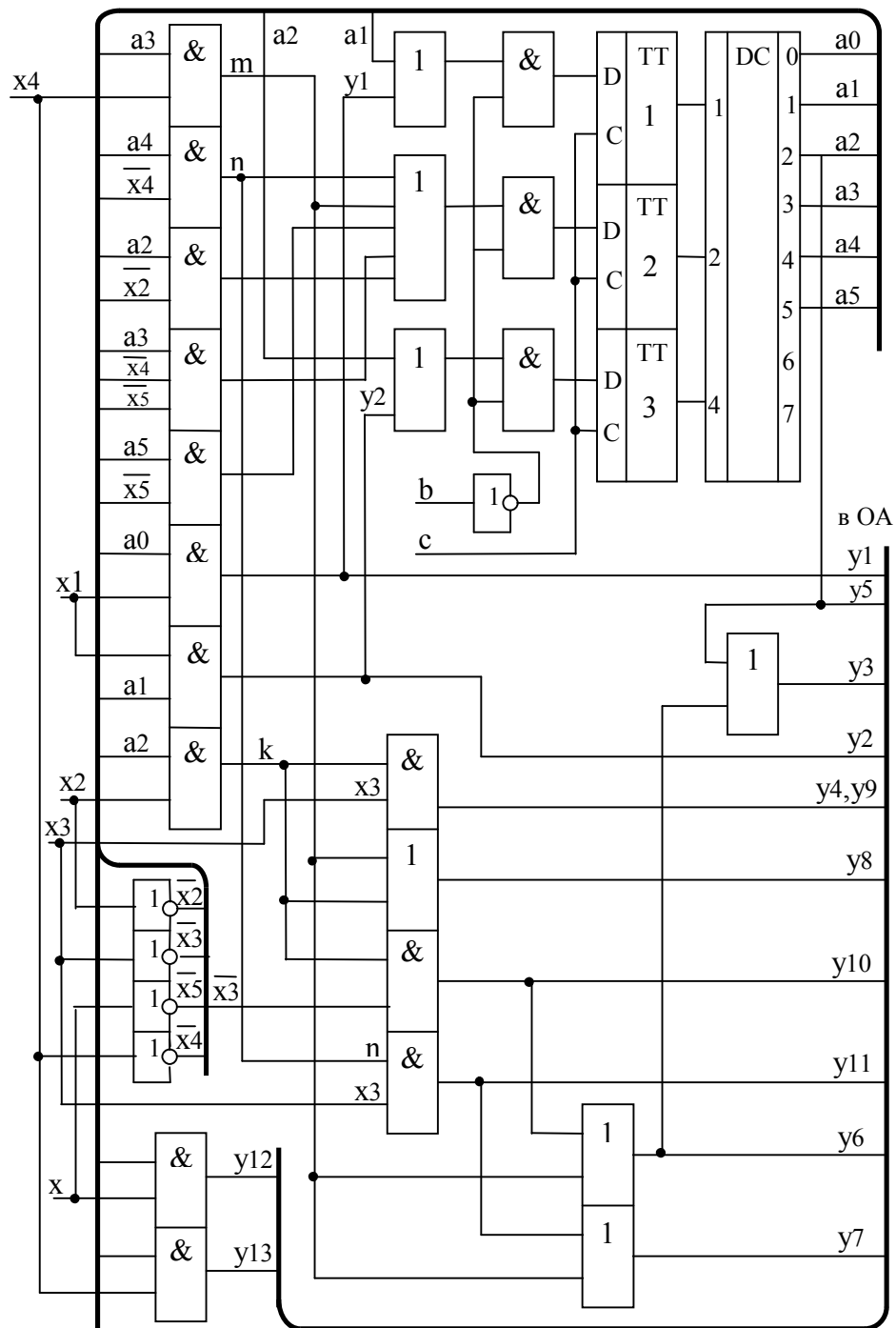


Рис.24. Функциональная схема МПА.

Таблица 9.

Номер варианта	Операция	Система счисления	Формат операндо в	Код для ФЗ	Код мантиссы для ПЗ	Порядок /характе- ристика	Способ	Особен- ности
1	2	3	4	5	6	7	8	9
1	Умнож- ие	2сс	ФЗ	ПК			I	
2	Умнож- ие	2сс	ФЗ	ПК			II	
3	Умнож- ие	2сс	ФЗ	ПК			III	
4	Умнож- ие	2сс	ФЗ	ПК			IV	
5	Умнож- ие	2сс	ФЗ	ДК			I	С прост. корр
6	Умнож- ие	2сс	ФЗ	ДК			II	-- --
7	Умнож- ие	2сс	ФЗ	ДК			III	-- --
8	Умнож- ие	2сс	ФЗ	ДК			IV	-- --
9	Умнож- ие	2сс	ФЗ	ДК			I	С автом. корр
10	Умнож- ие	2сс	ФЗ	ДК			II	-- --
11	Умнож- ие	2сс	ФЗ	ДК			III	-- --
12	Умнож- ие	2сс	ФЗ	ДК			IV	-- --
13	Умнож- ие	2сс	ПЗ		ПК	Поряд	I	
14	Умнож- ие	2сс	ПЗ		ПК	Поряд	II	
15	Умнож- ие	2сс	ПЗ		ПК	Поряд	III	
16	Умнож- ие	2сс	ПЗ		ПК	Поряд	IV	
17	Умнож- ие	2сс	ПЗ		ДК	Поряд	I	С прост. корр
18	Умнож- ие	2сс	ПЗ		ДК	Поряд	II	-- --
19	Умнож- ие	2сс	ПЗ		ДК	Поряд	III	-- --

20	Умнож- ие	2сс	ПЗ		ДК	Поряд	IV	-- --
1	2	3	4	5	6	7	8	9
21	Умнож- ие	2сс	ПЗ		ДК	Поряд	I	С автоматич. коррекцие й
22	Умнож- ие	2сс	ПЗ		ДК	Поряд	II	-- --
23	Умнож- ие	2сс	ПЗ		ДК	Поряд	III	-- --
24	Умнож- ие	2сс	ПЗ		ДК	Поряд	IV	-- --
25	Умнож- ие	2сс	ПЗ		ПК	Характ	I	
26	Умнож- ие	2сс	ПЗ		ПК	Характ	II	
27	Умнож- ие	2сс	ПЗ		ПК	Характ	III	
28	Умнож- ие	2сс	ПЗ		ПК	Характ	IV	
29	Умнож- ие	2сс	ПЗ		ДК	Характ	I	С простой коррекцие й
30	Умнож- ие	2сс	ПЗ		ДК	Характ	II	-- --
31	Умнож- ие	2сс	ПЗ		ДК	Характ	III	-- --
32	Умнож- ие	2сс	ПЗ		ДК	Характ	IV	-- --
33	Умнож- ие	2сс	ПЗ		ДК	Характ	I	С автом. корр
34	Умнож- ие	2сс	ПЗ		ДК	Характ	II	-- --
35	Умнож- ие	2сс	ПЗ		ДК	Характ	III	-- --
36	Умнож- ие	2сс	ПЗ		ДК	Характ	IV	-- --
37	Умнож- ие	2сс	ФЗ	ПК			I	С ускоре- нием 2-го порядка
38	Умнож- ие	2сс	ФЗ	ПК			II	-- --
39	Умнож- ие	2сс	ФЗ	ПК			III	-- --
40	Умнож- ие	2сс	ФЗ	ПК			IV	-- --

41	Умнож- ие	2сс	ФЗ	ПК			I	С ускоре- нием 3-го порядка
42	Умнож- ие	2сс	ФЗ	ПК			II	-- --
43	Умнож- ие	2сс	ФЗ	ПК			III	-- --
44	Умнож- ие	2сс	ФЗ	ПК			IV	-- --
1	2	3	4	5	6	7	8	9
45	Деление	2сс	ФЗ	ПК			I	С ВО и ОК
46	Деление	2сс	ФЗ	ПК			II	С ВО и ОК
47	Деление	2сс	ФЗ	ПК			I	Без ВО и ОК
48	Деление	2сс	ФЗ	ПК			II	Без ВО и ОК
49	Деление	2сс	ФЗ	ПК			I	С ВО и ОК
50	Деление	2сс	ФЗ	ПК			II	С ВО и ОК
51	Деление	2сс	ФЗ	ПК			I	Без ВО и ОК
52	Деление	2сс	ФЗ	ПК			II	Без ВО и ОК
53	Деление	2сс	ФЗ	ДК			I	С ВО
54	Деление	2сс	ФЗ	ДК			II	С ВО
55	Деление	2сс	ФЗ	ДК			I	Без ВО
56	Деление	2сс	ФЗ	ДК			II	Без ВО
57	Деление	2сс	ПЗ		ПК	Поряд	I	С ВО и ОК
58	Деление	2сс	ПЗ		ПК	Поряд	II	С ВО и ОК
59	Деление	2сс	ПЗ		ПК	Поряд	I	Без ВО и ОК
60	Деление	2сс	ПЗ		ПК	Поряд	II	Без ВО и ОК
61	Деление	2сс	ПЗ		ПК	Поряд	I	С ВО и ДК
62	Деление	2сс	ПЗ		ПК	Поряд	II	С ВО и ДК
63	Деление	2сс	ПЗ		ПК	Поряд	I	Без ВО и ДК
64	Деление	2сс	ПЗ		ПК	Поряд	II	Без ВО и ДК
65	Деление	2сс	ПЗ		ПК	Характ	I	С ВО и ОК
66	Деление	2сс	ПЗ		ПК	Характ	II	С ВО и ОК

67	Деление	2сс	ПЗ		ПК	Характ	I	Без ВО и ОК
1	2	3	4	5	6	7	8	9
68	Деление	2сс	ПЗ		ПК	Характ	II	Без ВО и ОК
69	Деление	2сс	ПЗ		ПК	Характ	I	С ВО и ДК
70	Деление	2сс	ПЗ		ПК	Характ	II	С ВО и ДК
71	Деление	2сс	ПЗ		ПК	Характ	I	Без ВО и ДК
72	Деление	2сс	ПЗ		ПК	Характ	II	Без ВО и ДК
73	Деление	2сс	ПЗ		ДК	Поряд	I	С ВО
74	Деление	2сс	ПЗ		ДК	Поряд	II	С ВО
75	Деление	2сс	ПЗ		ДК	Поряд	I	Без ВО
76	Деление	2сс	ПЗ		ДК	Поряд	II	Без ВО
77	Деление	2сс	ПЗ		ДК	Характ	I	С ВО
78	Деление	2сс	ПЗ		ДК	Характ	II	С ВО
79	Деление	2сс	ПЗ		ДК	Характ	I	Без ВО
80	Деление	2сс	ПЗ		ДК	Характ	II	Без ВО
81	Сложени е	2-10сс	ФЗ					Код 8421
82	Сложени е	2-10сс	ФЗ					Код 8421+3
83	Сложени е	2-10сс	ФЗ					Код 2421
84	Сложени е	2-10сс	ФЗ					Код 3а+2

Разрядность операндов - 4 байта. Если логический базис не задан преподавателем особо, то синтез комбинационной схемы следует выполнять в основном логическом базисе.

4. Требования к оформлению и защите курсовой работы

Курсовая работа оформляется в виде пояснительной записки и чертежей. Общий объем пояснительной записки не должен быть менее 25 - 30 листов. Пояснительная записка к курсовой работе должна отражать все основные этапы проектирования и содержать следующее.

1. Титульный лист в соответствии с образцом (Приложение1).
2. Бланк задания, подписанный руководителем и заведующим кафедрой.
3. Содержание, включающее наименования всех разделов и подразделов записки с указанием страниц.
4. Введение, отражающее содержательную сущность поставленной задачи, сравнительную оценку предлагаемого алгоритма выполнения операции по быстродействию и аппаратным затратам, область применения.
5. Основная часть включает следующие разделы.
 - 5.1. Обоснование и выбор функциональной схемы операционной части устройства. Определение списка микроопераций (МО) и логических условий (ЛУ).
 - 5.2. Разработка содержательной граф-схемы алгоритма (ГСА), которая выполняется в соответствии со словесным описанием алгоритма заданной операции и выбранной структурой операционного автомата (ОА).
 - 5.3. Разметка содержательной ГСА, составление графа автомата и прямой (или обратной) структурной таблицы переходов и выходов.
 - 5.4. Выбор и обоснование функциональной схемы управляющего автомата (УА) и типа элементов памяти (ЭП).
 - 5.5. Выбор способа кодирования состояний микропрограммного автомата (МПА), что позволяет по таблице кодов состояний завершить формирование структурной таблицы.

5.6. Получение логических уравнений для функции возбуждения ЭП и функций выходов, их минимизация.

5.7. Построение логической (функциональной) схемы управляющего МПА в заданном логическом базисе, включая цепи начальной установки, запуска и синхронизации.

5.8. Описание работы синтезированного устройства.

6. Заключение, в котором должны найти отражение основные технические характеристики синтезированного устройства, в частности, оценки сложности КС, быстродействия, а также соображения о возможных путях усовершенствования работы устройства.

7. Список использованной литературы должен содержать перечень источников, на которые в тексте пояснительной записки должны быть сделаны ссылки.

В пояснительной записке должны быть отражены все материалы, варианты, расчеты и решения, выполненные в процессе работы над заданием.

Пояснительная записка к курсовой работе пишется студентом на одной стороне листа бумаги формата 210 x 297 мм. При этом необходимо оставлять поля: слева - 30 мм, справа - 10 мм, сверху - 25 мм, снизу - 20 мм. Текст должен быть написан в соответствии со стандартами ЕСКД и правилами русского языка. Изложение должно быть четким и понятным с анализом всех альтернативных вариантов реализации того или иного пункта ПЗ.

Текст пояснительной записки делится на разделы, внутри которых можно выделять подразделы. Все иллюстрации - рисунки, таблицы, схемы - желательно располагать на отдельных страницах и помещать после той страницы пояснительной записки, где на нее дана первая ссылка.

Схемы алгоритмов следует оформлять в пояснительной записке в соответствии с требованиями ЕСПД.

Наряду с пояснительной запиской студент должен предоставить следующие чертежи:

- функциональную схему операционного автомата,
- содержательную или отмеченную ГСА,
- граф управляющего микропрограммного автомата,
- функциональную схему синтезируемого МПА.

Чертежи должны быть выполнены на чертежной бумаге формата А1 или А2, каждый должен быть заполнен на 70 - 80% общей площади и иметь рамку. При оформлении функциональных схем необходимо руководствоваться ГОСТами [6], а также руководящим нормативным документом РД 40 РСФСР-050-87.МВСОО.РСФСР, М., 1986.

Оформленная пояснительная записка и чертежи подписываются студентом и руководителем.

К защите допускаются работы, выполненные в соответствии с приведенными требованиями. Пояснительная записка и чертежи сдаются на проверку не позднее чем за два дня до предполагаемой даты защиты ! Защита проходит перед комиссией, включающей не менее 2-х преподавателей. Процедура защиты состоит из доклада на 3-5 минут и ответов на задаваемые комиссией вопросы. На защите студент должен показать знание основных понятий, определений, алгоритмов, используемых в работе, уметь обосновать выбор принятых решений, пояснить правильность функционирования приводимых схем.

Оценка за курсовую работу выставляется по результатам защиты с учетом содержания пояснительной записки, правильности и аккуратности выполнения графической части.

5. Календарный план выполнения курсовой работы

Курсовая работа рассчитана на 40-60 часов самостоятельной работы (в зависимости от подготовленности студента и темы задания). Объем работы по этапам распределяется так:

- проработка литературы по теме	5-10%
- разработка структуры операционного автомата	10%
- разработка содержательной ГСА	15%
- выбор функциональной схемы управляющего автомата	5-10%
- кодирование состояний, выбор типа ЭП	10-15%
- получение функций возбуждения и выходов и их совместная минимизация	10-15%
- построение функциональной схемы УА	10%
- оформление пояснительной записки и графической части	10-15%.

Распределение работы по неделям семестра:

- 1 неделя - выдача заданий.
- 2 неделя - 10%.
- 5 неделя - 30%.
- 7 неделя - 50%.
- 10 неделя - 80%.
- 11 неделя - 100%.
- 12-13 неделя - защита курсовой работы.

Рекомендуемая литература

1. Савельев А.Я. Прикладная теория цифровых автоматов. М.: ВШ, 1987.
2. Каган Б.М. электронные вычислительные машины и системы. М.: Энергоатомиздат, 1985.
3. Баранов С.И., Скляр В.А. Цифровые устройства на программируемых БИС с матричной структурой. М.: Радио и связь, 1986.
4. Майоров С.А., Новиков Г.И. структура электронных вычислительных машин. М.: Машиностроение, 1979.
5. Лысиков Б.Г. Арифметические и логические основы цифровых автоматов. Минск: ВМ, 1980.
6. ГОСТ 2.708-72 ЕСКД. Правила выполнения электрических схем цифровой вычислительной техники.
7. Кутепова Е.С., Фадеева Т.Р. Дискретная математика и цифровые автоматы. Двоичное умножение и деление. Методические указания к практическим занятиям. Киров: Ротапринт, 1987 - 27с.

Приложение

МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РФ
ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
КАФЕДРА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

УТВЕРЖДАЮ

ЭВМ

Зав. Кафедрой

подпись, инициалы, фамилия

"__" _____ 20__ г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по курсовой работе

"СИНТЕЗ МИКРОПРОГРАММНЫХ УПРАВЛЯЮЩИХ АВТОМАТОВ"

Специальность _____	номер, наименование
Студент группы _____	_____
_____	подпись, дата инициалы, фамилия
Руководитель _____	_____
_____	должность, дата инициалы, фамилия
Оценка _____	_____
_____	дата защиты
Комиссия: _____	(_____)
_____	инициалы, фамилия
_____	(_____)
_____	инициалы, фамилия
_____	(_____)
_____	подписи, дата инициалы, фамилия

ГОД