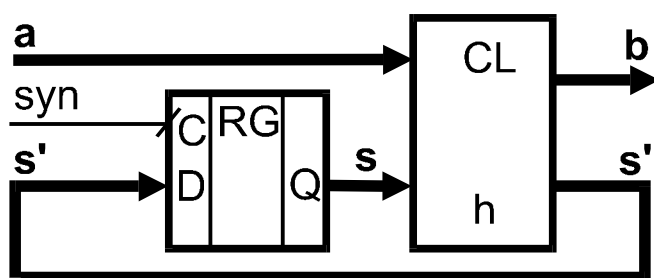


М.И. АНТИК

СИНХРОННЫЕ ЦИФРОВЫЕ АВТОМАТЫ



Москва 2006

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ
РАДИОТЕХНИКИ, ЭЛЕКТРОНИКИ И АВТОМАТИКИ
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)»

М.И. АНТИК

СИНХРОННЫЕ ЦИФРОВЫЕ АВТОМАТЫ

Москва 2006

Автор: М.И. Антик

Редактор: А.М. Романов

В пособии исследуется логический этап проектирования цифровых синхронных схем. При этом предполагается, что читатель знаком с основами логического проектирования комбинационных схем.

Печатается по решению редакционно-издательского совета Государственного образовательного учреждения высшего профессионального образования «московский государственный институт радиотехники, электроники и автоматики (технический университет)»

Рецензенты: д.т.н. А.В. Гноевой
к.т.н. А.В. Киселёв

© Государственное образовательное учреждение высшего профессионального образования «Московский государственный институт радиотехники, электроники и автоматики (технический университет)», 2004

Государственное образовательное учреждение высшего профессионального образования
«Московский государственный институт радиотехники, электроники и автоматики (технический университет)»
119454, Москва, просп. Вернадского, 78

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
ГЛАВА I. ОСНОВНЫЕ МЕТОДЫ АНАЛИЗА И СИНТЕЗА	6
1. <i>Основные понятия</i>	6
1.1. Каноническая структура синхронного цифрового автомата	6
1.2. Абстрактный конечный автомат	6
1.3. Соглашения... ..	8
2. <i>Основные структурные составляющие</i>	9
2.1. Комбинационная схема	9
2.2. Память автомата.....	10
3. <i>Основные типы синхронных автоматов</i>	10
3.1. СЦА полностью и частично определенные	11
3.2. Инициальные автоматы	11
3.3. Автомат Мура	11
3.4. Автомат Мили.....	12
4. <i>Способы описания автоматов</i>	12
4.1. Автоматная таблица	13
4.2. Автоматный граф (граф переходов)	15
4.3. Блок-схема.....	16
4.4. Блок-текст	17
5. <i>Автономные автоматы</i>	18
5.1. Параллельная композиция	18
5.2. Последовательная композиция	19
6. <i>Эквивалентные автоматы</i>	19
6.1. Изоморфные и эквивалентные автоматы.....	19
6.2. Минимальные автоматы.....	20
6.3. Эквивалентность автоматов Мура автоматам Мили	23
7. <i>Автоматы распознавания языков</i>	25
7.1. Регулярные языки	26
7.2. Асинхронный язык.....	31
7.3. Детерминизация источника.....	34
ГЛАВА II. СПЕЦИАЛЬНЫЕ МЕТОДЫ ПРОЕКТИРОВАНИЯ	38
1. <i>Декомпозиция автоматов</i>	38
2. <i>Автоматы на регистрах сдвига</i>	42
2.1. Автоматы без обратной связи	43
2.2. Автоматы с бинарной функцией обратной связи.....	47
2.3. Независимость от состояний	52
3. <i>Автоматы без потери информации</i>	52
4. <i>Условная синхронизация СЦА</i>	54
ГЛАВА III. ЛИНЕЙНЫЕ АВТОМАТЫ	55
1. <i>Основные определения</i>	55

2.	<i>Линейные автоматы без потери информации</i>	57
3.	<i>Формула полной реакции ЛА</i>	57
4.	<i>Изоморфные и эквивалентные линейные автоматы</i>	58
5.	<i>Минимальные линейные автоматы</i>	60
6.	<i>Минимально-канонические и простые канонические формы</i>	62
7.	<i>Независимость от состояний</i>	62
8.	<i>Автономные линейные автоматы (АЛА)</i>	63
8.1.	Анализ АЛА.....	64
8.2.	Синтез АЛА.....	65
9.	<i>Линейные автоматы с нулевым начальным состоянием</i>	66
9.1.	D–преобразование.....	67
9.2.	Передающая функция	69
9.3.	Связь структуры ЛА и его передающей функции.....	70
9.4.	Канонические структуры.....	72
9.5.	Эквивалентные преобразования	73
9.6.	Вычисление выходных значений.....	74
9.7.	Аннулирующие и аннигилирующие последовательности	75
10.	<i>Умножение и деление линейными автоматами</i>	78
10.1.	Умножение.....	78
10.2.	Деление	79
11.	<i>Обобщения</i>	81
11.1.	Многоканальный аналог двухполюсного ЛА.....	81
11.2.	Линейные автоматы над конечным полем	82
11.3.	Линейные автоматы над полем Галуа	83
ГЛАВА IV. УПРАВЛЯЮЩИЕ АВТОМАТЫ		84
1.	<i>Варианты взаимодействия ОА и УА</i>	86
2.	<i>Основные способы адресации микрокоманд</i>	88
2.1.	Схема с адресным ПЗУ	90
2.2.	Схема с явным указанием альтернативных адресов	92
2.3.	Схема с частичной записью адреса	93
2.4.	Схема с сокращенным тактом	94
2.5.	Схема с регулярной адресацией	96
2.6.	Схема с естественной адресацией и совмещенным назначением разрядов ячейки ПЗУ	98
2.7.	Функциональный переход и переход на микроподпрограмму с возвратом	100
2.7.1.	Функциональный переход	100
2.7.2.	Переход к микроподпрограмме с возвратом	101
3.	<i>Управление с предвосхищением</i>	102
БИБЛИОГРАФИЧЕСКИЙ СПИСОК		103

ВВЕДЕНИЕ

Синхронный цифровой автомат (СЦА) объединяет комбинационную схему и элементы памяти в единую структуру с относительно простой процедурой взаимодействия. В то же время, принципы такого взаимодействия являются базовыми для построения практически всей современной цифровой аппаратуры. Разработчик цифровой аппаратуры не может считаться достаточно грамотным, если он не владеет математическими методами анализа и синтеза синхронных автоматов. Применение таких методов гарантирует правильность и оптимальность решений по сравнению с интуитивными методами, основанными на некотором, опыте, аналогиях и т.п. Даже если решение получено на интуитивном уровне, то оно должно быть проверено, уточнено и переработано на основе точных методов. Теория дает также ответ на вопрос “что сделать нельзя?”, оставаясь в рамках структур синхронных цифровых автоматов.

Процесс проектирования цифрового устройства включает этапы логический и схемотехнический. В пособии исследуется логический этап проектирования. При этом предполагается, что читатель знаком с основами логического проектирования комбинационных схем. Схемотехническое проектирование (реализация в реально существующем элементном базисе, электрические параметры, нагрузки, помехи, быстродействие,...) в пособии не рассматривается.

Математический аппарат используется в той мере, в которой он необходим в инженерной практике анализа и синтеза синхронных автоматов.

ГЛАВА I. ОСНОВНЫЕ МЕТОДЫ АНАЛИЗА И СИНТЕЗА

1. Основные понятия

1.1. Каноническая структура синхронного цифрового автомата

Предметом изучения являются устройства, структура которых может быть представлена в виде конструкции из модулей двух типов - комбинационной схемы (КС), на рисунках обозначение – CL (combinational logic), и регистра – RG (register) с динамической синхронизацией (рис.1).

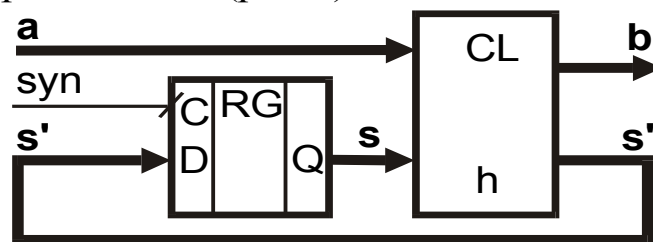


Рис.1. Синхронный автомат (схема структурная)

1.2. Абстрактный конечный автомат

В качестве основной математической модели СЦА используется абстрактный конечный автомат.

Абстрактный конечный автомат это - математическая структура с тремя основными множествами A , B , S .

A - называется множеством *входных символов*;

B - называется множеством *выходных символов*;

S - называется множеством *(символов) состояний*.

Абстрактный конечный автомат функционирует в *автоматном времени* $t=0,1,2,\dots$, определяемом как упорядоченное множество целых чисел. Это означает, что автоматное время дискретно, и существенным является лишь номер момента времени; упорядоченность множества означает, что автоматное время имеет “направление”.

Если обозначить через $a(t)$, $b(t)$, $s(t)$ переменные со значениями в соответствующих множествах в момент t , то работа абстрактного конечного автомата описывается уравнениями, которые называются каноническими

$$\begin{aligned} s(t+1) &= G(s(t), a(t)) \\ b(t) &= f(s(t), a(t)) \end{aligned}$$

G - называется *оператором перехода*. Оператор перехода G , кроме вычисления значения, выполняет также *сдвиг во времени* этого значения.

f - называется *функцией выхода*;

Принята также другая форма записи канонических уравнений:

$$\begin{aligned} \mathbf{s} &:= g(\mathbf{s}, \mathbf{a}) \\ \mathbf{b} &= f(\mathbf{s}, \mathbf{a}) \end{aligned}$$

Знак двоеточие ($:$) означает операцию сдвига значения на единицу времени или иначе – операцию запоминания; g – функцию перехода. Для фиксированного момента времени t аргумент время в записи уравнений можно опустить, а так же идентифицировать конкретные значения переменных, например с помощью индексов:

$$\begin{aligned} \mathbf{s}^j &:= g(\mathbf{s}^i, \mathbf{a}^p) \\ \mathbf{b}^k &= f(\mathbf{s}^i, \mathbf{a}^p) \end{aligned}$$

Если по каким-либо соображениям преобразование, выполняемое комбинационной схемой, удобно записать в виде одной функции, то возможны следующие формы записи канонических уравнений:

$$(\mathbf{s}; \mathbf{b}) = h(\mathbf{s}, \mathbf{a}),$$

Вычисления, выполняемые автоматом, можно рассматривать как процесс, т.е. строгую последовательность действий (шагов) преобразования символов, поступающих на \mathbf{a} вход автомата в символы, появляющиеся на его \mathbf{b} выходе.

$$\begin{aligned} \mathbf{a}(1), \mathbf{a}(2), \mathbf{a}(3), \dots, \mathbf{a}(t), \dots \\ \mathbf{b}(1), \mathbf{b}(2), \mathbf{b}(3), \dots, \mathbf{b}(t), \dots \end{aligned}$$

Каждый очередной выходной символ однозначно определяется последовательностью ранее поступивших входных символов, т.е. вычисление имеет свою “историю”. Эта история сохраняется в автомате в виде переменной – **состояние**. Состояние автомата - это та минимальная информация, которая необходима для предсказания дальнейшего поведения автомата. Поведение автомата трактуется как переходы из одного состояния в другое состояние в определенные моменты автоматного времени с одновременным вычислением выходного значения.

1.3. Соглашения ...

Для того чтобы абстрактный конечный автомат адекватно моделировал СЦА, необходимо выполнение некоторых соглашений.

Входные и выходные сигналы СЦА, в общем случае, являются двоичными многоразрядными наборами, которые кодируют символы соответствующих множеств.

Регистр в СЦА запоминает код состояния. Состояния кодируются двоичными наборами. Количество разрядов памяти, а значит и кода состояния, называется *мерностью* автомата.

КС в свою очередь реализуют функции переходов и выхода.

Ход времени в СЦА инициируется изменением значения единственного сигнала - сигнала синхронизации *syn*. На каком-либо из фронтов этого сигнала изменяется содержимое RG, а значит и состояние автомата (в схемах на рис.1 RG переключается по положительному (нарастающему) фронту сигнала *syn*). Переключающие фронты сигнала *syn* будем называть рабочими фронтами. Интервал времени между соседними рабочими фронтами называется *тактовым интервалом* или просто *тактом* (T).

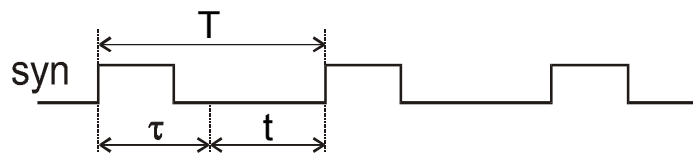


Рис.2. Структура такта

Для абстрактного автомата не имеет смысла понятие такта, как некоторого интервала “непрерывного” времени. (Применительно к абстрактному автомату можно понимать термин *такт* как изменение дискретного времени на одну единицу). Для реального синхронного цифрового автомата величина этого интервала один из важнейших технических параметров, определяющих быстродействие схемы. Величина этого интервала должна быть достаточной для того, чтобы закончились переходные процессы в КС, и установившиеся значения состояния могли быть записаны в RG. Переходные процессы начинаются из-за изменения либо состояния RG, либо изменения входных сигналов. Изменение значений на выходах RG происходит всегда на границе такта.

Для корректной работы синхронного автомата изменения входных сигналов должны происходить в интервале τ (рис.2) так, чтобы время $t=T-\tau$, было бы достаточным для завершения переходных процессов.

2. Основные структурные составляющие СЦА

2.1. Комбинационная схема

В соответствии с каноническими уравнениями можно детализировать схему СЦА на рис.1.1.-1 и представить СЦА в виде схемы на рис.2.1.-1, где одна КС представлена как декомпозиция двух КС – g и f .

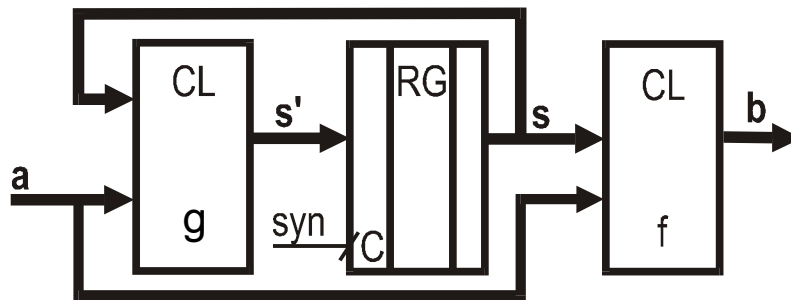


Рис.3. Функции перехода, выхода и памяти в СЦА

Одна из них - f реализует функцию выхода, другая - g вычисляет значение нового состояния, RG является *памятью*, он запоминает значение состояния, выполняя тем самым операцию сдвига (переноса) во времени.

КС, сама по себе, является частным случаем автомата - автомата без памяти или иначе автомата с одним состоянием.

При реализации СЦА декомпозиция КС должна быть доведена до комбинационных схем, вычисляющих каждый отдельный бит состояния и выходного значения.

2.2. Память автомата

Один двоичный разряд кода состояния запоминается триггером, обычно это – D-триггер, который может находиться в составе регистра.

Напомним поведение во времени D-триггера с динамиче-

ской синхронизацией (рис.4).

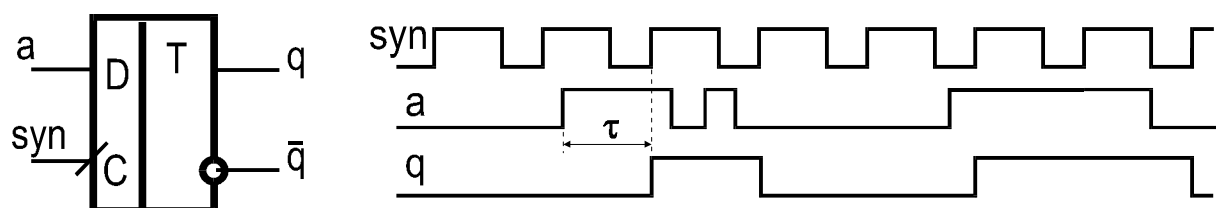


Рис.4. D-триггер (обозначение и временные диаграммы)

Триггер выдает в течение такта значение, которое он “увидел” на D-входе во время появления рабочего фронта сигнала на C-входе. Таким образом, D-триггер узнает об изменении сигнала на входе D с задержкой, которая называется *синхронизационной задержкой* (τ – на рис.4).

На изображении триггера (рис.4) не показаны входы асинхронной установки, обычно необходимые и присутствующие в реальных изделиях.

Если в качестве элементов памяти используются JK-триггеры, то для любой i -координаты кода состояния функции, вычисляющие значения на входах триггера J_i и K_i , не зависят от значения на выходе этого же триггера. (При доказательстве использовать зависимость $q := \bar{q} \cdot J \vee q \cdot \bar{K}$).

3. Основные типы синхронных автоматов

Синхронные автоматы могут быть классифицированы различным образом. Для начала определим разбиения на следующие классы синхронных автоматов:

1) полностью определенный и частично-определенный, 2) инициальный и неинициальный, а также выделим некоторый класс - 3) класс автоматов Мура.

3.1. СЦА полностью и частично определенные

Синхронный автомат *полностью определен*, если функции f и g определены на всех возможных парах из элементов множества A входных символов и множества S состояний. Количество элементов этих множеств в СЦА определяется разрядностью кодирующих двоичных наборов. Реализованный СЦА всегда полностью определен.

Если функции перехода и выхода определены не всюду, то синхронный автомат - *частично-определенный*. При этом если в каком-либо состоянии не определен переход для входного значения, то предполагается, что это значение не может появиться во входной последовательности. Если не определено выходное значение, то в этом случае оно безразлично.

3.2. Инициальные автоматы

Автомат называется *инициальным*, если в автомате выделено одно состояние, называемое *начальным* (например, s^0). Начальное состояние может быть параметром автомата; в этом случае говорят, что определено множество начальных состояний. Если нет выделенного начального состояния, то автомат *неинициальный*.

Для реализации инициального автомата схема СЦА должна быть сконструирована так, чтобы начальное состояние фиксировалось (запоминалось) в RG до начала работы автомата.

3.3. Автомат Мура

Автомат Мура - синхронный автомат, у которого значения выхода определяются только состоянием автомата в тот же дискретный момент времени. При этом КС, вычисляющая выходное значение, не связана непосредственно с входными сигналами.

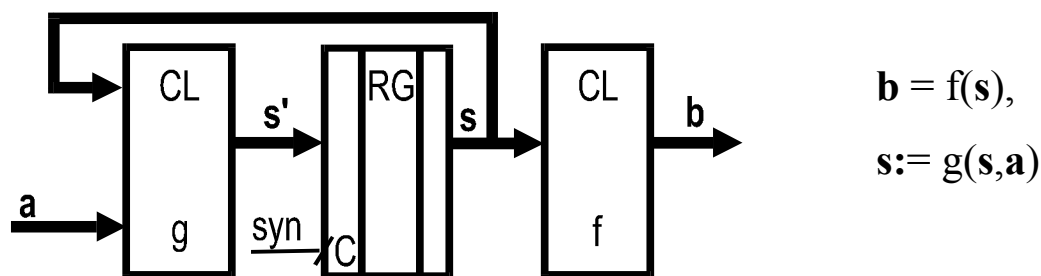


Рис.5. Автомат Мура

Поэтому момент изменения выходных сигналов зависит только от момента изменения сигнала синхронизации (рабочего фронта). О такой зависимости между входом и выходом принято говорить, что вход a и выход b “развязаны во времени”, или иначе - выход автомата “со сдвигом” зависит от входа.

3.4. Автомат Мили

Автоматы, у которых вход и выход не развязаны во време-

ни, т.е. хотя бы один выход зависит от текущего значения на входе, называют автоматами Мили.

В автомате Мили могут одновременно существовать выходные переменные, которые зависят от входных переменных, как со сдвигом, так и без сдвига, как, например, в схеме автомата, приведенной на рис.6.

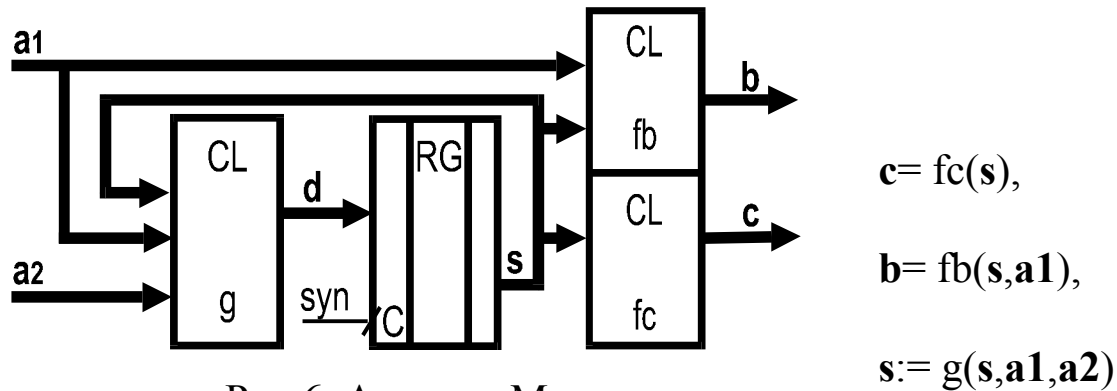


Рис.6. Автомат Мили

Переменная **c** зависит только от состояния автомата, т.е. со сдвигом от всех входных переменных. Переменная **b** зависит от переменной **a1** и состояния.

4. Способы описания автоматов

Функции f-выхода и g-переходов могут быть интерпретированы и заданы самым различным образом. Возможна следующая достаточно условная классификация способов задания функции переходов:

1) функция переходов задана как вычисляемая функция, в этом случае СЦА будем называть СЦА *операционного типа*, например $q := (q+1) \bmod N$, где q – код состояния;

2) состояние автомата интерпретируется только как идентификатор шага некоторого вычисления (алгоритма) – в этом случае СЦА будем называть СЦА *управляющего типа*.

Такая классификация условна, хотя бы потому, что в зависимости от целесообразной интерпретации один и тот же автомат может быть отнесен к любому из этих типов.

4.1. Автоматная таблица.

Функции f и g , также как и КС, могут быть заданы в виде

таблицы истинности в любой ее форме, но применительно к синхронным автоматам более других распространена форма двухвходовой таблицы, которая называется *автоматной таблицей*. Каждому состоянию автомата соответствует строка таблицы. Заголовками строк служат либо идентификаторы состояний, либо коды состояний. Каждому входному символу соответствует столбец с заголовком в виде символа или его кода. (Разумеется, можно использовать транспонированную таблицу со строками - входными символами и столбцами - состояниями.) Каждая клетка таблицы с координатами $[i, j]$ содержит, в общем случае, два значения: (b^p, s^q) , где $b^p = f(s^i, a^j)$ - выходное значение, $s^q := g(s^i, a^j)$ - новое (следующее) состояние. Таблица автомата Мура будет содержать во всех клетках одной строки одно и то же выходное значение. Поэтому изображение таблицы автомата Мура будет проще, если клетка содержит только одно значение - новое состояние, но есть столбец с выходными значениями для каждого состояния. Первая строка таблицы для инициальных автоматов, чаще всего, соответствует начальному состоянию автомата (начальному шагу алгоритма).

Пример 4.1.-1. Инкрементор. Автомат - инкрементор с одноразрядным входом (a) и одноразрядным выходом (b). На вход поступает последовательно многоразрядное число в дополнительном коде, начиная с младших разрядов. На выходе синхронно появляется результат - число на единицу большее, чем исходное.

Решение: Пусть автомат имеет два состояния $p1$ и $p0$.

$p1$ - соответствует значению переноса 1, это состояние является начальным;

$p0$ - соответствует значению переноса 0.

Тогда согласно правилам сложения одноразрядных двоичных кодов, получим таблицу 1. Таблица 2 получена из таблицы 1 присваиванием двоичных кодов состояниям. По таблице 2 могут быть получены таблицы истинности (например, в виде карты Карно) функций f и g - табл.3, табл.4.

Таблица-1		
состояние \ вход	0	1
p1	1,p0	0,p1
p0	0,p0	1,p0

Таблица-2		
	0	1
<u>1</u>	1, <u>0</u>	0, <u>1</u>
<u>0</u>	0, <u>0</u>	1, <u>0</u>

Таблица -3		
f	0	1
<u>1</u>	1	0
<u>0</u>	0	1

$$b = s + a$$

Таблица-4		
g	0	1
<u>1</u>	<u>0</u>	<u>1</u>
<u>0</u>	<u>0</u>	<u>0</u>

$$s' = s \& a$$

Схема СЦА - инкремента может быть такой, как на рис.7а.

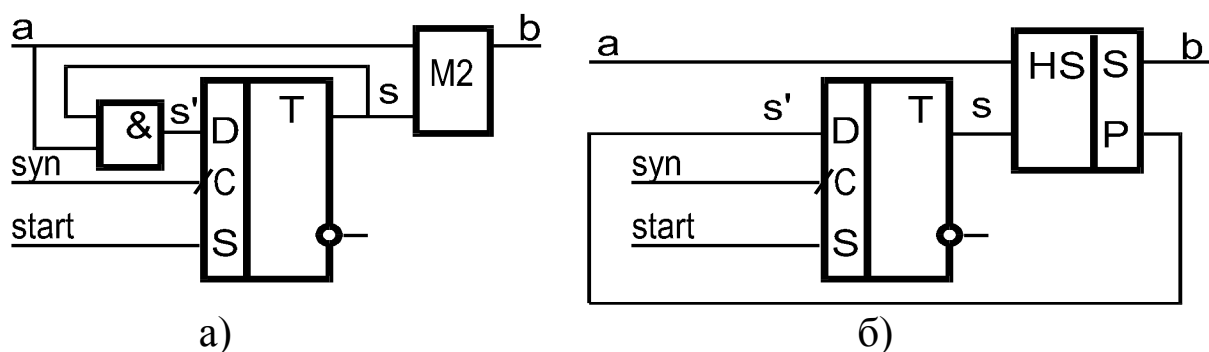


Рис.7. Инкрементор в дополнительном коде

Для правильной интерпретации работы схемы необходимо условиться о правилах (протоколе) взаимодействия схемы с внешней средой. Например: 1) начальное входное значение и начальное состояние устанавливается при $start=1$; 2) выходные значения читаются при значениях $start=0$ и $syn=0$. Если считать, что функция суммирования одноразрядных двоичных чисел задана $(p,s)=HS(x,y)$, то аналогичную схему можно получить, синтезируя СЦА как автомат операционного типа. В этом случае каноническое уравнение автомата конкретизируется как

$$(s;b)=HS(s,a) \text{ при } s(0)=1.$$

Соответствующая схема приведена на рис.7б.

4.2. Автоматный граф (граф переходов).

При описании автоматов удобно отобразить математическую структуру абстрактного автомата на другую математическую структуру - ориентированный граф.

$$G = (S, D),$$

где S - множество вершин графа,

D - множество дуг графа - упорядоченных пар вершин.

Структура графа помогает выделить два объекта математической структуры автомата: состояния - вершины графа и функцию переходов - дуги графа, остальные объекты автомата вводятся как пометки дуг или вершин. Дуги помечаются символами входного алфавита, а символами выходного алфавита помечаются дуги или вершины в зависимости от типа автомата

$G = (\{s_i\}, \{d_j[a_k/b_h]\})$ для автомата Мили,

$G = (\{s_i[b_h]\}, \{d_j[a_k]\})$ для автомата Мура.

Диаграммой будем называть какое-либо геометрическое изображение графа. Диаграмма автомата - инкрементора может иметь вид рис.8.

Дуги смежные одной и той же паре вершин и одинаково направленные (параллельные дуги) могут быть объединены в одну дугу, помеченную соответствующим образом - рис.8б (а - имя входной переменной).

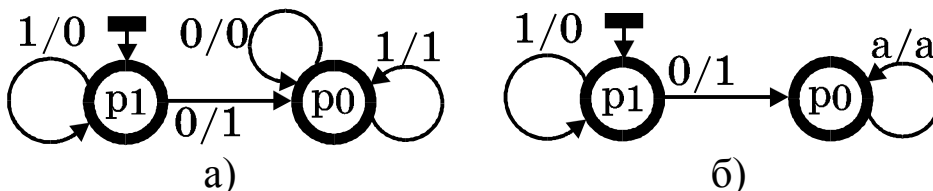


Рис.8. Диаграмма автомата - инкрементора

Автоматный граф должен удовлетворять условию однозначности, т.е. не должно существовать двух дуг, выходящих из одной и той же вершины, с одинаковыми входными пометками. Автоматный граф полностью определенного автомата удовлетворяет условию полноты, т.е. для всякой вершины s и для всякого входного символа q имеется дуга, помеченная символом q и выходящая из s .

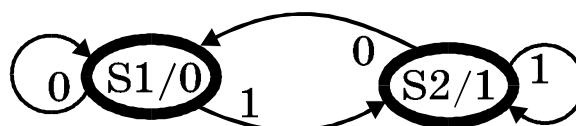


Рис.9. Автоматная диаграмма D-триггера

Примером диаграммы автомата Мура может служить авто-

матная диаграмма D-триггера (рис.9).

Отличие этой диаграммы от предыдущей в том, что выходным значением помечены не дуги, а состояния.

4.3. Блок-схема

Блок-схема автомата управляющего типа – это некоторая графическая вариация диаграммы автомата. В блок-схеме имеются вершины двух типов: 1) операторные (прямоугольные), в которых записывается выходное значение, в общем случае как функция входного значения (при этом одна операторная вершина соответствует одному состоянию автомата); 2) условные или предикатные (непрямоугольные, чаще всего ромбовидные), в которых записываются некоторые логические функции от двоичных входных переменных автомата. Выходящие из условной вершины стрелки помечаются значениями этих функций. Вершины соединяются дугами, показывающими все возможные переходы. Начальная вершина помечается.

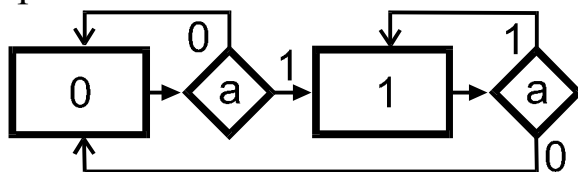


Рис.10. Блок-схема автомата D-триггера

Блок-схема удовлетворяет условию однозначности, если любой путь, соединяющий две операторные вершины, не содержит входных переменных, которые участвует в вычислении условия более одного раза. Блок-схема автомата D-триггера приведена на рис.10.

4.4. Блок-текст

Блок-текст это описание блок-схемы в виде последовательного текста. В блок-тексте используются блоки двух типов операторные и блоки переходов: операторные блоки - в скобках вида {...}, блоки перехода - в скобках вида <<...>>. И те, и другие блоки могут снабжаться метками, стоящими перед блоком. Операторный блок соответствует операторной вершине блок-схемы. В блоках перехода используется оператор GO в одной из двух форм:

GO m - безусловный переход,
 GO ($P; m_0, m_1, m_2, \dots$) - условный переход,
 где, m_0, m_1, \dots - метки блоков, P - значение, интерпретируемое оператором GO как неотрицательное целое число, являющееся порядковым номером метки в списке меток оператора GO. С этой метки должно быть продолжено выполнение алгоритма. Блоки условных переходов соответствуют условным вершинам блок-схемы. Например, см. рис.11.

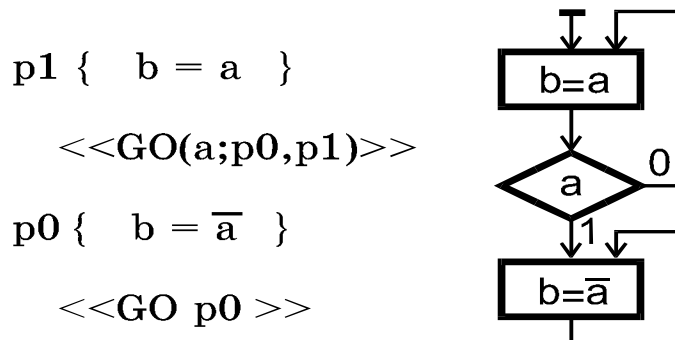


Рис.11. Блок-текст

5. Автономные автоматы

СЦА, у которого единственным изменяющимся входным сигналом является сигнал синхронизации, называется *автономным автоматом*.

Автомат, имеющий n -разрядную память, имеет 2^n состояний. Соответственно полный граф переходов автомата имеет 2^n вершин. У автономного автомата из каждой вершины выходит ровно по одной дуге. Граф автомата может иметь более одного компонента связности. В каждом компоненте связности графа автономного автомата может быть только один цикл, к этому циклу могут быть подвешены деревья, ориентированные в его сторону. Граф инициального автомата имеет только один компонент связности с числом вершин $N \leq 2^n$ (рис.12). Число состояний (вершин) в цикле инициального автономного автомата называют модулем счета.

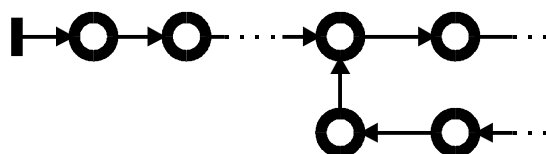


Рис.12. Диаграмма инициального автономного автомата

Любой неавтономный автомат можно сделать автономным, если зафиксировать входной символ. При этом все переходы в новое состояние осуществляются при одном и том же значении входного символа.

В инженерной практике автономные автоматы используются как счетчики и в качестве генераторов периодических последовательностей.

5.1. Параллельная композиция

Параллельная или синхронная композиция автономных автоматов приведена на рис.13. Интересны счетные возможности такой композиции, т.е. число состояний в цикле (модуль счета). Этот модуль равен наименьшему общему кратному всех модулей автоматов, входящих в синхронную композицию. Модуль будет наибольшим, если модули автоматов взаимно простые числа – тогда он будет равен их произведению.

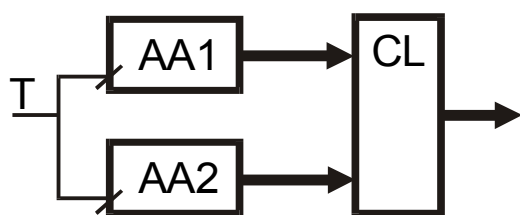


Рис.13. Параллельное включение АА

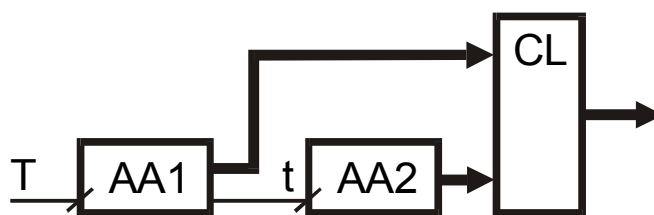


Рис.14. Последовательное включение АА

5.2. Последовательная композиция

Последовательная или асинхронная композиция синхронных автономных автоматов приведена на рис.14. Для переключения автомата АА2 используется изменение значения какого-либо одноразрядного выходного сигнала автомата АА1. Счетные возможности такой композиции максимальны, если сигнал t выбран так, что он изменяет свое значение за цикл автомата АА1 не более двух раз, т.е. переключающий фронт ровно один за цикл. Тогда модуль композиции равен произведению модулей автономных автоматов.

6. Эквивалентные автоматы

6.1. Изоморфные и эквивалентные автоматы

Автоматы *изоморфны* если их описание одинаково с точностью до переобозначений.

Два инициальных автомата будем называть *эквивалентными* автоматами, если любую одну и ту же входную последовательность они перерабатывают в одну и ту же выходную последовательность. Неинициальные автоматы будем называть эквивалентными, если для любого состояния взятого в качестве начального одного из автоматов найдется в другом автомате состояние, назначив которое начальным получим эквивалентность переработки информации входной в выходную.

6.2. Минимальные автоматы

Автомат, эквивалентный заданному и имеющий наименьшее возможное число состояний, называется *минимальным*. Если автоматы всюду определены и эквивалентны, то они имеют изоморфные минимальные автоматы. Для частично-определенных автоматов это положение может не выполняться.

Минимизация автомата возможна, если в автомате есть состояния, которые могут быть объединены в одно состояние. Такие состояния называют *эквивалентными состояниями*. Иначе говоря, автомат минимальный, если у него нет эквивалентных состояний.

Воспользуемся понятием эквивалентности автоматов и переформулируем его для состояний. Состояния s^i, s^j одного и того же автомата эквивалентны, если при подаче одной и той же (любой) входной последовательности с начальными состояниями s^i или s^j образуются одинаковые выходные последовательности.

Для эквивалентности двух состояний автомата Мили с N состояниями ($N > 1$) достаточно, чтобы совпадали реакции этих двух состояний на любые возможные входные последовательности длины, не превышающей $N-1$. Или иначе, если автомат Мили минимален, т.е. все N состояний автомата неэквивалентны, то для любой пары состояний существует входная последовательность длины не более $N-1$, различающая эти состояния.

Причем граница $N-1$ в общем случае не улучшаема см. рис.15 и рис.16.

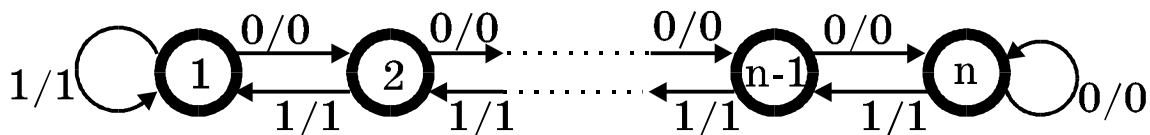


Рис.15. Автомат, у которого все состояния эквивалентны

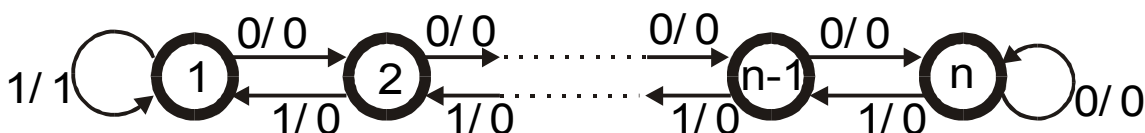


Рис.16. Автомат, у которого нет эквивалентных состояний, а для различения неэквивалентности состояний с номерами n и $(n-1)$ необходима последовательность длины $n-1$

Для автомата Мура эта же оценка равна $N-L$, где L – число выходных символов.

Достаточным условием эквивалентности состояний является совпадение соответствующих им строк в автоматной таблице. Такие состояния называют *явно эквивалентными*.

Необходимым условием эквивалентности состояний является совпадение выходных значений. Если в строках совпадают только выходные значения, то состояния, соответствующие этим строкам эквивалентны при эквивалентности новых состояний в этих строках.

За один просмотр таблицы нельзя определить в общем случае все эквивалентные состояния. Для этого нужна процедура повторных просмотров таблицы, сходящаяся к замкнутому разбиению множества состояний на классы эквивалентности. Разбиение замкнуто, если для любого состояния из одного класса эквивалентности при подаче одинакового входного воздействия автомат переходит снова в состояния из одного класса эквивалентности.

Процедуру минимизации опишем в виде алгоритма, основанного на последовательном применении необходимого условия эквивалентности. (Гилл - Введение ... с. 90-92, 3.6).

1) Явно эквивалентные между собой состояния заменяются

одним состоянием.

2) Состояниям с одинаковой комбинацией выходных значений присваивается один и тот же префикс. Таблица переписывается.

3) Состояниям с одинаковой комбинацией (выход, префикс) присваивается один и тот же префикс. Таблица переписывается.

Условие перехода: если количество различных префиксов не увеличилось или равно числу состояний, то переход к пункту 4, иначе повторяется пункт 3.

4) Различных классов эквивалентности столько же, сколько различных префиксов. Каждому классу эквивалентности в минимальном автомате сопоставляется одно состояние.

Пример 6.2-1. Автомат Мура, заданный табл.1. После первой итерации (табл.2) получаем табл.3 минимального автомата.

Таблица-1			
	выход	0	1
<u>1</u>	1	<u>2</u>	<u>3</u>
<u>2</u>	1	<u>1</u>	<u>4</u>
<u>3</u>	0	<u>4</u>	<u>1</u>
<u>4</u>	0	<u>3</u>	<u>2</u>

Таблица-2			
	выход	0	1
A- <u>1</u>	1	A- <u>2</u>	B- <u>3</u>
A- <u>2</u>	1	A- <u>1</u>	B- <u>4</u>
B- <u>3</u>	0	B- <u>4</u>	A- <u>4</u>
B- <u>4</u>	0	B- <u>3</u>	A- <u>2</u>

Таблица-3			
	выход	0	1
<u>A</u>	1	<u>A</u>	<u>B</u>
<u>B</u>	0	<u>B</u>	<u>A</u>

Пример 6.2-2. Автомат Мили, заданный табл.1. Строки автоматной таблицы - входные символы, столбцы - состояния. В результате четырех итераций (табл.2) окончательно получаем таблицу минимального автомата (табл.3).

Таблица-1								
шаг-0	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>
a	1, <u>2</u>	0, <u>1</u>	0, <u>1</u>	1, <u>5</u>	0, <u>7</u>	1, <u>5</u>	1, <u>3</u>	0, <u>7</u>
b	0, <u>2</u>	1, <u>3</u>	1, <u>2</u>	0, <u>3</u>	1, <u>8</u>	0, <u>2</u>	0, <u>3</u>	1, <u>6</u>
c	0, <u>4</u>	1, <u>3</u>	1, <u>2</u>	0, <u>3</u>	1, <u>5</u>	0, <u>7</u>	0, <u>6</u>	1, <u>6</u>

Таблица-2								
шаг-1	A- <u>1</u>	B- <u>2</u>	B- <u>3</u>	A- <u>4</u>	B- <u>5</u>	A- <u>6</u>	A- <u>7</u>	B- <u>8</u>
a	1,B- <u>2</u>	0,A- <u>1</u>	0,A- <u>1</u>	1,B- <u>5</u>	0,A- <u>7</u>	1,B- <u>5</u>	1,B- <u>3</u>	0,A- <u>6</u>
b	0,B- <u>2</u>	1,B- <u>3</u>	1,B- <u>2</u>	0,B- <u>3</u>	1,B- <u>8</u>	0,B- <u>2</u>	0,B- <u>3</u>	1,B- <u>8</u>
c	0,A- <u>4</u>	1,B- <u>3</u>	1,B- <u>2</u>	0,A- <u>1</u>	1,B- <u>5</u>	0,A- <u>7</u>	0,A- <u>6</u>	1,A- <u>6</u>
шаг-2	A- <u>1</u>	B- <u>2</u>	B- <u>3</u>	A- <u>4</u>	B- <u>5</u>	A- <u>6</u>	A- <u>7</u>	C- <u>8</u>
a	1,B- <u>2</u>	0,A- <u>1</u>	0,A- <u>1</u>	1,B- <u>5</u>	0,A- <u>7</u>	1,B- <u>5</u>	1,B- <u>3</u>	0,A- <u>6</u>
b	0,B- <u>2</u>	1,B- <u>3</u>	1,B- <u>2</u>	0,B- <u>3</u>	1,C- <u>8</u>	0,B- <u>2</u>	0,B- <u>3</u>	1,C- <u>8</u>
c	0,A- <u>4</u>	1,B- <u>3</u>	1,B- <u>2</u>	0,A- <u>1</u>	1,B- <u>5</u>	0,A- <u>7</u>	0,A- <u>6</u>	1,A- <u>6</u>

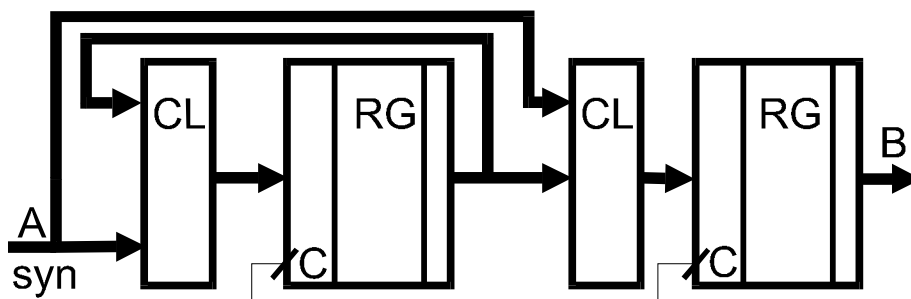
шаг-3	A-1	B-2	B-3	A-4	D-5	A-6	A-7	C-8
a	1,B-2	0,A-1	0,A-1	1,D-5	0,A-7	1,D-5	1,B-3	0,A-6
b	0,B-2	1,B-3	1,B-2	0,B-3	1,C-8	0,B-2	0,B-3	1,C-8
c	0,A-4	1,B-3	1,B-2	0,A-1	1,D-5	0,A-7	0,A-6	1,A-6
шаг-4	A-1	B-2	B-3	E-4	D-5	E-6	A-7	C-8
a	1,B-2	0,A-1	0,A-1	1,D-5	0,A-7	1,D-5	1,B-3	0,E-6
b	0,B-2	1,B-3	1,B-2	0,B-3	1,C-8	0,B-2	0,B-3	1,C-8
c	0,E-4	1,B-3	1,B-2	0,A-1	1,D-5	0,A-7	0,E-6	1,E-6
шаг-5	A	B	B	E	D	E	A	C

Таблица-3

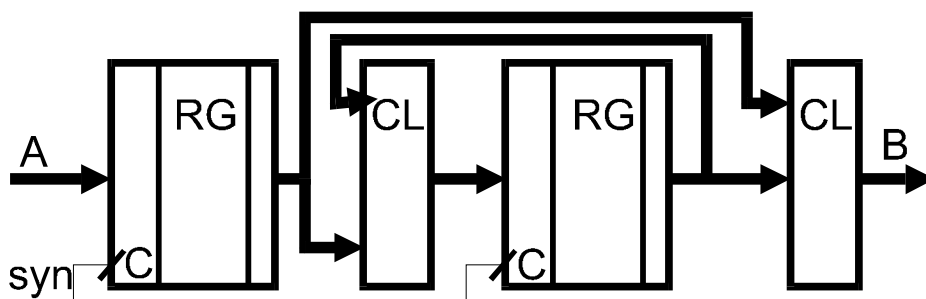
	A	B	C	D	E
a	1,B	0,A	0,E	0,A	1,D
b	0,B	1,B	1,C	1,C	0,B
c	0,E	1,B	1,E	1,D	0,A

6.3. Эквивалентность автоматов Мура автоматам Мили

6.3.1. На практике бывает удобно перейти к эквивалентному автомату, имеющему, быть может, большее число состояний, чем исходный автомат, но зато обладающему некоторыми другими “полезными” качествами.



а) автомат Мура M^A с регистром–развязкой на выходе



б) автомат Мура M^V с регистром–развязкой на входе

Рис.17. Автоматы Мура

Так обстоит дело при переходе от произвольно заданного автомата Мили к эквивалентному ему автомату Мура, который

обладает “полезным” свойством “развязывать” вход и выход. В таких случаях вместо склеивания состояний применяется в некотором смысле обратная операция “расщепления состояний”.

Автомат Мура, полученный как эквивалентный автомату Мили, выдает такие же выходные последовательности, но со сдвигом на такт, а значит выходом в начальный момент времени необходимо пренебречь.

На структурном уровне получение эквивалентного автомата Мура может выглядеть довольно просто, как некоторое структурирование памяти автомата - рис.17.

Преобразование автомата Мили в эквивалентный ему автомат Мура покажем на следующем примере.

Пример 6.3.-1. Инкрементор - рис.18. Автомат Мили.

s/a	0	1
C	1,C	0,D
D	0,D	1,D

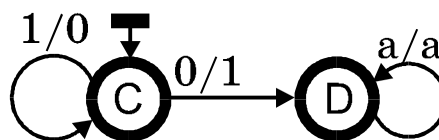


Рис.18. Таблица и диаграмма автомата Мили

В первом случае (рис.17а), состоянию s^i в автомате M (Мили), сопоставляется множество состояний $\{s^{ij}[b^j]\}$ автомата M^Δ (Мура), если есть переходы в s^i с выходом b^j ($s^i := g(x, a^k/b^j)$). Функция переходов g^Δ определяется следующим образом, если $s^q := g(s^i, a^k/b^p)$, то $s^{qp} := g^\Delta(s^{ij}, a^k)$ для каждого j .

Если автомат M инициальный с начальным состоянием s^0 , то в автомате M^Δ начальным может быть любое состояние s^{oi} . Если среди возможных начальных состояний автомата M^Δ есть состояние без входящих дуг, то такое состояние эквивалентно любому другому из возможных начальных, а его выходное значение не определено.

$s/b \backslash a$	0	1
C0	0	D1
D0	0	D0
D1	1	D0

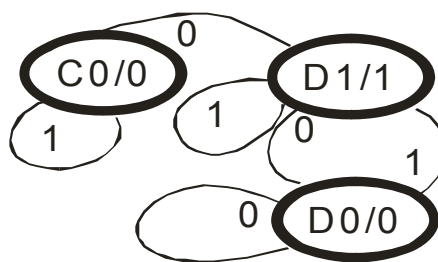


Рис.19. Таблица и диаграмма автомата Мура M^Δ

Во втором случае (рис.17б), каждой паре (a^j, s^i) автомата M (Мили) сопоставляется состояние s^{ji} автомата M^V (Мура).

Функция $s^q := g(s^i, a^j)$ трансформируется в

функцию $s^{kq} := g^V(s^{ji}, a^k)$, для каждого k .

Функция $b^p = f(s^i, a^j)$ трансформируется в

функцию $b^p = f^V(s^{ji})$.

Если автомат M инициальный с начальным состоянием s^0 , то в автомате M^V из всех состояний s^{i0} надо выбрать любое состояние с петлей ($s^{i0} := g^V(s^{i0}, a^i)$) и сделать его начальным, если такого нет, то начальным может быть любое состояние s^{i0} .

$s \backslash a$	0	1
1C	0C	1C
0C	1	0D
1D	1	0D
0D	0	0D

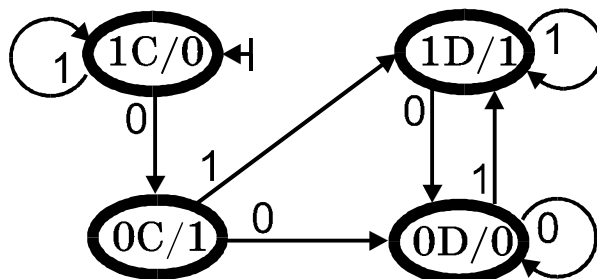


Рис.20. Таблица и диаграмма автомата Мура M^V

Минимальный автомат для того и другого случая конечно один и тот же (рис.21).

$s \backslash a$	0	1
K	0	L
L	1	N
N	0	N

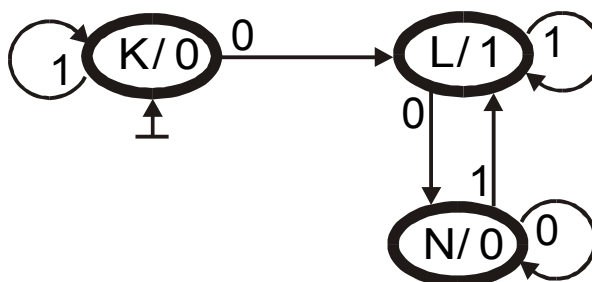


Рис.21. Таблица и диаграмма минимального автомата Мура

6.3.2. При замене автомата Мура равносильным ему автоматом Мили, сопоставляем каждой дуге $d_i[a_k]$, направленной к вершине $s_z[b_h]$ в автомате Мура дугу $d_i[a_k/b_h]$, направленную к вершине s_z в автомате Мили.

7. Автоматы распознавания языков

Синхронный автомат (любой) можно рассматривать как

устройство, распознающее некоторое множество Q слов конечной длины (само множество Q может быть бесконечным). Будем называть множество Q *языком*. Множество Q называют также *событием* - термином, который при проектировании может вызывать полезные ассоциации. Если во входной последовательности символов (букв) поступает слово, принадлежащее языку, то автомат выдает символ-индикатор этого языка.

Языки (события), которые синхронный автомат способен распознать, называют *регулярными* (*распознаваемыми, автоматными, определимыми, допускаемыми, разрешимыми, представимыми, рациональными*).

В автоматном графе инициального автомата всем событиям соответствуют те слова, которые переводят автомат из начального состояния к дугам или вершинам, помеченным выходным символом-индикатором.

Можно считать, что автономные автоматы способны распознавать слова только в однобуквенном алфавите. Распознаваемый автономным автоматом язык состоит из слов определенной длины.

7.1. Регулярные языки

Понятие “язык Q - регулярен” означает, что можно построить СЦА, распознающий язык Q .

Любое конечное множество слов конечной длины распознаваемо СЦА. В тоже время не всякий бесконечный язык регулярен, т.е. существуют языки, не распознаваемые СЦА. В некоторых случаях по описанию языка это свойство может быть установлено. Например, язык, состоящий из всех слов, в которых число нулей равно числу единиц, нерегулярен. Интуитивные соображения состоят в том, что автомат способен “считать” только до числа своих состояний, а поскольку число состояний автомата конечно, то он и не может распознать все слова из этого языка. Но в общем случае задача установления регулярности языка алгоритмически не разрешима.

В силу сказанного следует, что надо, по возможности, пользоваться средствами спецификации, обеспечивающими регуляр-

ность языка. Перечислим некоторые из способов задания регулярных языков.

7.1.1. Инициальный конечный автомат с выделенным подмножеством выходных символов I .

В автоматном графе всем словам распознаваемого языка и только словам этого языка соответствуют пути от начала к вершинам или дугам, помеченным символами из множества I . В большинстве случаев можно считать, что множество выходных символов $B = \{0, 1\}$, а множество $I = \{1\}$.

7.1.2. *Источник* - это ориентированный граф, у которого выделены начальные вершины и финальные вершины. Каждая дуга графа либо помечена символом алфавита A , либо непомечена.

Источник определяет регулярный язык Q над алфавитом A , порождаемый множеством всех путей из начальных вершин в финальные. Каждому такому пути соответствует слово, образованное символами алфавита A , являющихся пометками на дугах пути. Непомеченной (пустой) дуге соответствует пустой символ. Источник не содержит вершин, которые не достижимы из начальных, а также не содержит вершин, из которых недостижимы финальные.

В общем случае источник определяет регулярное множество слов в алфавите $A \times B$ (произведение алфавитов входного A и выходного B), при этом все вершины источника являются финальными.

В источнике может быть нарушено условие однозначности автоматного графа, т.е. из одной вершины могут выходить несколько дуг, помеченных одинаково, выходить непомеченные дуги, вообще не выходить дуг (только для финальных вершин), может быть более одной начальной вершины. Источник, у которого нарушено условие однозначности, называют также *недетерминированным* конечным автоматом.

При выполнении вычислений синхронным автоматом может существовать только один последовательный процесс, каждому шагу (такту) которого соответствует только одна вершина автоматного графа. В источнике можно, в общем случае, считать, что

так же существует один последовательный процесс, который в любом такте может находиться более чем в одной вершине источника. Источник всегда может быть преобразован (*детерминизирован*) в эквивалентный абстрактный автомат. Для этого все те вершины источника, которые в каком либо такте могли быть использованы одновременно, сопоставляются единственному состоянию автомата. Это можно сделать, если проследить все возможные варианты развития процессов в источнике. Алгоритм детерминизации см. ниже п.п.7.3.

Переход от недетерминированного автомата к соответствующему детерминированному автомату даёт увеличение числа состояний (верхняя граница 2^n , где n – число состояний недетерминированного автомата), поэтому удобно использовать недетерминированные автоматы, поскольку для представления события эта модель требует меньшего числа состояний и с более понятной структурой.

7.1.3. *Катенация (конкатенация)* языков L_1 и L_2 определяется как язык $L_1L_2 = \{ qd, \text{ где } q \text{ слово из } L_1, d \text{ слово из } L_2 \}$.

Катенация регулярных языков является регулярным языком. Пусть $L = L_1L_2$, а Q_1 и Q_2 соответственно источники, представляющие эти языки. Тогда источник Q , представляющий язык L , строится следующим образом: начальные вершины источника Q совпадают с начальными вершинами источника Q_1 ; финальные вершины источника Q совпадают с финальными вершинами источника Q_2 ; все финальные вершины Q_1 соединяются со всеми начальными вершинами Q_2 (можно это сделать через одну новую вершину).

Итерация (или катенативное замыкание) языка L является множеством L^* , которому принадлежат все слова, являющиеся катенацией любого конечного числа слов из L , а также пустое слово.

Итерация регулярного языка L является регулярным языком L^* . Если Q источник, представляющий язык L , то источник J , представляющий язык L^* , строится следующим образом: все финальные вершины соединяются со всеми начальными вершинами дугами без меток, при этом все начальные и финальные вершины

остаются теми же. Поэтому начальные вершины становятся также и финальными, и в языке появляется пустое слово.

7.1.4. Свойство быть регулярным множеством замкнуто относительно теоретико-множественных операций: объединения, пересечения, дополнения, разности, проекции, цилиндра.

1) *Объединение*. Пусть R_1 и R_2 регулярные множества, тогда объединение $R = R_1 \cup R_2$ также является регулярным множеством.

Если R_1 и R_2 представлены источниками, то источник, представляющий R , есть простое объединение этих источников такое, что множества начальных, финальных и промежуточных вершин являются объединением соответствующих вершин исходных источников с сохранением всех дуг без изменений.

Если R_1 и R_2 представлены инициальными автоматами с выходом $B = \{0, 1\}$, индикатором $I = \{1\}$, то синхронная параллельная композиция этих автоматов с общим входом и объединением выходов по **ИЛИ** будет автоматом, представляющим искомый язык.

2) *Пересечение*. Если R_1 и R_2 регулярные множества, то пересечение этих множеств $R = R_1 \cap R_2$ также регулярное множество.

Если R_1 и R_2 представлены инициальными автоматами с выходом $B = \{0, 1\}$, индикатором $I = \{1\}$, то синхронная параллельная композиция этих автоматов с общим входом и объединением выходов по **И** будет автоматом, представляющим искомый язык.

3) *Дополнение*. Если R язык над алфавитом A , то дополнение \bar{R} является языком из множества A^* всех слов в алфавите A , не принадлежащих языку R . Если R регулярное множество, то \bar{R} также регулярное множество и может быть представлено автоматом с выходным алфавитом $B = \{0, 1\}$, индикатором $I = \{1\}$ и инверсным выходом.

4) *Разность* двух множеств $R = R_1 \setminus R_2 = R_1 \cap \bar{R}_2$. Поэтому, если R_1 и R_2 регулярные множества, то регулярно также $R = R_1 \setminus R_2$.

5) *Проекция*. Для каждого слова в алфавите $Y \times Z$ с символами $yz \in Y \times Z$ его Y -проекцией называются слова с символами $y \in Y$, т.е. стирается «лишний» компонент. Если алфавит $Y \times Z$ был входным алфавитом автомата, то будет нарушено условие авто-

матности графа, автомат становится недетерминированным.

б) *Цилиндр*. Если задан язык L над алфавитом Y , то Z -цилиндром языка называется язык над алфавитом $Y \times Z$, Y -проекция которого принадлежит L . Если L регулярный язык, то его любой цилиндр также регулярный язык. Одним из «полезных» цилиндров является такой, в котором добавляемым компонентом является выходной алфавит автомата.

7.1.6. *Дефинитный (определенный)* язык может быть представлен как катенация A^*D , где A – входной алфавит, D – конечное множество слов ограниченной длины. Этот язык является бесконечным языком из слов, заканчивающихся словами из D .

Проектируя автомат, распознающий дефинитный язык, удобно сопоставить состояниям автомата все различные **начала распознаваемых слов**. Переход при поступлении нового символа осуществлять в состояние, которое соответствует одному из начал, совпадающему с хвостом нового слова, если таких совпадений более одного, то выбирается самое длинное. Одно из состояний должно соответствовать пустому началу. Будем его обозначать символом Λ .

Пример 7.1.-1. Спроектировать автомат, который устанавливает на выходе 1, если на двухразрядном входе автомата в последних 3-х тактах перед появлением кода 11 (в 4-м такте) появился только один раз код 00.

Введем, для удобства, следующие символы для обозначения комбинаций сигналов на входе автомата:

00	→	O
11	→	I
(01 или 10)	→	X
(I или X)	→	H

Автомат должен распознать в последовательности символов следующие слова: ОННІ, НОНІ, ННОІ. Этим словам соответствуют следующие различные начала: Λ , O, H, OH, HO, HH, OHN, HON, HNO. Для построения автомата Мили достаточно каждому из этих начал сопоставить свое состояние. Для построения автомата Мура надо добавить еще состояния, соответствующие сло-

вам: ОННІ, НОНІ, ННОІ. Пусть имена состояний автомата совпадают с началами слов, им сопоставленным. Тогда автомат Мура задается следующей автоматной таблицей:

<u>сост.</u>	<u>вых.</u>	<u>О</u>	<u>Х</u>	<u>І</u>
<u>Λ</u>	0	<u>О</u>	<u>Н</u>	<u>Н</u>
<u>О</u>	0	<u>О</u>	<u>ОН</u>	<u>ОН</u>
<u>Н</u>	0	<u>НО</u>	<u>НН</u>	<u>НН</u>
<u>ОН</u>	0	<u>НО</u>	<u>ОНН</u>	<u>ОНН</u>
<u>НО</u>	0	<u>О</u>	<u>НОН</u>	<u>НОН</u>
<u>НН</u>	0	<u>ННО</u>	<u>НН</u>	<u>НН</u>
<u>ОНН</u>	0	<u>ННО</u>	<u>НН</u>	<u>ОННІ</u>
<u>НОН</u>	0	<u>НО</u>	<u>ОНН</u>	<u>НОНІ</u>
<u>ННО</u>	0	<u>О</u>	<u>НОН</u>	<u>ННОІ</u>
<u>ОННІ</u>	1	<u>ННО</u>	<u>НН</u>	<u>НН</u>
<u>НОНІ</u>	1	<u>ННО</u>	<u>НН</u>	<u>ОННІ</u>
<u>ННОІ</u>	1	<u>НО</u>	<u>ОНН</u>	<u>НОНІ</u>

7.2. Асинхронный язык.

В асинхронный язык входят слова, в которых “длительность” любого из символов может быть произвольной. Точнее, если слово W принадлежит асинхронному языку, то в языке также содержатся все слова, полученные из W повторениями любых букв из W либо вычеркиванием из W некоторых повторений отдельных букв.

Функция переходов автомата, определяющего асинхронный язык, имеет следующую особенность: для любого состояния s выполняется: если $s := g(z, a)$, то $s := g(s, a)$. Автомат может перейти в другое состояние только при изменении входного символа.

Назовем *ядром* асинхронного языка язык, в котором нет повторений символов. Асинхронный язык регулярен, если ядро этого языка – регулярное множество.

В частности, если ядро – дефинитный язык, то при синтезе автомата, распознающего асинхронный язык с таким ядром можно воспользоваться приемом из пункта 7.1.6.

При реализации автоматов, распознающих асинхронный

язык, удобно использовать в кодах состояний коды входных символов, либо воспользоваться методом условной синхронизации (см.п.П.4), либо использовать то и другое.

Пример 7.2.-1. Спроектировать автомат с двухразрядным входом (in1,in2) и одноразрядным выходом, который индицирует следующее событие: после нулей на обеих линиях по каждой из линий in1 и in2 прошло ровно по одному блоку единиц любой длительности.

Решение будем искать в виде симметричной композиции двух одинаковых автоматов см. п.7.1.4.

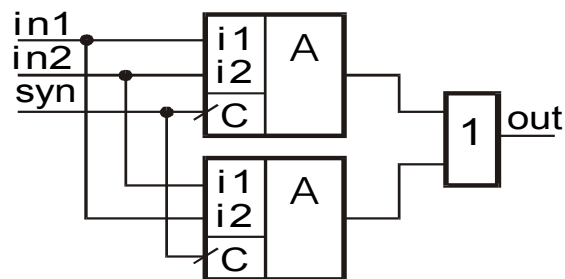


Рис.22. Композиция автоматов

Обозначим возможные символы (сочетания значений сигналов) на входах автомата следующим образом:

$$\frac{\text{in1}}{\text{in2}} \quad \frac{0}{0} = a \quad \frac{1}{0} = b \quad \frac{0}{1} = q \quad \frac{1}{1} = c$$

Если удалить все повторения символов, то получим для одного из автоматов дефинитный язык со следующими минимальными последовательностями, приводящими к событию, которое должен распознать автомат:

$$\begin{array}{lll} a \ c \ a & a \ b \ q \ a & a \ b \ a \ q \ a \\ & a \ c \ b \ a & a \ b \ c \ b \ a \\ & a \ b \ c \ a & a \ b \ c \ q \ a \end{array}$$

(Другой автомат распознает последовательности, полученные из указанных заменой b на q, q на b.)

Так же как в примере 7.1.-1, состояниям автомата сопоставим все различные начала распознаваемых слов. Эти же начала и будут именами состояний (столбец 1) в автоматной таблице 1.

Таблица 7.2.-1

1	2				3	4
имя	a	b	c	q	№	код
<u>Λ</u>	<u>a</u>	<u>Λ</u>	<u>Λ</u>	<u>Λ</u>	1	x10
<u>a</u>	S	<u>ab</u>	<u>ac</u>	<u>Λ</u>	2	a00
<u>ab</u>	<u>aba</u>	S	<u>abc</u>	<u>abq</u>	3	b00
<u>ac</u>	l, <u>a</u>	<u>acb</u>	S	<u>Λ</u>	4	c00
<u>aba</u>	S	<u>ab</u>	<u>ac</u>	<u>abaq</u>	5	a01
<u>acb</u>	l, <u>a</u>	S	<u>Λ</u>	<u>Λ</u>	6	b01
<u>abc</u>	l, <u>a</u>	<u>abcb</u>	S	<u>abcq</u>	7	c01
<u>abq</u>	l, <u>a</u>	<u>Λ</u>	<u>Λ</u>	S	8	q00
<u>abaq</u>	l, <u>a</u>	<u>Λ</u>	<u>Λ</u>	S	8	q00
<u>abcb</u>	l, <u>a</u>	S	<u>Λ</u>	<u>Λ</u>	6	b01
<u>abcq</u>	l, <u>a</u>	<u>Λ</u>	<u>Λ</u>	S	8	q00

Таблица 7.2.-2

код	a	b	c	q
b10	a00	—	c10	q10
c10	a00	b10	—	q10
q10	a00	b10	c10	—
a00	—	b00	c01	q10
b00	a01	—	c01	q00
c00	1, a00	b01	—	q10
a01	—	b00	c00	q00
b01	1, a00	—	c10	q10
c01	1, a00	b01	—	q00
q00	1, a00	b10	c10	—

Комментарии к автоматной таблице. Буква S означает переход в то же состояние в силу асинхронности языка. Использование такого символа позволяет упростить процесс минимизации состояний автомата – (эквивалентные состояния выглядят как явно эквивалентные). Чтобы не загромождать таблицу, выходное значение обозначено только там, где должно индифицироваться искомым событием. В столбце 3 строки-состояния перенумерованы так, что эквивалентные состояния имеют один и тот же номер. В столбце 4 код состояний выбран так, что бы два разряда кода

формировались кодом символа, поступившим последним. Для состояния Λ буква x означает любой символ кроме a .

Удалив эквивалентные состояния, получим автоматную таблицу 2 (состоянию Λ соответствуют первые три строки таблицы).

Реализовать автомат можно в виде схемы рис.23. Контурная часть схемы идентична выше расположенным CL—RG с иным подключением входов:

$$\begin{array}{ll} A1 \rightarrow i2 & B1 \rightarrow i4 \\ A2 \rightarrow i1 & B2 \rightarrow i3 \end{array}$$

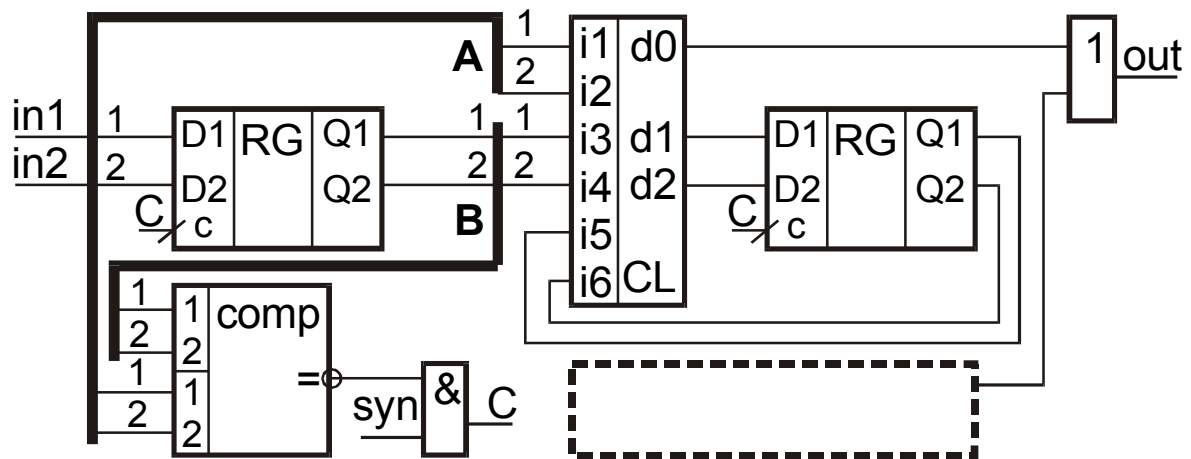


Рис.23. Реализация автомата

7.3. Детерминизация источника

Рассмотрим процедуру, позволяющую преобразовать источник в автомат Мура.

Термины: Чтобы различать сходные понятия в источнике и в автоматном графе вершины источника будем называть *узлами*, а дуги источника будем называть *стрелками*. Стрелки нумеруются. Узлам сопоставляются *имена*, а также *индексы*, состоящие из подмножества номеров стрелок. Состояниям автомата сопоставляется *шифр*, состоящий из подмножеств номеров стрелок.

Действия:

1) Нумерация стрелок. Петли без меток из источника удаляются; каждая помеченная стрелка получает уникальный номер,

начиная с 2 и далее 3,4,...

2) Таблица источника. Заготавливается таблица (пустая), в которой должны быть следующие столбцы: имена узлов, индексы узлов, остальные столбцы соответствуют всем символам входного алфавита.

3) Индексация узлов. Образуется новый единственный входной узел, который получает индекс 1. Этот узел соединяется стрелками без меток со всеми старыми входными узлами. Каждому узлу (за исключением узлов, из которых выходят только пустые стрелки) присваиваются уникальные имена, которые помещаются в таблицу. Узлам присваивается индекс, состоящий из номеров всех входящих в этот узел стрелок. Индекс узла, из которого выходит пустая стрелка, приписывается к индексу узла, в который эта стрелка входит. Индексы именованных узлов помещаются в таблицу.

4) В столбцы, именованные входными символами, записываются номера стрелок, выходящих из узла, соответствующего данной строке и данному входному символу.

5) Автоматная таблица. Заготавливается таблица (пустая), в которой должны быть следующие столбцы: шифр состояния, выходное значение, остальные столбцы соответствуют всем символам входного алфавита.

6) Заполнение строки автоматной таблицы. Первой строке, соответствующей начальному состоянию автомата, присваивается шифр 1. Начиная с этой строки, в столбцы, именованные входными символами, записываются номера стрелок, имеющих метку, совпадающую с именем столбца и выходящих из всех узлов, в индексах которых содержится какой-либо номер из шифра этого состояния (строки). Если таких стрелок нет, то записывается номер 0. В строке с шифром 0 во всех столбцах содержатся нули. (Состоянию с шифром 0 соответствует “тупиковое” состояние.)

7) Порождение новых строк. После заполнения очередной строки в столбцах, именованных входными символами, записаны шифры состояний. Если в строке появились новые шифры, не встречавшиеся ранее, то они порождают новые строки.

8) Выходные значения. Выходным значением-индикатором помечается то состояние, в шифре которого есть номер, совпадающий с номером из индекса финального узла.

Пример 7.3.-1. Воспользуемся ранее рассмотренным примером. Спроектировать автомат, который устанавливает на выходе 1, если в последних 3-х тактах на двухразрядном входе автомата перед появлением кода 11 (в 4-м такте) появился только один раз код 00.

Введем следующие обозначения для сигналов на входе автомата:

$$\begin{aligned} 00 &\rightarrow O \\ 11 &\rightarrow I \\ (01,10) &\rightarrow X \end{aligned}$$

Построим источник дефинитного языка (рис.1) и детерминируем его.

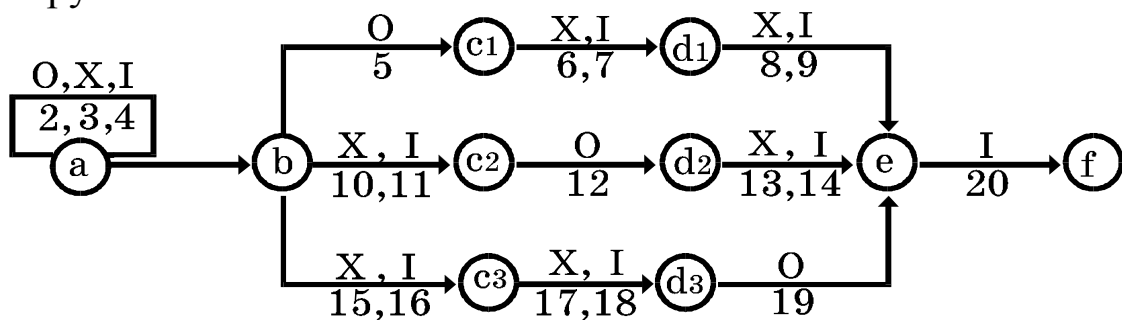


Рис.24. Источник дефинитного языка

Таблица источника:

имя	индекс	O	X	I
a	1,2,3,4	2	3	4
b	1,2,3,4	5	10,15	11,16
c1	5	--	6	7
c2	10,11	12	--	--
c3	15,16	--	17	18
d1	6,7	--	8	9
d2	12	--	13	14
d3	17,18	19	--	--
e	8,9,13,14,19	--	--	20
f	20	--	--	--

Таблица автомата:

шифр		О	Х	І	
1		2,5	3,10,15	4,11,16	A
2,5		2,5	3,6,10,15	4, 7,11,16	B
3,10,15		2,5,12	3,10,15,17	4,11,16,18	C
4,11,16		2,5,12	3,10,15,17	4,11,16,18	C
3,10,15,6		2,5,12	3,8,10,15,17	4,9,11,16,18	D
4,11,16,7		2,5,12	3,8,10,15,17	4,9,11,16,18	D
2,5,12		2,5	3,6,10,13,15	4,7,11,14,16	E
3,10,15,17		2,5,12,19	3,10,15,17	4,11,16,18	F
4,11,16,18		2,5,12,19	3,10,15,17	4,11,16,18	F
3,10,15,17,8		2,5,12,19	3,10,15,17	4,11,16,18,20	G
4,11,16,18,9		2,5,12,19	3,10,15,17	4,11,16,18,20	G
3,10,15,6,13		2,5,12	3,8,10,15,17	4,9,11,16,18,20	H
4,11,16,7,14		2,5,12	3,8,10,15,17	4,9,11,16,18,20	H
2,5,12,19		2,5	3,8,10,15,17	4,7,11, 14,16,20	J
4,11,16,18,20	1	2,5,12,19	3,10,15,17	4,11,16,18	FI
4,11,16,7,14,20	1	2,5,12	3,8,10,15,17	4,7,11,14,16,20	HI
4,11,16,18,9,20	1	2,5,12,19	3,10,15,17	4,11,16,18,20	GI

Удаляя явно эквивалентные состояния, получаем автоматную таблицу, в последнем столбце которой приведены обозначения состояний из примера 7.1.-1.

Таблица 7.3.-1

	О	Х	І	
A	B	C	C	Λ
B	B	D	D	O
C	E	F	F	H
D	E	G	G	OH
E	B	H	H	HO
F	J	F	F	HH
G	J	F	FI	OHH
H	E	G	GI	HOH
J	B	H	HI	HNO
FI,1	J	F	F	OHHI
HI,1	E	G	GI	HNOI
GI,1	J	F	FI	HOHI

ГЛАВА II. СПЕЦИАЛЬНЫЕ МЕТОДЫ ПРОЕКТИРОВАНИЯ

1. Декомпозиция автоматов

Рассмотрим два примера синтеза автоматов с применением метода декомпозиции.

Пример 1.-1. Синтезировать автомат с одноразрядным входом и одноразрядным выходом, на выходе которого фиксируется по модулю 2 количество единичных блоков с нечетным числом единиц.

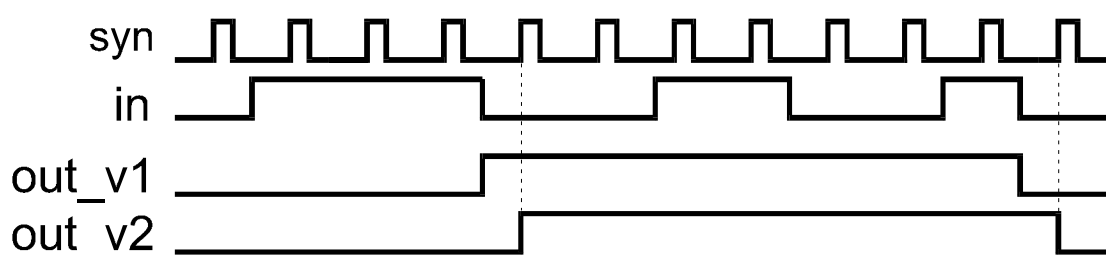


Рис.1. Временные диаграммы автомата

На временной диаграмме (рис.1) приведены два возможных варианта выхода: *out_v1* - с “привязкой” изменения к входному сигналу и *out_v2* - с “привязкой” изменения к сигналу синхронизации.

Решение 1. В этой задаче нужно считать по модулю 2, как количество блоков, так и количество единиц в блоке. Поэтому возможна следующая композиция автоматов - рис.2.

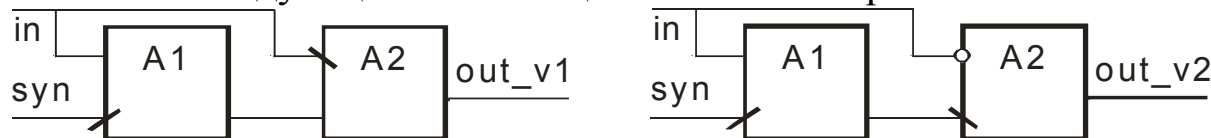


Рис.2. Декомпозиция автомата

Автомат A1 считает по модулю 2 количество единиц в блоке и сбрасывается в ноль, если нет единиц на входе. Автомат A2 счетчик по модулю 2 с параметром. В первом варианте A2 синхронизируется отрицательным фронтом входного сигнала *in*, переключается при единице на *q*; во втором варианте A2 - синхронизируется отрицательным фронтом сигнала *q*, переключается при нулевом входном сигнале *in*. Временные диаграммы (рис.3), автоматные таблицы и схема (рис.4) иллюстрируют решение. В схеме использованы JK-триггеры ради минимизации комбинационной схемы автомата.

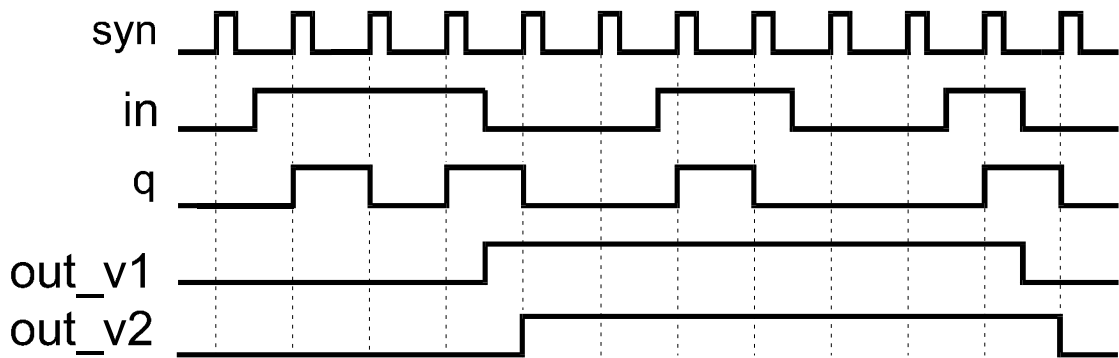


Рис.3. Временные диаграммы автоматов

Автоматные таблицы и таблицы возбуждения триггеров

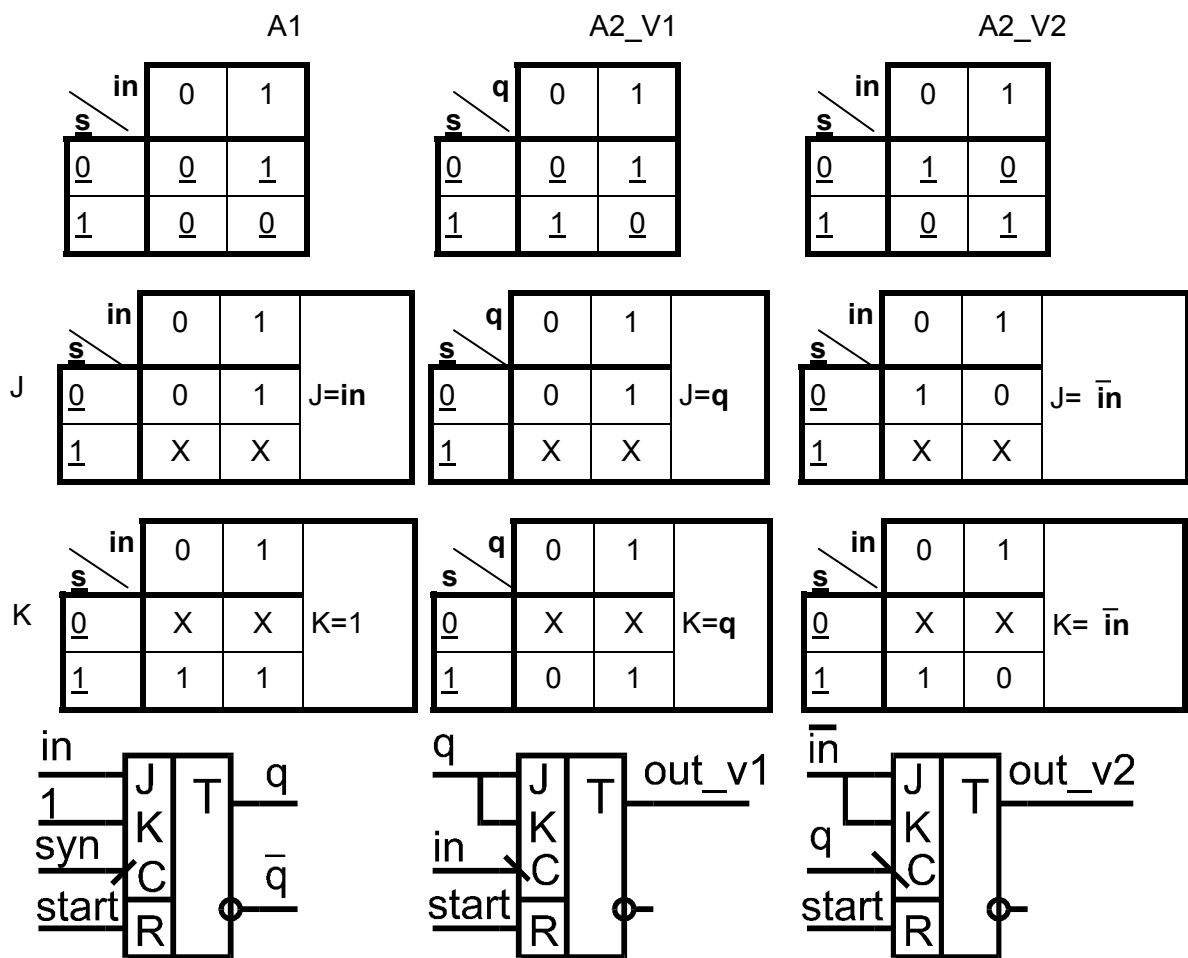


Рис.4. Схемы автоматов

Решение 2 - без использования декомпозиции. Пусть иско-
мый автомат имеет две группы состояний:

xP - состояния, в которых автомат находится, если в текущей
входной последовательности нечетное количество блоков с не-
четным числом единиц;

xR - все тоже самое, но с четным количеством блоков.

В каждую из групп входят состояния:

Kx - закончился или не начался блок единиц;

Lx - текущее количество единиц в блоке нечетно;

Mx - текущее количество единиц в блоке четно;

Чтобы получить временную диаграмму по варианту-1 (out_v1), надо синтезировать автомат Мили, а, если – по варианту-2 (out_v2), то надо синтезировать автомат Мура.

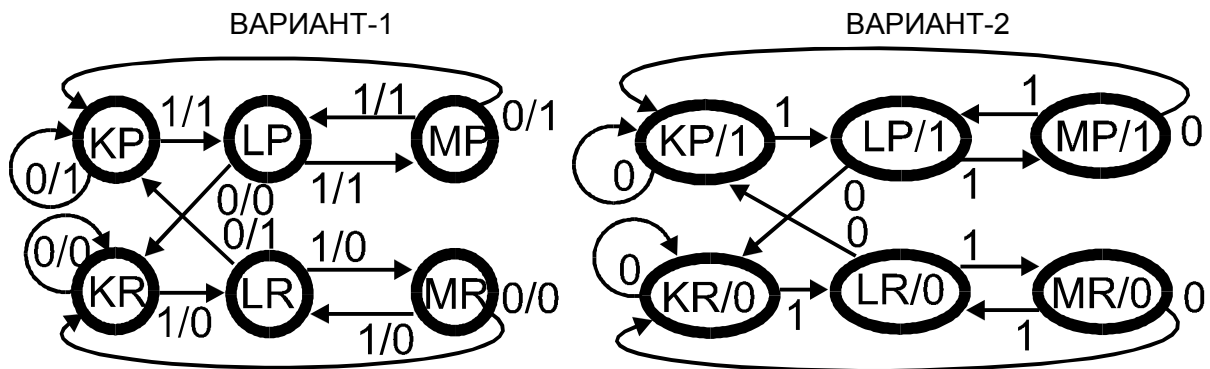


Рис.5. Диаграммы автоматов без декомпозиции

Состояния (KP,MP) и (KR,MR) явно эквивалентны, поэтому в графах останется по четыре состояния.

Если коды состояний для автомата Мура (вариант-2) выбрать так, чтобы начальное состояние (KR) имело код [00], а выходное значение совпадало со значением одного из разрядов кода, то способов кодирования состояний, приводящих к различным схемам, всего 2.

	KR	LR	KP	LP
код1	00	10	01	11
код2	00	10	11	01

После минимизации комбинационной части автомата получаем: при реализации памяти на D-триггерах переменные d_i , а для JK-триггеров – переменные (J_i, K_i) .

s_2, s_1	00	01	11	10	код1
a	00	01	00	01	
0	00	01	00	01	$(d1 = a \cdot s_2 \cdot \bar{s}_1 \vee \bar{s}_2 \cdot s_1 \vee a \cdot s_1)$ ($J1=K1=\bar{a} \cdot s_2$) $(d2 = \bar{a} \cdot s_2)$ ($J2 = a, K2=1$)
1	10	11	01	00	

$\begin{array}{c c} & s2,s1 \\ \hline a & \end{array}$	00	01	11	10	код2
0	00	00	11	11	$(d1 = \bar{a} \cdot s2 \vee a \cdot s1) \quad (J1 = \bar{a} \cdot s2, K1 = \bar{a} \cdot \bar{s}2)$
1	10	11	01	00	$(d2 = a + s2) \quad (J2 = K2 = a)$

$\begin{array}{c c} & s2,s1 \\ \hline a & \end{array}$	00	01	11	10	автомат Мили (ВАРИАНТ-1)
0	0	0	1	1	код2
1	0	1	1	0	$out_v1 = d1 = \bar{a} \cdot s2 \vee a \cdot s1$

Пример 1.-2. Синтезировать устройство, которое вычисляет максимальное из нескольких чисел. Каждое из чисел поступает по одnorазрядной шине, начиная со старших разрядов. Одновременные разряды всех чисел поступают синхронно. На одnorазрядном выходе синхронно появляется результат - значения разрядов максимального из чисел.

Для каждого отдельного числа спроектируем локальный автомат со следующими состояниями:

М - число максимальное - значение на выходе равно входному;

Н - число не максимальное - на выходе автомата 0, т.е. наименьшее возможное значение.

Теперь можно сравнить значения на выходе всех автоматов и выдать максимальное, т.е. выполнить дизъюнкцию, см. рис.6.

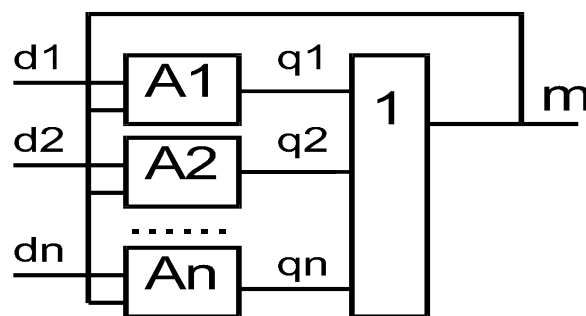


Рис.6. Структура автомата поиска максимального

Локальный автомат (A_i) переходит из состояния М в состояние Н тогда, когда число перестает быть максимальным, т.е. когда значение входного разряда числа меньше значения на выходе дизъюнкции.

Выход m , а значит и выход q_i должен зависеть без сдвига от переменной d_i . В то же время выход автомата q_i должен зависеть

со сдвигом от переменной m , поскольку это переменная обратной связи. Поэтому структура автомата должна быть такой, как на рис.7.

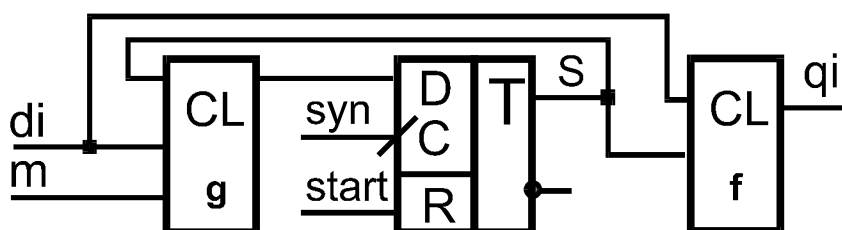


Рис.7. Схема локального автомата

Автоматная таблица:					Таблица f:			Таблица g:				
d,m	00	01	11	10	d..	0	1	d,m	00	01	11	10
s	0	1	0	1	s	0	1	s	0	1	0	1
M	0,M	0,N	1,M	X,X	0	0,	1,	0	,0	,1	,0	,X
N	0,N	0,N	0,N	0,N	1	0,	0,	1	,1	,1	,1	,1

К методам декомпозиции можно также отнести приемы, рассмотренные в гл.I, п.п. 7.1.3, 7.1.4., 7.2.

2. Автоматы на регистрах сдвига

Автоматы, память которых реализована на регистрах сдвига, имеют следующие достоинства:

1) Облегчается решение «вечной» проблемы контроля и диагностики сложных схем. (Современные схемотехнические варианты процедур диагностики связаны с объединением элементов памяти цифрового устройства в регистр сдвига. Построение автомата на регистрах сдвига позволяет уменьшить число обратных связей, что облегчает решение математической стороны той же проблемы.)

2) В ряде случаев такая структура памяти позволяет упростить комбинационную часть схемы автомата.

2.1. Автоматы без обратной связи

Назовем *индексом обратной связи* минимальное число ветвей, которые должны быть «рассечены» для того, чтобы в схеме (ориентированном графе) не было бы контуров. Нулевой индекс обратной связи означает, что обратных связей (контуров) нет. Простейшей реализацией автомата без обратных связей является

автомат с регистром сдвига на входе (рис.8), запоминающий (записывающий) входную последовательность определенной длины. Этот записанный отрезок последовательности входных символов и является текущим кодом состояния автомата.

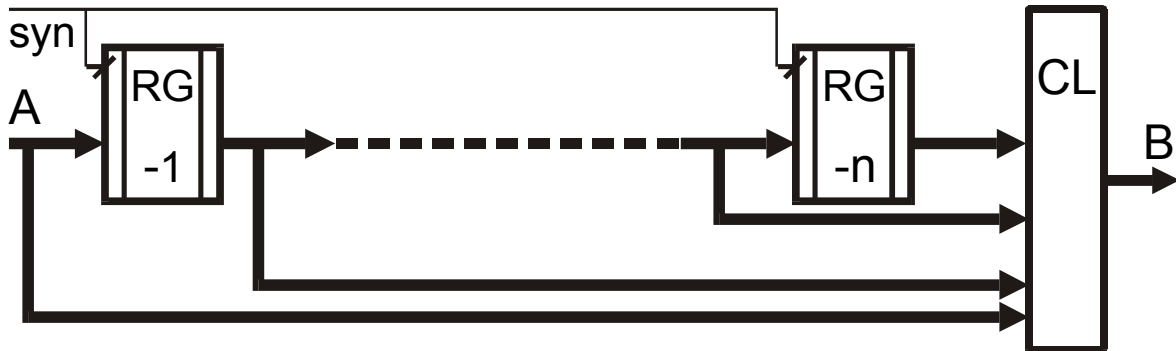


Рис.8. Реализация автомата без обратной связи

Например, таким образом можно реализовать автомат, распознающий дефинитный язык (см. гл.I.п.7.1.6).

Что бы проверить возможность такой реализации в самом общем случае, используем так называемый *универсальный парный граф переходов (УниПарГраф)*. УниПарГраф автомата **М** состоит из

- 1) вершин, соответствующих каждой неупорядоченной паре различных между собой состояний (s^i, s^j) $i < j$ автомата **М**;
- 2) дуг, помеченных входным символом a^k и направленных из вершины (s^i, s^j) к вершине (s^p, s^q) , если существуют в автомате **М** переходы по a^k либо $(s^i \rightarrow s^p) \& (s^j \rightarrow s^q)$, либо $(s^i \rightarrow s^q) \& (s^j \rightarrow s^p)$.

Утверждение. Автомат **М** может быть реализован без обратных связей тогда и только тогда, когда в УниПарГрафе автомата отсутствуют контуры.

Необходимость. Если в УниПарГрафе есть контур и входная последовательность проводит автомат по этому контуру, то какой бы длины не был отрезок входной последовательности, по нему нельзя однозначно определить финальное состояние.

Достаточность. Если в УниПарГрафе нет контуров, а самая длинная цепь составлена из k вершин, то это означает, что любой отрезок из k или более входных символов, однозначно

определяет текущее состояние автомата. Если автомат реализуется без обратных связей, то k входных символов запоминаются в регистре сдвига.

Для определения кодов состояний в реализации с регистром сдвига можно воспользоваться деревом–преемника, в котором

- вершины это–подмножества S^i множества S состояний автомата;
- корень – соответствует множеству S всех состояний автомата;

для каждого входного значения a^h строится ветвь, направленная от S^i к узлу преемнику, представляющему множество всех следующих состояний, если исходное состояние включено в S^i и приложен вход a^h .

Не обязательно строить полное дерево с k уровнями, где k число вершин в самой длинной цепи в УниПарГрафе. Информация о преемниках будет достаточной, если заканчивать дерево вершинами идентичными вершинам более раннего уровня и вершинами, состоящими из одного состояния.

Дерево–преемника удобнее представлять в табличном виде, см. пример ниже.

Одному состоянию автомата могут соответствовать более одного кода. Интерпретировать это можно двояко либо как многозначное кодирование одного состояния, либо как существование в автомате эквивалентных состояний.

Пример 2.1.-1. Автомат задан автоматной таблицей (выходные значения не указаны).

Автоматная таблица

	a	b	c
1	1	5	5
2	1	4	5
3	2	5	5
4	3	4	5
5	3	4	6
6	3	4	6

Таблица дерева–преемника

	a	b	c
123456	123	45	56
123	12	45	5
45	3	4	56
56	3	4	6
12	1	45	5

После построения УниПарГрафа ясно, что автомат можно реализовать без обратных связей. При такой реализации входной

регистр сдвига должен запоминать 5 входных символов.

После построения дерева-преемника рис.10 для определения кодов состояний, последовательно увеличивая число информативных разрядов, до тех пор, когда состояния станут различными, получаем все варианты кодов каждого из состояний.

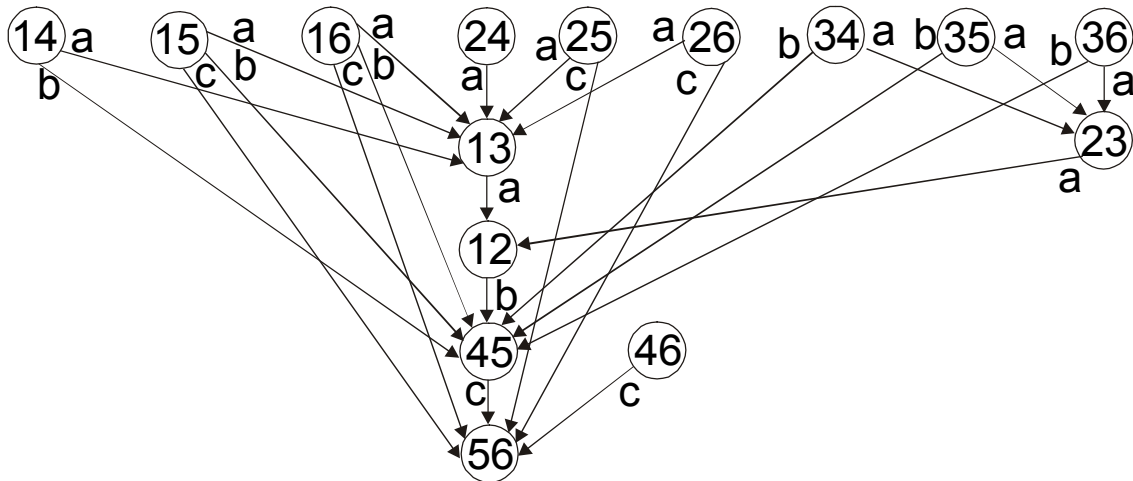


Рис.9. Универсальный парный граф

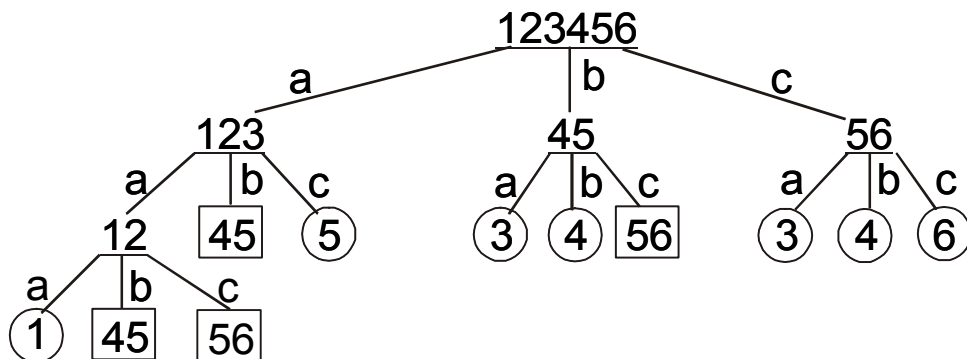


Рис.10. Дерево-преемника

a	123
b	45
c	56

aa	a123	12
ab	a45	3
ac	a56	3
ba	b123	45
bb	b45	4
bc	b56	4
ca	c123	5
cb	c45	56
cc	c56	6

aaa	a12	1
aab	a3	2
aac	a3	2
baa	b12	45
bab	b3	5
bac	b3	5
cba	c45	56
cbb	c4	5
cbc	c4	5

baaa	b1	5
baab	b2	4
baac	b2	4
cbaa	c45	56
cbab	c5	6
cbac	c5	6

cbaaa	c5	6
cbaab	c4	5
cbaac	c4	5

Сведем коды всех состояний в следующую таблицу:

1	aaaxx
2	aabxx
–	aacxx
3	abxxx
–	acxxx
4	bbxxx
–	bcxxx
–	baabx
–	baacx

5	caxxx
–	babxx
–	bacxx
–	cbbxx
–	cbsxx
–	baaax
–	cbaab
–	cbaac

6	ccxxx
–	cbabx
–	cbacx
–	cbaaa

2.2. Автомат с бинарной функцией обратной связи

Если в УниПарГрафе автомата есть контуры, то такой автомат нельзя реализовать без обратных связей. Задача ставится следующим образом – реализовать автомат с единичным индексом обратной связи или иначе с бинарной функцией обратной связи, с использованием регистров сдвига, как на входе, так и для обратной связи рис.11.

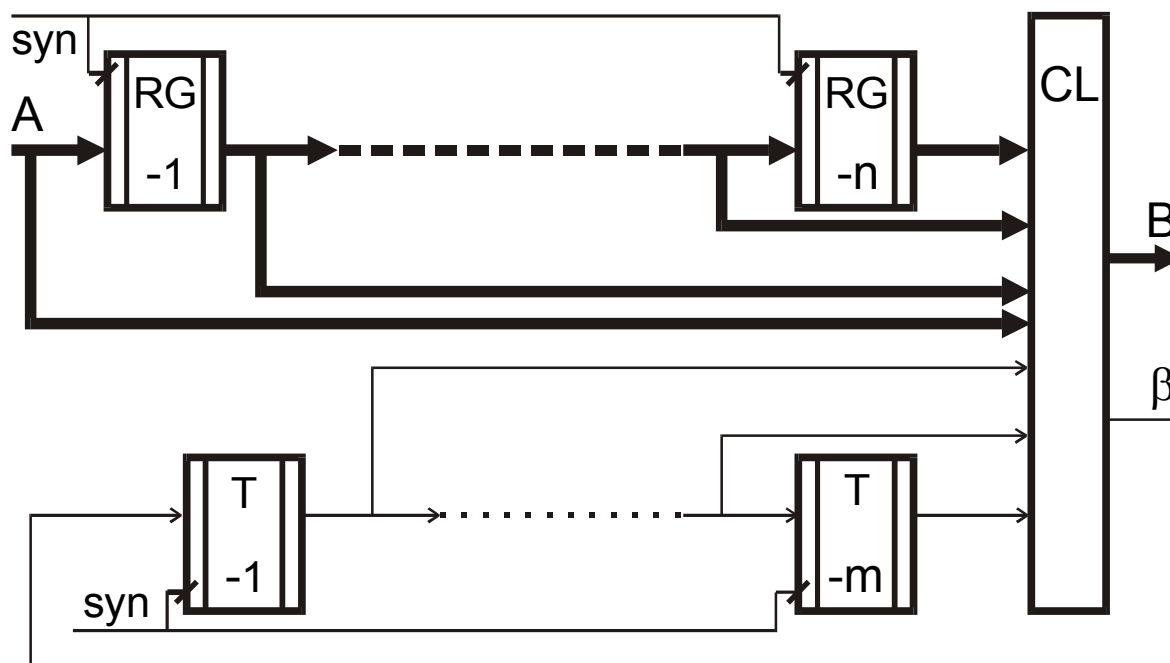


Рис.11. Автомат с единичным индексом обратной связи

Пример 2.2.-1. Автомат задается таблицей

	a	b
<u>1</u>	0, <u>2</u>	1, <u>4</u>
<u>2</u>	1, <u>1</u>	0, <u>3</u>
<u>3</u>	0, <u>3</u>	1, <u>2</u>
<u>4</u>	1, <u>4</u>	1, <u>3</u>

УниПарГраф этого автомата представлен на рис.12. слева. *Парный граф (ПарГраф)* аналог УниПарГрафа, но с учетом значений обратной связи, в этом примере совпадающих с выходными, представлен на том же рисунке справа. В ПарГрафе нет контуров, и самая длинная цепь состоит из 5 вершин, рис.12.

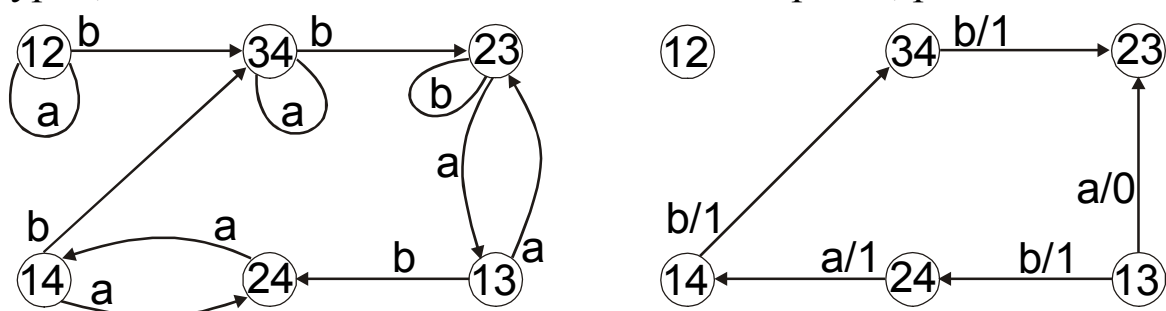


Рис.12. Универсальный парный и Парный графы

Это означает, что последовательность из 5 входных и двоичных символов обратной связи идентифицируют все состояния автомата со структурой рис.1. Что бы найти коды состояний построим дерево-преемника (таблицу) с учетом значений функции обратной связи.

	a/0	a/1	b/0	b/1
1234	23	14	3	234
23	3	1	3	2
14	2	4	—	34
234	3	<u>14</u>	3	<u>23</u>
34	3	4	—	<u>23</u>

Действуя аналогичным предыдущему примеру образом, получим следующие коды состояний см. таблицу.

УниПарГраф может быть использован для определения функции обратной связи $\Psi(s,d)$, как функции состояния и входа. Обозначим значение этой функции $x_{sd} = \Psi(s,d)$. Если для какой либо дуги, помеченной входным символом d и выходящей из вершины (h,k) , выбрать значения функции Ψ такими, чтобы $x_{hd} \neq x_{kd}$, то этой дуги в ПарГрафе уже не будет. Для определения функции

обратной связи надо для значений этой функции составить систему неравенств, решение которой позволяет избавиться от контуров в ПарГрафе.

Таблица кодов состояний:

1	aaxxx 10xxx	3	bxxxx 0xxxx	4	aaxxx 11xxx
-	abaxx 110xx	-	aaxxx 00xxx	-	abaxx 111xx
-	abbxx 110xx	-	abxxx 00xxx	-	baaxx 110xx
-	abbxx 111xx	-	abxxx 01xxx	-	babax 1110x
2	aaxxx 01xxx	-	baaxx 111xx	-	babbx 1110x
-	baxxx 10xxx	-	babax 1111x	-	babbx 1111x
-	bbxxx 10xxx	-	bbaax 1110x		
-	bbbxx 111xx	-	bbaba 11110		
-	bbaax 1111x	-	bbabb 11110		
-	bbaba 11111	-	bbabb 11111		

Запишем систему неравенств для примера 1:

$x_{1a} \neq x_{2a}$ –разрывает контур $\{12 \underline{a}\}$,

$x_{3a} \neq x_{4a}$ –разрывает контур $\{34 \underline{a}\}$,

$x_{2b} \neq x_{3b}$ –разрывает контур $\{23 \underline{b}\}$,

$(x_{1a} \neq x_{4a}) \vee (x_{2a} \neq x_{4a})$ –разрывает контур $\{14 \underline{a} 24 \underline{a}\}$,

$(x_{1a} \neq x_{3a}) \vee (x_{2a} \neq x_{3a})$ –разрывает контур $\{13 \underline{a} 23 \underline{a}\}$,

последние два условия должны выглядеть сложнее – не должно быть так же контура $\{13 \underline{b} 24 \underline{a} 14 \underline{b} 34 \underline{b} 23 \underline{a}\}$.

Для двоичных значений переменных x_{si} система неравенств может не иметь решения или иметь одно или более решений. В поиске решения (двоичного) будем записывать, если это возможно, неравенства так, чтобы одна и та же переменная находилась в одной и той же части (левой или правой) неравенства. Для указанной выше функции решение выглядит так

$$\begin{aligned}
 X_{1a} &\neq X_{2a} & X_{2b} &\neq X_{3b} \\
 X_{3a} &\neq X_{4a} \\
 X_{1a} &\neq X_{4a} \\
 X_{3a} &\neq X_{2a}
 \end{aligned}$$

Присваивая переменным левой части неравенства значение 0, а правой – 1, получаем решение. Поскольку у исходной системы неравенств много решений, то можно выбрать такое, чтобы длина цепей в ПарГрафе, а значит и разрядность регистров сдвига, была минимальной, например:

$$\begin{aligned}
 X_{1a} &\neq X_{2a} & X_{2b} &\neq X_{3b} \\
 X_{4a} &\neq X_{3a} & X_{1b} &\neq X_{3b}, \text{ удаляя дугу } (13 \rightarrow 24) \\
 X_{4a} &\neq X_{2a} & X_{4b} &\neq X_{3b}, \text{ удаляя дугу } (34 \rightarrow 23) \\
 X_{1a} &\neq X_{3a}
 \end{aligned}$$

Результатом решения является бинарная функция обратной связи. ПарГраф соответствующий такому решению имеет цепи максимальной длины – 2, рис.13.

функция обратной

связи	a	b
1	0	0
2	1	0
3	1	1
4	0	0

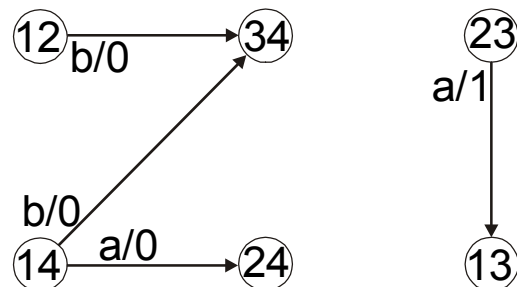


Рис.13. Парный граф

Рассмотрим пример, в котором для двоичных значений переменных система неравенств не имеет решения.

Пример 2.2.-2: Автомат задан таблицей (без выхода). Уни-ПарГраф – на рис.14.

Автоматная

таблица	a	b
1	2	3
2	1	4
3	4	1
4	3	2

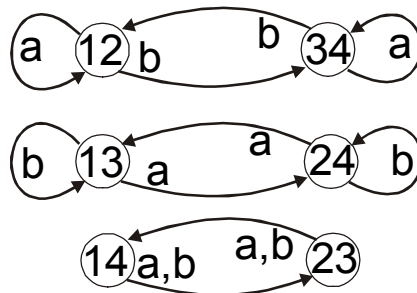


Рис.14. Универсальный парный граф

Система неравенств для значений функции ψ :

$$x_{1a} \neq x_{2a},$$

$$x_{1b} \neq x_{3b},$$

$$x_{3a} \neq x_{4a},$$

$$x_{2b} \neq x_{4b},$$

$$(x_{1a} \neq x_{3a}) \vee (x_{2a} \neq x_{4a}),$$

$$(x_{1b} \neq x_{2b}) \vee (x_{3b} \neq x_{4b}),$$

$$(x_{1a} \neq x_{4a}) \& (x_{1b} \neq x_{4b}) \vee (x_{2a} \neq x_{3a}) \& (x_{2b} \neq x_{3b}).$$

Для двоичных значений у этой системы неравенств нет решения.

Рассмотрим метод, используя который можно для любого автомата получить бинарную функцию обратной связи. Назовем s -словом последовательность из пар (ВходнойСимвол, ЗначениеФункцииОбратнойСвязи).

Сформулируем, непосредственно следующее из правил построения ПарГрафа и используемое в методе

утверждение: Для того, что бы в ПарГрафе не было контуров, необходимо и достаточно, что бы не существовало s -слова, которое переводит автомат из состояния i в i и оно же из состояния j в j (для какой-либо пары состояний).

Если система неравенств для значений функции ψ не имеет решения, то метод получения бинарной функции обратной связи ψ состоит в следующем:

1. Перечислить все элементарные контуры в автоматном графе.
2. Добавить эквивалентные состояния и значения ψ функции так, что бы слова в контурах удовлетворяли выше указанному УТВЕРЖДЕНИЮ.

Для рассматриваемого примера:

Автоматная
таблица

	a	b
1	2	3
2	1	4
3	4	1
4	3	2

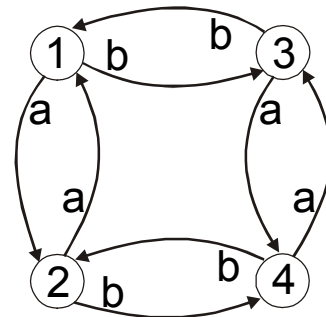


Рис.15. Диаграмма переходов

Элементарные
контуры

Исправленные контуры

Автоматная
таблица с
функцией ψ

$\{1 \underline{a} 2 \underline{a}\}$	$\{1 \underline{a/0} 2 \underline{a/1}\}$
$\{3 \underline{a} 4 \underline{a}\}$	$\{3 \underline{a/1} 4 \underline{a/0} 3^+ \underline{a/0} 4^+ \underline{a/0}\}$
$\{1 \underline{b} 3 \underline{b}\}$	$\{1 \underline{b/0} 3 \underline{b/1}\}$
$\{2 \underline{b} 4 \underline{b}\}$	$\{2 \underline{b/1} 4 \underline{b/0} 2^+ \underline{b/0} 4^+ \underline{b/0}\}$
$\{1 \underline{a} 2 \underline{b} 4 \underline{a} 3 \underline{b}\}$	$\{1 \underline{a/0} 2 \underline{b/1} 4 \underline{a/0} 3^+ \underline{b/0}\}$
$\{1 \underline{b} 3 \underline{a} 4 \underline{b} 2 \underline{a}\}$	$\{1 \underline{b/0} 3 \underline{a/1} 4 \underline{b/0} 2^+ \underline{a/0}\}$

	a	b
1	0,2	0,3
2	1,1	1,4
2 ⁺	0,1	0,4 ⁺
3	1,4	1,1
3 ⁺	0,4 ⁺	0,1
4	0,3 ⁺	0,2 ⁺
4 ⁺	0,3	0,2

2.3. Независимость от состояний

Об автоматах, которые могут быть реализованы либо без обратных связей, либо выходная функция может быть функцией обратной связи, можно говорить как об автоматах, *независящих от состояний*, т.е. зависимость выхода от входа может быть выражена без привлечения понятия состояние.

$$b_k = f(a_k, a_{k-1}, \dots, a_{k-n})$$

$$b_k = f(a_k, a_{k-1}, \dots, a_{k-n}, b_{k-1}, \dots, b_{k-n})$$

3. Автоматы без потери информации

Автомат можно рассматривать как устройство, которое осуществляет кодирование входных последовательностей в выходные. Если после такого кодирования можно путем декодирования восстановить исходную информацию, то такие автоматы называют автоматами *без потери информации*, *ИЛ-автоматами* (Information Lossless).

Рассмотрим только такие ИЛ-автоматы, для которых можно построить *инверсный* автомат, т.е. автомат, восстанавливающий исходную информацию. Это

- 1) *ИЛ(B)-автомат*, для которого входную последовательность можно определить по его начальному состоянию и выходной последовательности.
- 2) *ИЛ(E)-автомат*, для которого входную последовательность можно определить по его конечному состоянию и выходной последовательности.

Если автомат – *ИЛ(B)-автомат*, то не разумно использовать его как

$IL(E)$ -автомат, так как кодирование то же, а декодирование сложнее.

3.1. Автомат является $IL(V)$ -автоматом тогда и только тогда, когда для каждого состояния выходные значения различны для различных входных значений.

Необходимость этого условия следует из следующего рассуждения. Если в некотором состоянии при различных входных значениях выходное значение одно и то же, то в этом случае входное значение нельзя восстановить однозначно.

Достаточность условия следует из возможности однозначного определения входного значения по отличающимся друг от друга выходным значениям в каждом состоянии автомата. Построение инверсного автомата очевидно.

3.2. Для восстановления входной последовательности $IL(E)$ -автомата по выходной, надо эту выходную последовательность запомнить и затем, двигаться от финального состояния к началу, вычислять входные значения. Что бы проверить возможность такого вычисления, надо построить тестовую таблицу всех «обратных движений», допускающих однозначное восстановление входного значения.

Столбцы тестовой таблицы соответствуют всем возможным выходам – \mathbf{b}^i .

В первой части тестовой таблицы строки соответствуют отдельным состояниям автомата – \mathbf{s}^j . Содержимым клетки с координатами $(\mathbf{b}^i, \mathbf{s}^j)$ является множество состояний – \mathbf{S}^{ij} , предшествующих такому событию, т.е. все те состояния, у которых в строках автоматной таблицы содержится элемент $(\mathbf{b}^i, \mathbf{s}^j)$. При этом, если такой элемент появляется в разных столбцах (при разных входных значениях) автоматной таблицы, то однозначное восстановление входного значения невозможно. Делается вывод, что это не – $IL(E)$ -автомат. (Это *необходимое* условие можно проверить до построения тестовой таблицы. *Достаточным* условием является единственность каждого из элементов $(\mathbf{b}^i, \mathbf{s}^j)$ в автоматной таблице, что так же можно проверить до построения тестовой таблицы.)

Строками второй части тестовой таблицы являются вновь

появившиеся в первой части таблицы множества состояний. Клетка с координатами (b^q, S^{ij}) заполняется аналогично множеству состояний, предшествующих событиям из множества $\{(b^q, s^k), \forall s^k \in S^{ij}\}$. При этом, если элементы этого множества встречаются в разных столбцах автоматной таблицы, то делается вывод, что это не $IL(E)$ -автомат.

Последующие части тестовой таблицы строятся аналогично второй. Процесс построения таблицы естественным образом заканчивается.

Можно построить инверсный автомат, который вычислит входную последовательность по записанной выходной последовательности и финальному состоянию $IL(E)$ -автомата. В $IL(E)$ -автомате ориентацию всех дуг изменить на противоположную, в метке каждой дуги поменять местами входы и выходы. При этом может получиться недетерминированный автомат – источник. Начальное состояние источника соответствует финальному $IL(E)$ -автомата.

4. Условная синхронизация

Наряду с канонической структурой СЦА (рис.1. гл.I) в некоторых случаях для более экономной реализации комбинационной части автомата возможно использование структуры, представленной на рис.16.

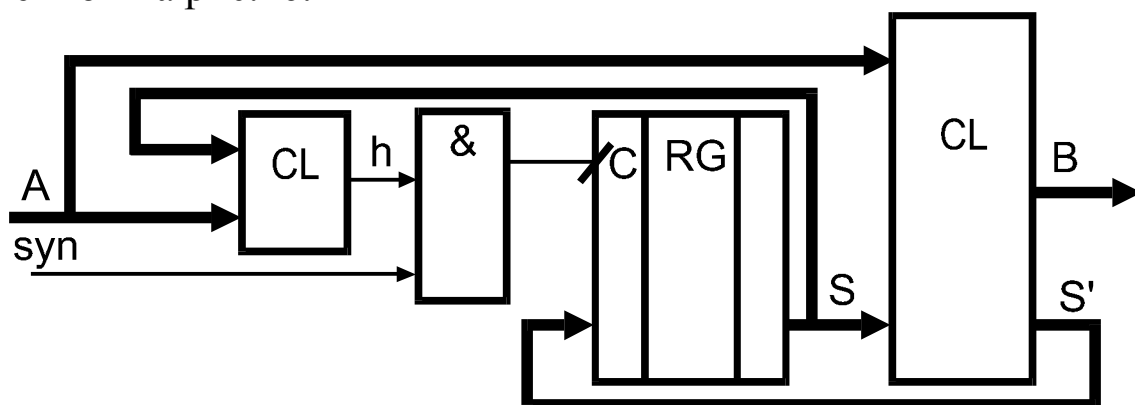


Рис.16. СЦА с условной синхронизацией

Для реализации переходов по петлям, т.е. в тех случаях, когда нужно оставить автомат в предыдущих состояниях, достаточно с помощью дополнительной КС запретить синхронизацию па-

мента автомата. При этом значение на информационных входах RG безразлично. Временные параметры сигнала синхронизации должны быть определены так, чтобы на выходе h не было переключений при $\text{syn}=1$.

Пример см. гл. I, п. 7.2.

ГЛАВА III. ЛИНЕЙНЫЕ АВТОМАТЫ

Линейные автоматы (ЛА) используются в аппаратуре и программном обеспечении систем кодирования и декодирования данных, цифровой фильтрации, устройств обнаружения и исправления ошибок, контроля цифровых схем (сигнатурный анализ).

1. Основные определения

Синхронный цифровой автомат линейен, если комбинационная логика для вычисления функций перехода и выхода линейна. По определению функция линейна, если она удовлетворяет принципу суперпозиции. Для булевых функций это означает:

$$f(0, \dots, 0) = 0$$

$$f(x_1, x_2, \dots, x_n) = f(x_1, 0, \dots, 0) + f(0, x_2, \dots, 0) + \dots + f(0, 0, \dots, x_n)$$

Комбинационная логика линейна, если используется единственная булевская операция «сложения по модулю 2». Элемент, ее реализующий (M2), будем называть сумматором.

Каждому из одноразрядных выходов КС линейного автомата можно сопоставить один многовходовой сумматор по модулю 2 (Σ_2). Тогда значения состояний и выходов могут быть заданы уравнениями:

$$s_i(t+1) := \sum_{j=1}^n {}_2 g_{ij} s_j(t) + \sum_{j=1}^k {}_2 c_{ij} a_j(t) \quad (1.1')$$

$$b_i(t) = \sum_{j=1}^n {}_2 d_{ij} s_j(t) + \sum_{j=1}^k {}_2 e_{ij} a_j(t), \quad (1.1'')$$

где коэффициенты (g_{ij} , c_{ij} , d_{ij} , e_{ij}) равны $\{0, 1\}$ и означают отсутствие или присутствие булевской переменной на входе сумматора.

Эти же уравнения можно рассматривать не как булевские уравнения, а как линейные уравнения над полем GF(2). Коэффи-

циенты и переменные имеют только двоичные значения $\{0,1\}$. Сложение и вычитание выполняются по модулю 2 ($\text{mod}2$). Поскольку $(-1=1)\text{mod}2$, то операция вычитания по $\text{mod}2$ может быть заменена операцией сложения по $\text{mod}2$, которую и обозначаем знаком $+$.

Уравнения (1) можно переписать в матричной форме:

$$\mathbf{s}(t+1) : = \mathbf{G} \mathbf{s}(t) + \mathbf{C} \mathbf{a}(t) \quad (1.2')$$

$$\mathbf{b}(t) = \mathbf{D} \mathbf{s}(t) + \mathbf{E} \mathbf{a}(t), \quad (1.2'')$$

где $\mathbf{a} = \|a_i\|_{k \times 1}$, $\mathbf{s} = \|s_i\|_{n \times 1}$, $\mathbf{b} = \|b_i\|_{m \times 1}$

векторы входной, состояний, выходной; матрицы

$$\mathbf{G} = \|g_{ij}\|_{n \times n}, \mathbf{C} = \|c_{ij}\|_{n \times k}$$

$$\mathbf{D} = \|d_{ij}\|_{m \times n}, \mathbf{E} = \|e_{ij}\|_{m \times k}$$

называются *характеристическими* (\mathbf{G} —называется *основной характеристической*); n —количество двоичных элементов памяти, k —количество двоичных входов, m —количество двоичных выходов.

Для ЛА с одним входом и одним выходом (*двухполюсного* ЛА) матрицы \mathbf{C} и \mathbf{D} становятся векторами, матрица \mathbf{E} вырождается в единственный коэффициент равный 0 или 1.

2. Линейные автоматы без потери информации

Если ранг матрицы \mathbf{E} равен k (количеству входов), то ЛА является автоматом *без потери информации*, см. п. II-3, по начальному состоянию и выходной последовательности можно однозначно восстановить входную последовательность ($IL(B)$ —автомат). Это утверждение следует из уравнений (2) индукцией по t .

3. Формула полной реакции ЛА

Методом индукции по t можно показать, что для любого $t \geq 0$

$$\mathbf{s}(t+1) := \mathbf{G}^{t+1} \mathbf{s}(0) + \sum_{v=0}^t \mathbf{G}^{t-v} \mathbf{C} \mathbf{a}(v) \quad (3.1')$$

$$\mathbf{b}(t) = \mathbf{D} \mathbf{G}^t \mathbf{s}(0) + \sum_{v=0}^t \mathbf{W}(t-v) \mathbf{a}(v), \quad (3.1'')$$

$$\text{где } \mathbf{W}(t-v) = \begin{cases} \mathbf{E} & (t-v=0) \\ \mathbf{D}\mathbf{G}^{t-v-1}\mathbf{C} & (t-v>0) \end{cases}$$

Формулу (3.1") называют *формулой полной реакции ЛА*.

Из (3.1") следует, что выходной сигнал состоит из двух составляющих: *свободного движения*

$$\mathbf{b}(t)|_{\text{своб}} = \mathbf{D}\mathbf{G}^t \mathbf{s}(0),$$

получаемого при $\mathbf{a}(t) = 0$ для всех $t \geq 0$,

и *вынужденного движения*

$$\mathbf{b}(t)|_{\text{вын}} = \sum_{v=0}^t \mathbf{W}(t-v) \mathbf{a}(v),$$

получаемого при $\mathbf{s}(0) = 0$. Для любой заданной входной последовательности $\mathbf{a}(t)$ ($t = 0, 1, 2, \dots$) и заданного начального состояния $\mathbf{s}(0)$ эти составляющие могут быть вычислены отдельно, а затем сложены.

Для ЛА с одним входом и одним выходом – *двухполюсного* ЛА $\mathbf{W}(t)$ становится скалярной функцией $w(t)$, а уравнение (3") принимает форму

$$b(t) = \sum_{v=0}^t w(t) a(v),$$

4. Изоморфные и эквивалентные линейные автоматы

Определения изоморфизма и эквивалентности (см. I-6.1), разумеется, распространяются и на ЛА.

Если ЛА A характеризуется матрицами \mathbf{G} , \mathbf{C} , \mathbf{D} , \mathbf{E} , а ЛА \tilde{A} характеризуется матрицами

$$\tilde{\mathbf{G}} = \mathbf{P}\mathbf{G}\mathbf{P}^{-1}, \tilde{\mathbf{C}} = \mathbf{P}\mathbf{C}, \tilde{\mathbf{D}} = \mathbf{D}\mathbf{P}^{-1}, \tilde{\mathbf{E}} = \mathbf{E}, \quad (4.1)$$

где \mathbf{P} – некоторая неособенная матрица ($\det \mathbf{P} \neq 0$) над полем $\text{GF}(2)$, то A и \tilde{A} изоморфны.

Состоянию \mathbf{s} автомата A изоморфно соответствует состояние $\tilde{\mathbf{s}} = \mathbf{P}\mathbf{s}$ автомата \tilde{A} . Изоморфизм автоматов проверяется подстановкой в (2).

Матрицы \mathbf{G} и $\tilde{\mathbf{G}}$ называются *подобными*. Две матрицы подобны тогда и только тогда, когда у них равны характеристические полиномы ($\det(\mathbf{M} - x\mathbf{I})$), с точностью до мультипликативной

константы.

Преобразование подобия изменяет матрицы линейных операторов, и соответственно – структуру (схему) линейного автомата. С помощью такого преобразования можно привести матрицу к виду удобному для реализации.

Внутренней сетью линейного автомата называется часть реализации, которая определяется только основной характеристической матрицей \mathbf{G} . Внутренняя сеть соединяет между собой элементы памяти автомата. Среди возможных структур внутренней сети рассмотрим структуры удобные для реализации.

Сопровождающей матрицей $\mathbf{M}_{P(x)}$ нормированного полинома над полем $GF(2)$

$$P(x) = p_0 + p_1x + p_2x^2 + \dots + p_{n-1}x^{n-1} + x^n$$

называется (n, n) – матрица:

$$\mathbf{M}_{P(x)} = \begin{vmatrix} 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & & 0 \\ & & & \ddots & \\ & & & & \ddots & \\ 0 & 0 & 0 & & & 0 \\ 0 & 0 & 0 & & & 1 \\ p_0 & p_1 & p_2 & & & p_{n-1} \end{vmatrix} \quad (4.2')$$

Полином $P(x)$ является характеристическим полиномом ($P(x) = \det(\mathbf{M} - x\mathbf{I})$) и *минимальным аннулирующим* полиномом этой матрицы, т.е. полиномом наименьшей степени, для которого $P(\mathbf{M}) = 0$. Характеристический полином – всегда аннулирующий.

Матрица $\mathbf{M}_{P(x)}$ подобна своей транспонированной матрице

$$\mathbf{M}_{P(x)}^{TP} = \begin{vmatrix} 0 & 0 & & & 0 & p_0 \\ 1 & 0 & & & 0 & p_1 \\ 0 & 1 & & & 0 & p_2 \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ 0 & 0 & & & 1 & p_{n-1} \end{vmatrix} \quad (4.2'')$$

Внутренняя сеть, реализующая сопровождающую матрицу, называется *односумматорным регистром сдвига*.

Внутренняя сеть, реализующая транспонированную сопровождающую матрицу, называется *многосумматорным регистром сдвига*.

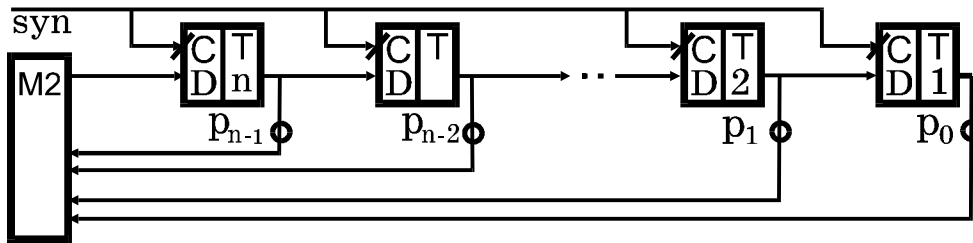


Рис.1. Односумматорный регистр сдвига

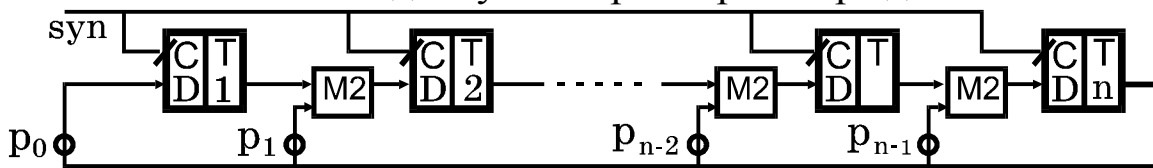


Рис.2. Многосумматорный регистр сдвига

Полином $P(x)$ называется *полиномом обратной связи*, поскольку однозначно определяет скалярные константы в цепях обратной связи регистров сдвига.

Если характеристический полином может быть представлен как произведение нормированных полиномов $P_1(x)$, $P_2(x)$, ..., $P_k(x)$, то матрица \mathbf{M} с характеристическим полиномом $P(x)$ подобна матрице блочно - диагонального вида:

$$\mathbf{M}^* = \begin{pmatrix} \mathbf{M}_{P_1(x)} & & & \\ & \mathbf{M}_{P_2(x)} & & \\ & & \ddots & \\ & & & \mathbf{M}_{P_k(x)} \end{pmatrix} \quad (4.3)$$

Подобие \mathbf{M}^* и \mathbf{M} не нарушается, если меняется порядок блоков или любой блок заменяется транспонированным.

5. Минимальные линейные автоматы

Общее определение минимальности автоматов (п.І-6.2) справедливо и для ЛА.

Линейный автомат, не имеющий эквивалентных состояний – *минимальный*.

Из формулы полной реакции (3.1") следует, что s_1 и s_2 эквивалентны тогда и только тогда, когда для любого $t \geq 0$ и любого $a(t)$

$$\mathbf{DG}^t \mathbf{s}_1 + \sum_{v=0}^t \mathbf{W}(t-v) \mathbf{a}(v) = \mathbf{DG}^t \mathbf{s}_2 + \sum_{v=0}^t \mathbf{W}(t-v) \mathbf{a}(v)$$

или в том и только в том случае, если для любого $t \geq 0$

$$\mathbf{DG}^t \mathbf{s}_1 = \mathbf{DG}^t \mathbf{s}_2 \quad (5.1')$$

или иначе

$$\mathbf{DG}^t (\mathbf{s}_1 - \mathbf{s}_2) = \mathbf{0} \quad (5.1'')$$

Чтобы установить эквивалентность состояний n -мерного автомата с $N=2^n$ состояниями, достаточно последовательности длины $N-1$ (п. I-6.1); для ЛА достаточно последовательности длины n .

Поскольку для матрицы \mathbf{G} минимальный аннулирующий полином степени не выше n , то \mathbf{G}^t для любого $t \geq n$ выражается полиномом относительно \mathbf{G} степени не выше $n-1$ (см. п.5). Соответственно $\mathbf{DG}^t \mathbf{u}$ при любом $t \geq n$ выражается в виде линейной комбинации матриц \mathbf{Du} , $\mathbf{DG}\mathbf{u}$, $\mathbf{DG}^2\mathbf{u}$, ..., $\mathbf{DG}^{n-1}\mathbf{u}$. Поэтому, если равны нулю все эти матрицы, то $\mathbf{DG}^t \mathbf{u} = \mathbf{0}$ для всех $t \geq 0$.

Составим *диагностическую матрицу* (n, n)

$$\mathbf{H} = \begin{pmatrix} \mathbf{D} \\ \mathbf{DG} \\ \mathbf{DG}^2 \\ \vdots \\ \mathbf{DG}^{n-1} \end{pmatrix}$$

В силу равенства (7) $\mathbf{H}\mathbf{s}_1 = \mathbf{H}\mathbf{s}_2$ или $\mathbf{H}(\mathbf{s}_1 - \mathbf{s}_2) = \mathbf{0}$. Состояния \mathbf{s}_1 и \mathbf{s}_2 эквивалентны, если вектор $\mathbf{u} = \mathbf{s}_1 - \mathbf{s}_2$ принадлежит ядру (нуль-пространству) матрицы \mathbf{H} . Размерность ядра называют дефектом (defekt) матрицы. Если ранг диагностической матрицы \mathbf{H} равен r ($\text{rang} \mathbf{H} = r$), то $\text{defekt} \mathbf{H} = n - r$; число классов эквивалентности равно 2^r (для общего случая поля $GF(p)$ это число равно p^r).

Для автомата A с матрицами \mathbf{G} , \mathbf{C} , \mathbf{D} , \mathbf{E} построим минимальный автомат \tilde{A} следующим образом: состояниям \mathbf{s} автомата A поставим в соответствие состояния $\check{\mathbf{s}} = \mathbf{T}\mathbf{s}$ автомата \tilde{A} , где (r, n) -матрица \mathbf{T} составлена из r ($r < n$, если $r = n$, то автомат уже минимальный) линейно независимых строк диагностической матрицы \mathbf{H} . Характеристические матрицы автомата \tilde{A} определим следующим образом:

$$\check{\mathbf{G}} = \mathbf{TGR}, \quad \check{\mathbf{C}} = \mathbf{TC}, \quad \check{\mathbf{D}} = \mathbf{DR}, \quad \check{\mathbf{E}} = \mathbf{E},$$

где \mathbf{R} (n, r) -матрица правая обратная к \mathbf{T} ($\mathbf{TR} = \mathbf{I}_r$).

Для автомата A , находящегося в состоянии s_j , новое состояние определяется из уравнения (2')

$$s_f := Gs_j + Ca$$

Определим, в какое состояние перейдет автомат \tilde{A} , находящийся в состоянии $\check{s}_j = Ts_j$

$$\check{G}\check{s}_j + \check{C}a = TGR(Ts_j) + TCa$$

Рассмотрим состояние $\hat{s} = (RT)s$, для которого $T\hat{s} = TRTs = Ts$, это означает, что состояния \hat{s} и s эквивалентны. Поэтому

$$TG(RTs_j) + TCa = T(Gs_j + Ca) = T(s_f) = \check{s}_f,$$

т.е. автомат \tilde{A} оказывается в состоянии эквивалентном состоянию s_f автомата A .

Выходное значение автомата \tilde{A} определяем из (1.2")

$$\check{D}\check{s} + \check{E}a = DRTs + Ea = Ds + Ea = b$$

Это значение такое же, как и в автомате A . Это означает, что автоматы эквивалентны.

6. Минимально-канонические и простые канонические формы

Минимальный ЛА преобразованием подобия может быть приведен к эквивалентному минимальному автомату с основной характеристической матрицей, имеющей блочно-диагональный вид. В этом случае внутренняя сеть состоит из взаимно-независимых регистров сдвига. Такой ЛА называется *минимально-каноническим*.

Минимальный ЛА с одним выходом можно привести к автомату с основной характеристической матрицей в форме сопровождающей. Такой автомат имеет простейшую выходную схему, выход связан только с одним элементом памяти регистра сдвига. Для произвольного ЛА с более чем одним выходом может быть построен эквивалентный автомат, как минимальный по отношению к каждому выходу отдельно. Структура такого ЛА называется *простой канонической*. В общем случае он не является минимальным автоматом, но имеет простые выходные соединения.

7. Независимость от состояний

Поскольку ЛА всегда может быть приведен к простому каноническому виду, это означает, что выходное значение можно определить как функцию только входных и выходных значений за предыдущие r тактов, т.е. независимо от состояний, см. так же п.П-2.3. Разумеется, этот результат можно получить и аналитически. Исходя из формулы полной реакции (3.1"), обозначим

$$\mathbf{z}(t) = \mathbf{D}\mathbf{G}^t\mathbf{s}(0) = \mathbf{b}(t) - \sum_{v=0}^t \mathbf{W}(t-v) \mathbf{a}(v)$$

Если минимальный аннулирующий полином матрицы \mathbf{G} имеет

$$\text{вид: } \Omega_{\mathbf{G}} = \omega_0 + \omega_1 x + \omega_2 x^2 + \dots + \omega_{r-1} x^{r-1} + x^r, \text{ то}$$

$$\mathbf{D}\mathbf{G}^r\mathbf{s}(0) = -\omega_0\mathbf{D}\mathbf{s}(0) - \omega_1\mathbf{D}\mathbf{G}\mathbf{s}(0) - \omega_2\mathbf{D}\mathbf{G}^2\mathbf{s}(0) - \dots - \omega_{r-1}\mathbf{D}\mathbf{G}^{r-1}\mathbf{s}(0)$$

$$\text{или } \mathbf{z}(r) = -\omega_0\mathbf{z}(0) - \omega_1\mathbf{z}(1) - \omega_2\mathbf{z}(2) - \dots - \omega_{r-1}\mathbf{z}(r-1),$$

$$\text{тогда } \mathbf{b}(r) - \sum_{v=0}^r \mathbf{W}(r-v)\mathbf{a}(v) = -\sum_{t=0}^{r-1} \omega_t (\mathbf{b}(t) - \sum_{v=0}^t \mathbf{W}(t-v)\mathbf{a}(v))$$

$$\mathbf{b}(r) = -\sum_{t=0}^{r-1} \omega_t \mathbf{b}(t) + \sum_{v=0}^r \mathbf{W}(r-v)\mathbf{a}(v) + \sum_{t=0}^{r-1} \omega_t \sum_{v=0}^t \mathbf{W}(t-v)\mathbf{a}(v),$$

$$\mathbf{b}(r) = f(\mathbf{a}(0), \dots, \mathbf{a}(r), \mathbf{b}(0), \dots, \mathbf{b}(r-1))$$

Частный случай – ЛА без обратных связей. Выходное значение в этом случае зависит только от входных значений в том же такте и за предыдущие k тактов ($k \leq r$). ЛА обладает этим свойством, если k является наименьшим числом, для которого $\mathbf{D}\mathbf{G}^k = \mathbf{0}$.

8. Автономные линейные автоматы (АЛА)

ЛА, входы и состояния которых не зависят от входных воздействий, называются *автономными линейными автоматами* (АЛА). Схема внутренней сети любого ЛА является схемой АЛА и (см. п.6) всегда может быть приведена к каноническому виду. Поэтому, рассматривая схемы АЛА, достаточно ограничиться только схемами регистров сдвига – рис.1 и рис.2.

Для этих схем можно записать *выходную рекуррентную последовательность*:

$$b(t+n) = p_{n-1}b(t+n-1) + \dots + p_1b(t+1) + p_0b(t).$$

Выходная последовательность АЛА однозначно определяется начальным состоянием $s(0)$ и является свободной составляющей полной реакции ЛА (см. п.3).

8.1. Анализ АЛА.

Любой неособенный, т.е. не делящийся на x , полином $P(x)$ над полем $GF(k)$ является делителем полинома $1-x^i$ для некоторого целого числа i . Наименьшее такое положительное число i называется *показателем, которому принадлежит полином $P(x)$* . Если $\deg P = n$ (степень полинома) и полином принадлежит показателю T , то число T является делителем числа $k^n - 1$.

Пусть $\Psi(x)$ *неприводимый* полином над полем $GF(k)$, т.е. полином, единственными делителями которого являются константа α и $\Psi(x)$. Если полином $\Psi(x)$ принадлежит показателю T , то полином $(\Psi(x))^j$ принадлежит показателю $k^r T$, где $k^{r-1} < j \leq k^r$.

Пусть $P(x) = P_1(x)P_2(x)$, а $P_1(x)$ и $P_2(x)$ взаимно просты и принадлежат соответственно показателям T_1 и T_2 , тогда $P(x)$ принадлежит показателю $T = \text{н.о.к.}(T_1, T_2)$.

Для строго периодической последовательности

$$b_t : b_1 b_2 \dots b_T, b_1 b_2 \dots b_T, \dots, \quad (8.1')$$

с периодом T назовем *представляющим* полином

$$B(x) = b_1 x^{T-1} + b_2 x^{T-2} + \dots + b_{T-1} x + b_T \quad (8.1'')$$

Пусть АЛА A имеет неособенный полином обратной связи $P(x)$, принадлежащий показателю T . Тогда полином

$$\vartheta(x) = (1 - x^T) / P(x)$$

называется *воспроизводящим полиномом* АЛА A . Если $\deg P = n$, то $\deg \vartheta = T - n$. Любой полином, принадлежащий векторному пространству с базисом

$$\{ x^{n-1} \vartheta(x), x^n \vartheta(x), \dots, x^2 \vartheta(x), x \vartheta(x), \vartheta(x) \},$$

представляет одну из строго периодических последовательностей (с периодом T), являющихся выходными последовательностями АЛА.

Пусть АЛА A имеет неособенный полином обратной связи $P(x)$, принадлежащий показателю T . Если полином $P_1(x)$, принадлежащий показателю T_1 , является делителем полинома $P(x)$, то

$$\vartheta_1(x) = (1 - x^T) / P_1(x)$$

является полиномом, воспроизводящим в базисе

$$\{ x^{n-1} \vartheta_1(x), x^n \vartheta_1(x), \dots, x^2 \vartheta_1(x), x \vartheta_1(x), \vartheta_1(x) \}$$

строго периодические последовательности с периодом T_1 .

8.2. Синтез АЛА

Для любого n и любого простого k существует, по крайней мере, один неприводимый полином n -й степени над полем $GF(k)$, принадлежащий показателю $(k^n - 1)$. Этот полином называется *полиномом, принадлежащим максимальному показателю* или *короче примитивным полиномом*. Если полином обратной связи примитивный, то АЛА называются *АЛА максимального периода*. Для $k=2$ период выходной последовательности в автомате равен $2^n - 1$. Все ненулевые выходные последовательности имеют период $2^n - 1$. Любые n последовательных символов (кроме n нулей) последовательности максимального периода однозначно задают все последующие символы, а первые $2^n - 1$ подпоследовательностей длины n последовательности максимального периода составляют множество всевозможных различных ненулевых последовательностей длины n . За время периода любая ненулевая подпоследовательность длины $m < n$ встречается 2^{n-m} раз, нулевая такой же длины $2^{n-m} - 1$. Одно из применений последовательностей максимального периода – порождение псевдослучайных чисел.

Пусть необходимо построить АЛА, воспроизводящий заданное множество строго периодических последовательностей $b^1(t), b^2(t), \dots, b^r(t)$, причем $b^i(t)$ имеет минимальный период T_i . Находим $T = \text{н.о.к.}(T_1, T_2, \dots, T_r)$. Последовательности $b^i(t)$ дополняем до строго периодических последовательностей периода T , а затем представим, аналогично (1), в виде полиномов $B_i(x)$. Найдем

$$\vartheta(x) = \text{н.о.д.}(1 - x^T, B_1(x), B_2(x), \dots, B_r(x)),$$

тогда $P(x) = (1 - x^T) / \vartheta(x)$ будет полиномом обратной связи АЛА наименьшей размерности, воспроизводящего заданные периодические последовательности.

Если в «худшем» случае $\vartheta(x) = 1$, или если не стремиться к

наименьшей размерности автомата, то всегда можно принять $P(x)=1-x^T$, что соответствует АЛА в виде *кольцевого регистра сдвига* с простейшими связями и без сумматоров. Кольцевой регистр сдвига реализует все периоды равные делителям числа T .

9. Линейные автоматы с нулевым начальным состоянием

ЛА с нулевым начальным состоянием $s(0)=0$ обозначим — ${}_0\text{ЛА}$.

На выходе ${}_0\text{ЛА}$ образуется только вынужденное движение. Во-первых, это часто применяемый режим работы линейных автоматов, используемых как преобразователи входных последовательностей в выходные. Во-вторых, вынужденное движение является одной из составляющих полной реакции ЛА в дополнение к свободному движению (см. п.3).

При $s(0)=0$ из формул полной реакции

$$\mathbf{b}(t) = \sum_{v=0}^t \mathbf{W}(t-v) \mathbf{a}(v), \quad (9.1)$$

следует, что значение сигнала на каждом выходе

$$b_i(t) = \sum_{j=1}^k \sum_{v=0}^t w_{ij}(t-v) a_j(v) = \sum_{j=1}^k b_{ij}(t)$$

равно сумме составляющих, каждая из которых зависит от одного входа.

Поэтому в дальнейшем при изучении ${}_0\text{ЛА}$ достаточно рассматривать двухполюсные автоматы.

Для двухполюсного ${}_0\text{ЛА}$ $\mathbf{W}(t)$ становится скалярной функцией $w(t)$, а уравнение (1) принимает форму

$$b(t) = \sum_{v=0}^t w(t) a(v),$$

9.1. D–преобразование

Для решения задач анализа и синтеза ${}_0\text{ЛА}$ используется аппарат дискретного преобразования Лапласа (D–преобразования).

9.1.1. Дискретное преобразование Лапласа двоичной последовательности $f(t) = f(0), f(1), f(2), f(3), \dots$ (при $t < 0$ $f(t)=0$) обозначается $D[f(t)]$ или F и определяется как формальный степенной ряд

над полем $GF(2)$:

$$D[f(t)] = F = \sum_{t=0}^{\infty} f(t)d^t = f(0) + f(1)d + f(2)d^2 + f(3)d^3 + \dots \quad (9.1)$$

Последовательность $f(t)$ и ее дискретное преобразование F называются парой соответствия. Последовательность $f(t)$ называется *оригиналом*, а функция F – *изображением*.

9.1.2. Сдвиг последовательности $f(t-k)$ действует на изображение как умножение на d^k . (По этому d называют *оператором запаздывания*.)

$$\sum_{t=k}^{\infty} f(t-k)d^t = \sum_{t'=0}^{\infty} f(t')d^{t'+k} = d^k \sum_{t'=0}^{\infty} f(t')d^{t'} = d^k F$$

9.1.3. Пусть $f(t)$ строго периодическая последовательность с периодом T , т.е. $f(t) = f(t+T)$, тогда согласно предыдущему пункту:

$$\begin{aligned} F = \sum_{t=0}^{\infty} f(t)d^t &= (f(0) + f(1)d + \dots + f(T-1)d^{T-1}) \times \\ &\times (1 + d^T + d^{2T} + \dots) = \\ &= \frac{f(0) + f(1)d + \dots + f(T-1)d^{T-1}}{1 - d^T} \end{aligned}$$

9.1.4. Функцию F называют *обратимой функцией*, если в результате обратного преобразования $D^{-1}[F]$ может быть получен оригинал:

$$D^{-1}[F] = f_t = f(t)$$

Любые *обратимые* изображения, встречающиеся при изучении ЛА, могут быть представлены в виде: $F = h(d)/q(d)$,

где $h(d)$ и $q(d)$ полиномы от d , такие, что минимальная степень d в $q(d)$ не превышает минимальной степени d в $h(d)$. Тогда функция F может быть выражена в виде степенного ряда от d , получаемого формальным делением $h(d)$ на $q(d)$.

$$F = h(d)/q(d) = f_0 + f_1d + f_2d^2 + f_3d^3 + \dots$$

$$D^{-1}[F] = f_t = f(t)$$

Пример 9.1.4.-1:

$$F = \frac{d^3 + d^4 + d^5 + d^6 + d^9}{1 + d + d^3} =$$

$$= d^3 + d^5 + d^6 + d^7 + d^{10} + \dots$$

Соответственно последовательность: 00010111001....

9.1.5. Для любого ненулевого полинома $a(d)$, если $h(d)/q(d)$ – обратимая функция, то ее оригинал такой же, как у функции $a(d)h(d)/(a(d)q(d))$.

Это означает, что обратимую функцию можно привести к функции вида – *нормализовано обратимой*, т.е. такой, что $h(d)$ и $q(d)$ взаимно просты и $q(0)=1$.

9.1.6. Пусть последовательность периодическая с предпериодом длины τ и периодом длины T

$$f(t): a_0 a_1 \dots a_{\tau-1} \cdot a_{\tau} a_{\tau+1} \dots a_{\tau+T-1}, \dots$$

тогда, следуя п.9.1.3 и 9.1.4, получим изображение

$$F = a_0 + a_1 d + \dots + a_{\tau-1} d^{\tau-1} +$$

$$+ d^{\tau} \times \frac{a_{\tau} + a_{\tau+1} d + \dots + a_{\tau+T-1} d^{T-1}}{1-d^T}$$

После приведения к общему знаменателю получим изображение в виде обратимого отношения двух полиномов относительно d .

9.1.7. И наоборот, пусть изображение представлено обратимым отношением полиномов $F=h(d)/q(d)$, где $q(d) \neq 0$. Функцию F всегда можно представить в виде

$$F = P(d) + d^{\tau} r(d)/q(d),$$

где $P(d)$ - полином относительно d и $\deg(r) < \deg(q)$. Если $r(d)=0$, то F является полиномом, а последовательность $f(t)$ имеет конечную длину. Если же $r(d) \neq 0$, то, обозначив через T показатель, которому принадлежит $q(d)$, можно записать:

$$q(d)b(d) = 1 - d^T$$

$$F = P(d) + d^{\tau} r(d)b(d)/(1 - d^T),$$

где $\deg(r(d)b(d)) < T$. Это означает, что последовательность $f(t)$ периодична с периодом T .

Пример 9.1.7.-1:

$$\begin{aligned}
F &= \frac{d^3 + d^4 + d^5 + d^6 + d^9}{1 + d + d^3} = \\
&= d^3 + d^5 + d^6 + d^7 \times \frac{1 + d}{1 + d + d^3} = \\
&= d^3 + d^5 + d^6 + d^7 \times \frac{(1 + d + d^2 + d^4)(1 + d)}{1 + d^7} = \\
&= d^3 + d^5 + d^6 + d^7 \times \frac{1 + d^4 + d^5}{1 + d^7}
\end{aligned}$$

Получили последовательность в оригинале

$$f(t): 0001011.1000110, 1000110, 1. \dots$$

9.2. Передаточная функция

Для двухполюсного ЛА канонические уравнения:

$$s(t) := \mathbf{G} s(t-1) + \mathbf{C} a(t-1) \quad (9.2')$$

$$b(t) = \mathbf{D} s(t) + e a(t), \quad (9.2'')$$

Пусть \mathbf{S} обозначает вектор-столбец, i -я координата которого равна изображению i -й координаты вектора состояния $s(t)$, A и B являются изображениями последовательностей $a(t)$ и $b(t)$. Тогда, применив к (2) D-преобразование, получим

$$\mathbf{S} = d\mathbf{G}\mathbf{S} + d\mathbf{C}A$$

$$B = \mathbf{D}\mathbf{S} + eA,$$

где из (11) следует, что

$$-d\mathbf{C}A = (d\mathbf{G} - \mathbf{I})\mathbf{S}$$

$$\mathbf{S} = -(d\mathbf{G} - \mathbf{I})^{-1}d\mathbf{C}A$$

$$B = (e - \mathbf{D}(d\mathbf{G} - \mathbf{I})^{-1}d\mathbf{C})A$$

$$B = \left(e - \frac{\mathbf{D}(d\mathbf{G} - \mathbf{I})^{\Pi}d\mathbf{C}}{\det(d\mathbf{G} - \mathbf{I})} \right) A$$

где $(\mathbf{M})^{\Pi}$ матрица, состоящая из алгебраических дополнений $\det(\mathbf{M})$. Отношение B/A , т.е. отношение изображений выходной и входной последовательностей называется *передаточной функцией*.

$$J = \frac{B}{A} = e - \frac{\mathbf{D}(\mathbf{dG} - \mathbf{I})^n \mathbf{dC}}{\det(\mathbf{dG} - \mathbf{I})}$$

Функция представима в виде

$$J = \frac{h_0 + h_1 d + h_2 d^2 + \dots + h_n d^n}{1 + q_1 d + q_2 d^2 + \dots + q_n d^n}$$

В тоже время знаменатель этой дроби

$$\det(\mathbf{dG} - \mathbf{I}) = \det(\mathbf{d}(\mathbf{G} - (1/d)\mathbf{I})) = d^n \det(\mathbf{G} - (1/d)\mathbf{I})$$

является полиномом двойственным к характеристическому полиному $P(x) = \det(\mathbf{G} - x\mathbf{I})$, т.е. полиному обратной связи.

(Полином $[P(x)]^*$, двойственный полиному $P(x)$ n -й степени, определяется как $x^n P(1/x)$.)

9.3. Связь структуры ЛА и его передаточной функции

Передаточная функция может быть получена исходя из структуры автомата.

Передаточная функция D-триггера $J=d$.

Если автомат может быть представлен в виде последовательного соединения линейных автоматов с передаточными функциями J_k ($k=1,2,\dots,r$) (рис.3),

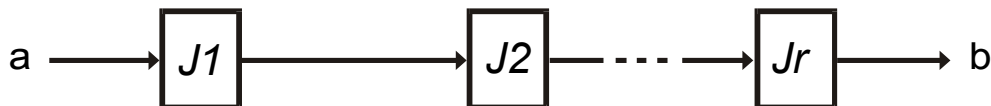


Рис.3. Последовательное соединение

то передаточная функция всей конструкции:

$$J = J_1 \cdot J_2 \cdot \dots \cdot J_r$$

В частности, если путь от входа к выходу в линейном автомате проходит через сумматоры и D-триггеры, то передаточная функция этого пути $J = d^k$, где k - количество триггеров на этом пути.

Для параллельного соединения линейных автоматов (рис.4)

$$J = J_1 + J_2 + \dots + J_r$$

В частности, передаточная функция линейного двухполюсного автомата может быть определена как сумма передаточных функций всех путей от входа к выходу.

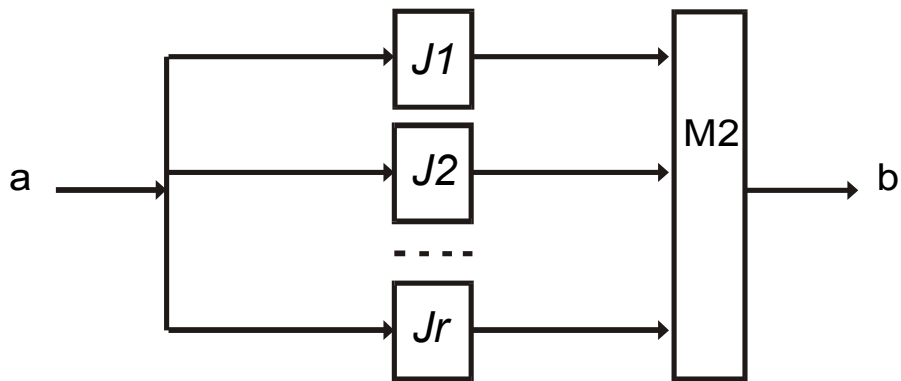


Рис.4. Параллельное соединение

Множество путей, соединяющих вход и выход, может быть бесконечным, если существует путь, содержащий замкнутый контур (рис.5).

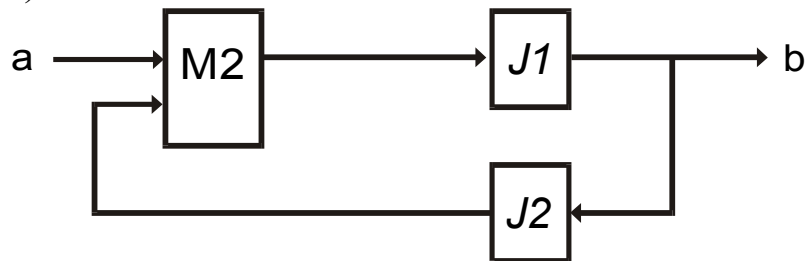


Рис.5. Замкнутый контур

Передаточную функцию для структуры, представленной на рис.3, можно найти из уравнения

$$B = J_1 A + J_2 J_1 B \rightarrow J = B/A = J_1 / (1 + J_1 J_2)$$

Пример 9.3.-1: Т-триггер (рис.6), изменяющий свое состояние и значение выходного сигнала только тогда, когда входной сигнал равен 1, имеет передаточную функцию $J = d/(1+d)$.

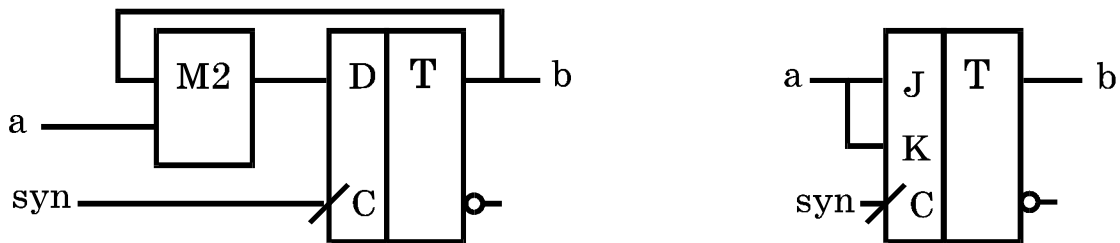


Рис.6. Т-триггер

9.4. Канонические структуры

Передаточной функции вида

конструкцию с искомой передаточной функцией. Вид функции $G(x)$ определен так, чтобы получить $J^{\wedge}(x)$ обратимой и, следовательно, реализуемой.

Пример 9.5.-1. Т-триггер с передаточной функцией $L(d)=d/(1+d)$ является примитивным автоматом, так как существует определенная выше функция

$$G(x) = x/(1+x),$$

$$G(L(d)) = (d/(1+d))/(1+(d/(1+d))) = d$$

Пусть определен автомат с реализацией рис.9 и соответственно передаточной функцией

$$J(d) = \frac{1+d^2}{1+d+d^2+d^3+d^4}$$

В таком случае

$$J(G(x)) = \frac{1+(x/(1+x))^2}{1+(x/(1+x))+(x/(1+x))^2+(x/(1+x))^3+(x/(1+x))^4}$$

$$J^{\wedge}(x) = \frac{1+x^2}{1+x+x^4}$$

с реализацией на Т-триггерах, представленной на рис.10.

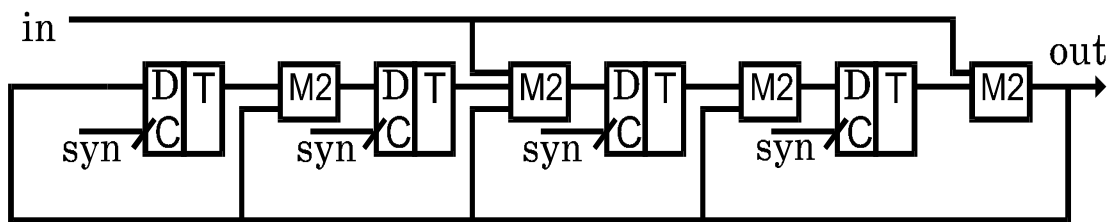


Рис.9. Исходная схема автомата

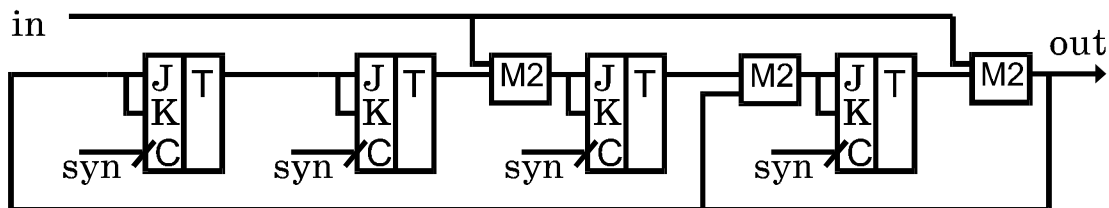


Рис.10. Автомат на Т-триггерах

9.6. Вычисление выходных значений \circ ЛА

Для вычисления выходной последовательности двухполюсного \circ ЛА надо выполнить следующие действия:

1. Найти передаточную функцию J линейного автомата.
2. Вычислить изображение входной последовательности $A=D[a(t)]$, см. пп.9.1.1-5.
3. Вычислить изображение выходной последовательности $B=AJ$.
4. Изображение выходной последовательности представить в виде $B= P(d) + d^T g(d)/(1-d^T)$, см. п.9.1.7 и записать выходную последовательность по вычисленному изображению $b(t)=D^{-1}[B]$.

Задача: Вычислить реакцию \circ ЛА на импульсную функцию $i(t)$ $\{=1$ при $t=0$; $=0$ при $t \neq 0\}$ (какой будет длина предпериода, длина периода?).

9.7. Аннулирующие и аннигилирующие последовательности

Реакция \circ ЛА на входную последовательность, состоящую из одних нулей, представляет собой выходную последовательность одних нулей. Однако, это не так, если память ЛА не нулевая. Для перевода памяти в нулевое состояние требуется на вход подать последовательность, которая называется *аннулирующей*.

Если существует входная последовательность, которая с первого же такта заставляет ЛА с не нулевой памятью выдавать на выходе только нули, то такая последовательность называется *аннигилирующей*.

Аннулирующие и аннигилирующие последовательности полезны при использовании ЛА для исправления ошибок.

9.7.1. Аннулирующую последовательность будем искать как конечной длины последовательность $u(t)$, которая, будучи приложена в момент τ , ликвидирует реакцию \circ ЛА на произвольную входную последовательность $y(t)$ конечной длины, приложенную до момента τ .

Пусть ЛА имеет нормализованно обратимую передаточную функцию $J=h(d)/q(d)$. К входу \circ ЛА с момента $t=0$ и до момента $t=\tau-1$ приложена последовательность $y(t)$. Представим изображение реакции $B=YJ$ в следующем виде: $B= P(d) + d^T r(d)/q(d)$,

где $P(d)$ полином с $\deg(P)=\tau-1$ (может быть с нулевыми коэффициентами при старших степенях). Изображение выходной последовательности, совпадающей с $b(t)$ с момента $t=\tau$, будет $B_\tau=r(d)/q(d)$.

Изображение полной реакции ${}_0J_A$, начиная с момента $t=\tau$, от последовательностей $y(t)$ и $u(t)$ равно:

$$B_\tau + UJ = \frac{r(d)}{q(d)} + \frac{V(d)h(d)}{q(d)} = \frac{r(d)+V(d)h(d)}{q(d)} \quad (9.3)$$

где $V(d)$ – полином – изображение искомой последовательности $u(t)$ (в силу конечности $u(t)$).

Полином $V(d)$ будем искать, исходя из следующих соображений. Для $h(d)$ и $q(d)$ выполняется соотношение

$$1 = h(d)a_h(d) + q(d)a_q(d),$$

которое можно получить, воспользовавшись алгоритмом Евклида нахождения н.о.д. двух полиномов. В этом алгоритме для полиномов $p_0(x)$ и $p_1(x)$, при $\deg(p_0) \geq \deg(p_1)$, выполняется цепочка делений

$$\begin{aligned} p_0(x) &= p_1(x)e_1(x) + p_2(x), \quad \deg(p_2) < \deg(p_1), \\ p_1(x) &= p_2(x)e_2(x) + p_3(x), \quad \deg(p_3) < \deg(p_2), \\ p_2(x) &= p_3(x)e_3(x) + p_4(x), \quad \deg(p_4) < \deg(p_3), \\ &\dots\dots\dots, \end{aligned}$$

до получения остатка равного нулю. Предыдущий остаток является н.о.д.. Если он не содержит x , то исходные полиномы взаимно простые. Последовательной подстановкой можно установить, что, если

н.о.д. $(p_0(x), p_1(x)) = p_2$, то $p_2 = p_0(x) - p_1(x)e_1(x)$, иначе

н.о.д. $(p_0(x), p_1(x)) = p_n = (-1)^n [p_0(x) - p_1(x)(1 + e_1(x) \dots e_{n-1}(x))]$

Отсюда следует, поскольку $h(d)$ и $q(d)$ взаимно просты, то

$$1 = h(d)a_h(d) + q(d)a_q(d).$$

Преобразуя

$$\frac{1 - h(d)a_h(d)}{q(d)} = a_q(d)$$

и умножив обе части на $r(d)$, получим

$$\frac{r(d) - h(d) r(d) a_h(d)}{q(d)} = r(d) a_q(d)$$

Установив подобие с последней дробью выражения (3), положим $V(d) = -r(d)$. Заметим, что изображение реакции на обе последовательности $B_\tau + UJ = r(d) a_q(d)$ является изображением последовательности конечной длины.

Пример 9.7.1.-1: Пусть двухполюсный ЛА над полем $GF(2)$ имеет передаточную функцию

$$J = \frac{h(d)}{q(d)} = \frac{1 + d + d^2}{1 + d + d^2 + d^3}$$

Найдем последовательность $u(t)$ конечной длины, которая будучи приложенной в момент $t=2$, нейтрализует реакцию на единичный импульс. В этом случае изображение выходной реакции

$$B = \frac{1 + d + d^2}{1 + d + d^2 + d^3} = 1 + d^2 \times \frac{d}{1 + d + d^2 + d^3}$$

$$r(d) = d$$

$$h(d) = q(d) e_1(d) + p_2(d) =$$

$$= 1 + d + d^2 + d^3 = (1 + d + d^2)d + 1$$

$$a_h(d) = d, \quad a_q = 1$$

Изображение искомой последовательности

$$U = V(d) = -r(d) a_h(d) = d^2$$

Соответственно последовательность $u(t)$, начиная с момента $t=2$, равна $u(t)$: 0010...

Изображение выходной последовательности $r(d) a_q(d) = d$.

Соответственно выход, начиная с момента $t=2$, – 010...

9.7.2. Аннигилирующая последовательность, будучи приложенной в момент $t=\tau$, должна мгновенно (в том же такте) установить только нулевые выходные значения. Поэтому, если такая последовательность существует, то

$$B_\tau + U^* J = \frac{r(d)}{q(d)} + U^* \times \frac{h(d)}{q(d)} = 0$$

$$U^{\wedge} = -B_{\tau}/J = -r(d)/h(d) \quad (9.4)$$

Аннигилирующая последовательность $u^{\wedge}(t)$ существует тогда, когда отношение $r(d)/h(d)$ обратимо. Чтобы это отношение было обратимо при любых входных воздействиях, должно выполняться $h(0) \neq 0$, т.е. ЛА не должен быть автоматом Мура или иначе выход должен зависеть от текущего значения на входе. Согласно (4) аннигилирующая последовательность $u^{\wedge}(t)$ может не иметь конечной длины, а значит и не сбрасывать память автомата в нулевое состояние.

Пример 9.7.2.-1: В условиях предыдущего примера изображение аннигилирующей последовательности $u^{\wedge}(t)$, начинающейся с момента $t=2$

$$U^{\wedge} = \frac{-r(d)}{h(d)} = \frac{d}{1 + d + d^2}$$

Приведя к виду в соответствии с п. 9.1.7, получим

$$U^{\wedge} = \frac{d + d^2}{1 + d^3}$$

Аннигилирующая последовательность $u^{\wedge}(t)$: 011,011,...

10. Умножение и деление линейными автоматами

Продemonстрируем возможность использования линейных автоматов для выполнения операций умножения и деления двоичных полиномов. Такие операции выполняются при кодировании и декодировании помехозащищенных кодов и в различных цифровых фильтрах.

10.1. Умножение

Рассмотрим линейный автомат с передаточной функцией

$$J = 1 + h_1d + h_2d^2 + \dots + d^n,$$

пусть задана входная последовательность

$$a_t: a_0, a_1, \dots, a_k, 0, \dots,$$

тогда изображение выходной последовательности

$$\begin{aligned} B = J * A &= (1 + h_1d + h_2d^2 + \dots + d^n) \times \\ &\times (a_0 + a_1d + a_2d^2 + \dots + a_kd^k) = \\ &= b_0 + b_1d + b_2d^2 + \dots + b_{(n+k)}d^{n+k} \end{aligned}$$

Происходит умножение полиномов.

В том случае, если коэффициенты произвольного полинома следуют, начиная с коэффициента при наивысшей степени, то полученные результаты интерпретируются следующим образом. Один из множителей $C(x)$ фиксирован и задан структурой автомата. Если

$$J = 1 + h_1d + h_2d^2 + \dots + d^n, \text{ то}$$

$$C(x) = x^n + h_1x^{(n-1)} + \dots + h_{(n-1)}x + 1$$

Коэффициенты произвольного полинома степени не выше k задаются $(k+1)$ цифрами входной последовательности, начиная с коэффициента при старшей степени, после которых должны следовать n нулей. При нулевых начальных значениях триггеров в результате на выходе будет получена последовательность, $(k+n+1)$ цифр которой интерпретируются как коэффициенты искомого полинома. Первая цифра выходной последовательности является коэффициентом при наивысшей степени $x^{(k+n)}$.

Пример 10.1.-1: линейные автоматы с $J(d)=1+d+d^3$ рис.11, рис.12 можно использовать для умножения на фиксированный полином $C(x)=x^3+x^2+1$.

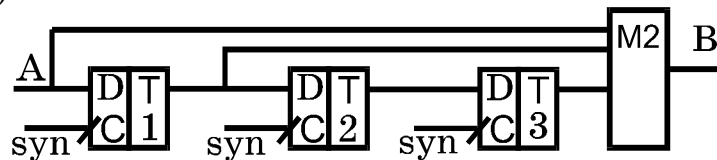


Рис.11. ЛА односумматорный для умножения

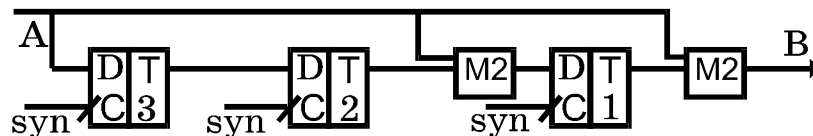


Рис.12. ЛА многосумматорный для умножения

Пусть $C(x)$ умножается на полиномы степени не выше 4.

Пример 10.1.-2: $A(x) = x^2 + x$,

тогда входная последовательность $a : 0 0 1 1 0'0 0 0$,

и ее изображение $A(d) = d^2 + d^3$.

Изображение выходной последовательности

$$B(d) = J(d) * A(d) = (1+d+d^3) * (d^2+d^3) = d^2+d^4+d^5+d^6,$$

выходная последовательность $b : 00101110$,

соответственно $B(x) = x^5 + x^3 + x^2 + x$.

10.2. Деление

При делении полинома $A(x)$ степени k на полином $C(x)$ степени $n \leq k$ однозначно определяются частное $B(x)$ - полином степени $(k-n)$ и остаток $R(x)$ - полином степени m ($0 \leq m < n$).

Рассмотрим линейный автомат с передаточной функцией

$$J = d^n / (1 + q_1 d + q_2 d^2 + \dots + d^n).$$

Пусть задана входная последовательность $a(t)$: $a_0, a_1, \dots, a_k, 0, \dots$. Изображение выходной последовательности в виде полинома получаем как разложение в степенной ряд непосредственным делением числителя на знаменатель выражения $A(d) * J(d)$. Начальные $k+1$ значения выходной последовательности будут иметь следующий вид: $b : 0, 0, \dots, 0, b_n, \dots, b_{(k-n)}$, при этом $b_n = a_0$.

Эти результаты должны интерпретироваться следующим образом. Произвольный полином степени не выше k

$$A(x) = a_0 x^k + \dots + a_k,$$

делится на фиксированный полином

$$C(x) = x^n + q_1 x^{(n-1)} + \dots + q_{(n-1)} x + 1.$$

При нулевом начальном состоянии за время $t=k+1$ будет получена выходная последовательность, в начале которой будет n нулей затем значения коэффициентов частного, начиная с коэффициента при $x^{(k-n)}$, и остаток, коэффициенты полинома которого содержатся в памяти автомата. Коэффициент при $x^{(n-1)}$ - старшей степени полинома остатка содержит триггер с номером 1 и т.д.

Пример 10.2.-1: линейный автомат с $J(d) = d^3 / (1 + d + d^3)$ рис.13 или рис.14 можно использовать для деления на фиксированный полином $C(x) = x^3 + x^2 + 1$.

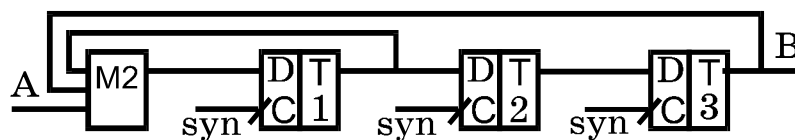


Рис.13. ЛА односумматорный для деления

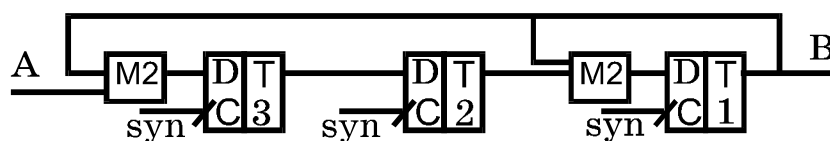


Рис.14. ЛА многосумматорный для деления

Предположим, делимое произвольные полиномы степени не выше 6. Например, $A(x)=x^6+x^5+x^4+x^3+1$, тогда входная последовательность $a : 1111001$, изображение входной последовательности $A(d)=1+d+d^2+d^3+d^6$. Вычислим изображение выходной последовательности

$$\begin{array}{r}
 d^3 + d^4 + d^5 + d^6 + 0 + 0 + d^9 \\
 \underline{d^3 + d^4 + 0 + d^6} \\
 0 + d^5 + 0 + 0 \\
 \quad d^5 + 0 + 0 + 0 \\
 \quad \underline{d^5 + d^6 + 0 + d^8} \\
 \qquad d^6 + 0 + d^8 + d^9 \\
 \qquad \underline{d^6 + d^7 + 0 + d^9} \\
 \qquad \qquad d^7 + d^8 + 0
 \end{array}
 \quad
 \begin{array}{r}
 \overline{1 + d + 0 + d^3} \\
 d^3 + 0 + d^5 + d^6
 \end{array}$$

Ее вид $b : 0001011$. Это означает, что

$$\begin{aligned}
 B(x) &= x^3 + x + 1 \\
 R(x) &= x^2 + x
 \end{aligned}$$

Проверим полученный результат $B(x) = A(x)/C(x)$

$$\begin{array}{r}
 x^6 + x^5 + x^4 + x^3 + 0 + 0 + 1 \\
 \underline{x^6 + x^5 + 0 + x^3} \\
 0 + x^4 + 0 + 0 \\
 \quad x^4 + 0 + 0 + 0 \\
 \quad \underline{x^4 + x^3 + 0 + x} \\
 \qquad x^3 + 0 + x + 1 \\
 \qquad \underline{x^3 + x^2 + 0 + 1} \\
 \qquad \qquad x^2 + x + 0
 \end{array}
 \quad
 \begin{array}{r}
 \overline{x^3 + x^2 + 0 + 1} \\
 x^3 + 0 + x + 1
 \end{array}$$

Аналогично можно получить схему, которая умножает на один полином, а делит на другой.

11. Обобщения

11.1. Многоканальный аналог двухполюсного ЛА

Двухполюсный автомат имеет один вход (кроме входа синхросигнала) и один выход. Многоканальный аналог двухполюсного автомата A , обозначаемый $A_{(k)}$, представляет собой автомат с k входами и k выходами (рис.15). Входные и выходные вектора соответствуют входным и выходным последовательностям длины k автомата A .

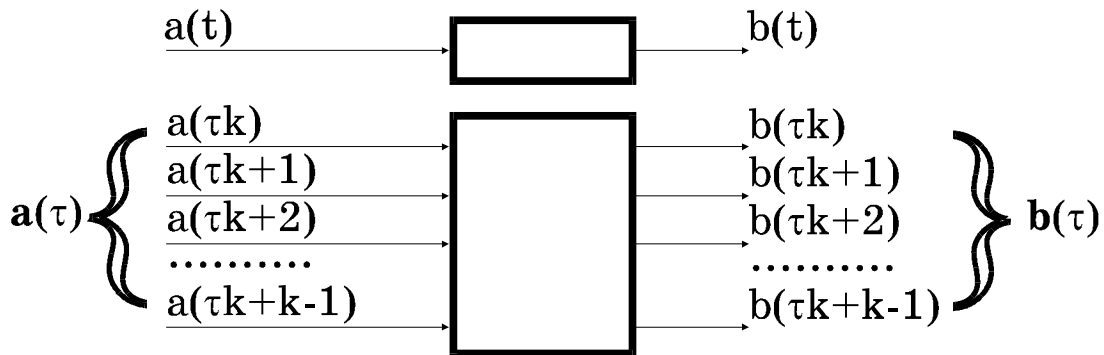


Рис.15. Двухполюсный ЛА и его многоканальный аналог

Таким образом, имея k каналов вместо одного, автомат $A_{(k)}$ работает в k раз быстрее, чем автомат A . Структура характеристических матриц k -канального аналога линейного автомата A следует из уравнений (3.1).

$$\begin{aligned}
 G_{(k)} &= G^k & B_{(k)} &= \begin{bmatrix} G^{k-1}C & G^{k-2}C & \dots & G^2C & GC & C \end{bmatrix} \\
 D_{(k)} &= \begin{bmatrix} D \\ DG \\ DG^2 \\ \vdots \\ DG^{k-1} \end{bmatrix} & E_{(k)} &= \begin{bmatrix} D & 0 & 0 & \dots & 0 & 0 \\ DG & 0 & 0 & \dots & 0 & 0 \\ DGC & DG & D & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ DG^{k-2}C & DG^{k-3}C & DG^{k-4}C & \dots & DC & E \end{bmatrix}
 \end{aligned}$$

Размерность состояний автомата $A_{(k)}$ осталась такой же, как и у автомата A . Входные и выходные цепи разумеется усложняются.

11.2. Линейные автоматы над конечным полем

До сих пор рассматривались ЛА с двоичными сигналами $\{0,1\}$. Если считать, что значения сигналов принадлежат конечному множеству $\{0,1,\dots,p-1\}$, где p - простое число, то операции (сложение и умножение на константу) над сигналами в линейном автомате должны выполняться по правилам конечного поля $GF(p)$. Реализация таких автоматов, хотя и возможна в двоичном элементном базисе, но удобнее программный вариант.

11.3. Линейные автоматы над полем Галуа

Достаточно универсальным обобщением, но в тоже время удобным для реализации в двоичном элементном базисе, являются ЛА над полем Галуа $GF(2^k)$ - полем полиномов степени не выше $k-1$ над полем $GF(2)$. Теория ЛА справедлива, если 2 заменить любым простым целым числом, но аппаратная реализация эффективна для 2. В этом случае двоичные ЛА, т.е. автоматы над полем $GF(2)$ преобразуются следующим образом - каждая линия связи заменяется шиной из k линий, пронумерованных числами $0,1,\dots,k-1$. По шинам передаются коэффициенты полинома над полем $GF(2)$

$$A(x) = a_0 + a_1x + a_2x^2 + \dots + a_{k-1}x^{k-1}$$

Каждый сумматор заменяется k сумматорами (рис.16).

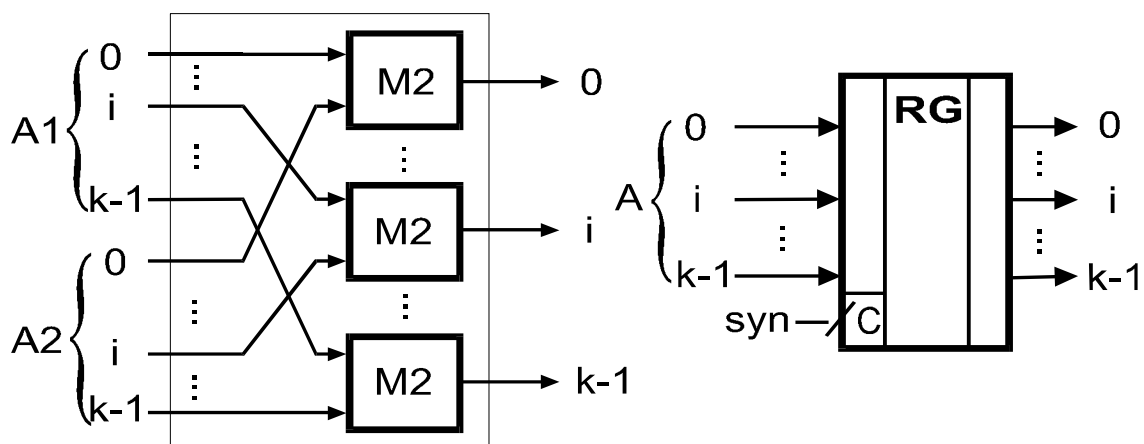


Рис.16. Сумматор и память ЛА над полем Галуа

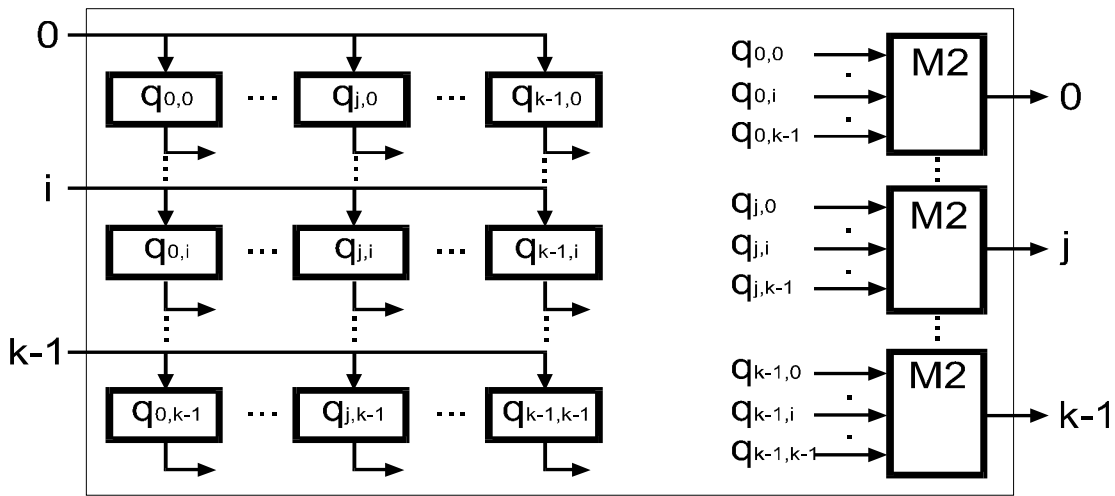


Рис.17. Умножитель ЛА над полем Галуа

Вместо умножения на постоянный коэффициент происходит умножение на постоянный полином

$$C(x) = c_0 + c_1x + c_2x^2 + \dots + c_{(k-1)}x^{k-1}$$

Это умножение реализуется как произведение \mathbf{Qa} входного вектора \mathbf{a} на матрицу \mathbf{Q} .

$$\mathbf{Q} = \|q_{ij}\|_{k \times k} = c_0 \mathbf{I} + c_1 \mathbf{M}_{P(x)}^{\text{TP}} + c_2 (\mathbf{M}_{P(x)}^{\text{TP}})^2 + \dots + c_{k-1} (\mathbf{M}_{P(x)}^{\text{TP}})^{k-1}$$

где $\mathbf{M}_{P(x)}^{\text{TP}}$ матрица (5") транспонированная к сопровождающей неприводимый полином над полем $\text{GF}(p)$

$$P(x) = p_0 + p_1x + p_2x^2 + \dots + p_{n-1}x^{n-1} + x^n,$$

порождающий поле $\text{GF}(2^k)$ (операции над полиномами выполняются в поле по модулю $P(x)$). Выбор конкретного полинома $P(x)$ определяется тем приложением, ради которого и строится этот автомат.

Таким образом, умножитель реализуется в виде схемы, представленной на рис.17.

ГЛАВА III. УПРАВЛЯЮЩИЕ АВТОМАТЫ

При проектировании вычислительного устройства, выполняющего сложную обработку цифровой информации, одним из вариантов декомпозиции является представление синхронного вычислителя в виде композиции двух автоматов операционного и управляющего - рис.1.

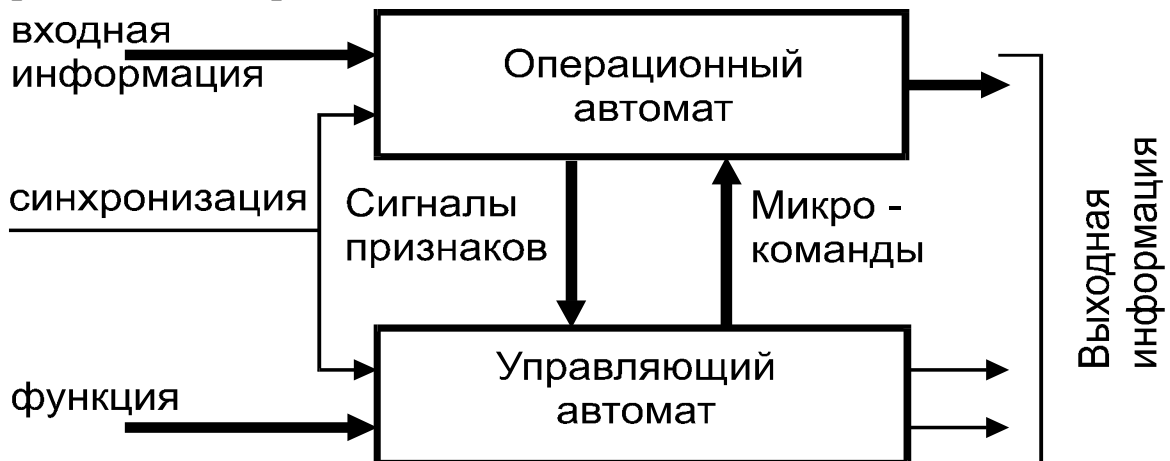


Рис.1. Структура вычислителя

Операционный автомат (ОА) это, в свою очередь, некоторая композиция из автоматов, но если все регистры этих автоматов синхронизируются одинаково, то ОА можно представлять себе как один автомат Мили. Для исключения гонок по замкнутым цепям управляющий автомат (УА) должен быть автоматом Мура.

Совместная работа этих двух автоматов может быть описана следующим образом. На границе такта (фронт, синхронизирующий регистры автоматов) изменяется содержимое регистров как ОА так и УА и соответственно выходные значения автоматов. Это приводит к тому, что УА формирует новые значения сигналов для ОА (сигналы эти на рис.1 обозначены как *микрокоманда*). В том же такте ОА сформирует новые значения сигналов для УА (сигналы эти на рис.1 обозначены как *признаки*).

Приведем также терминологию, сложившуюся в инженерной практике проектирования вычислительных устройств такого типа.

Микропрограмма - любая формализованная форма описания совместной работы операционного и управляющего автоматов - будем использовать *блок-схемы* и *блок-тексты*.

Микрооперация - базисное (элементарное) действие, выполняемое в ОА.

Микроблок - набор микроопераций, выполняемых в ОА одновременно (в одном такте).

Признак (условие) - логическое значение, используемое при переходе к одному из возможных шагов алгоритма; - результат выполнения микрооперации.

Микрокоманда - набор сигналов, поступающий из УА в ОА и интерпретируемый как управляющий, т.е. необходимый для выполнения всех микроопераций одного микроблока.

Микрокомандой также иногда называют слово управляющей памяти, являющейся частью УА. Для различения этих понятий слово управляющей памяти будем называть *микроинструкцией*.

1. Варианты взаимодействия ОА и УА

Схема взаимодействия автоматов, более детальная, чем на рис. 1, представлена на рис.2.

Отметим, что сигналы признаков могут быть двух типов

- а) $PA(t)=f_A(Y(t))$ зависит без сдвига от сигналов управления,
- б) $PB(t+1):=f_B(Y(t))$ зависит со сдвигом от сигналов управления.

Минимальная длительность такта работы схемы определяется наиболее длинными цепями между регистрами. Для схемы на рис.1, которую будем называть последовательной схемой взаимодействия, зададимся (так чаще всего бывает), что такой критической цепью является цепь (CL_y,CL_a,CL_p,RG). Поэтому длительность такта определяется:

$$T > t_y + t_a + t_p + t_{rg},$$

где t_j - время установления соответствующего компонента цепи.

Чтобы сократить длину этой цепи, применяют другой вариант взаимодействия автоматов - *конвейерный*- рис.3. УА для этого варианта взаимодействия отличается от УА варианта последовательного взаимодействия добавленным регистром RG_FF (регистром флагов) и несколько иной компоновкой регистров и комбинационных схем. При этом варианте взаимодействия такой длинной цепи, как в предыдущем случае, нет. Эта цепь разделена

регистрами RG_FF и RG_MK (регистр микрокоманды) на две цепи. Продолжительность такта становится меньше и ее можно определить следующим образом:

$$T > \max[t_a, (t_p + t_y)] + t_{rg} ,$$

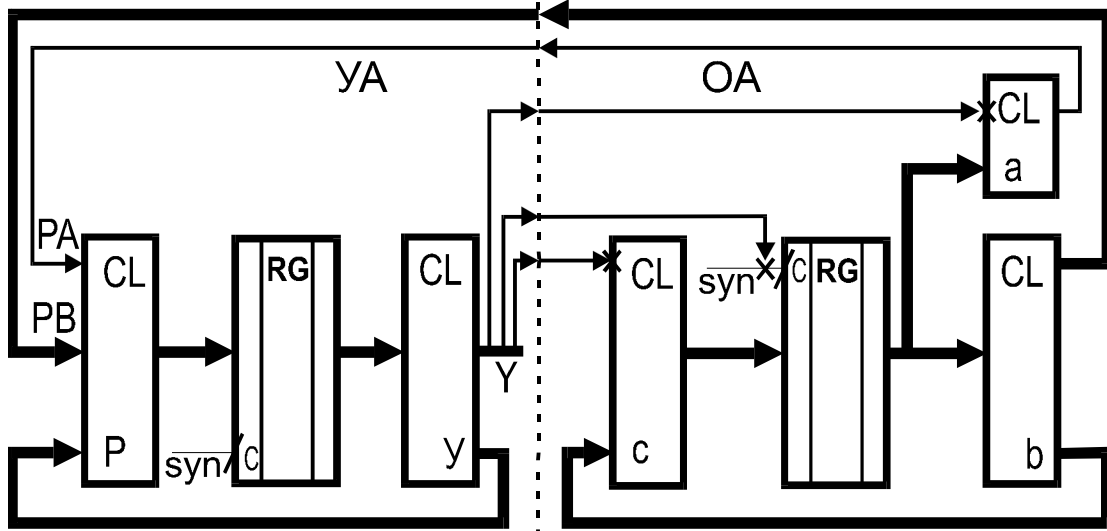


Рис.2. Последовательный вариант взаимодействия

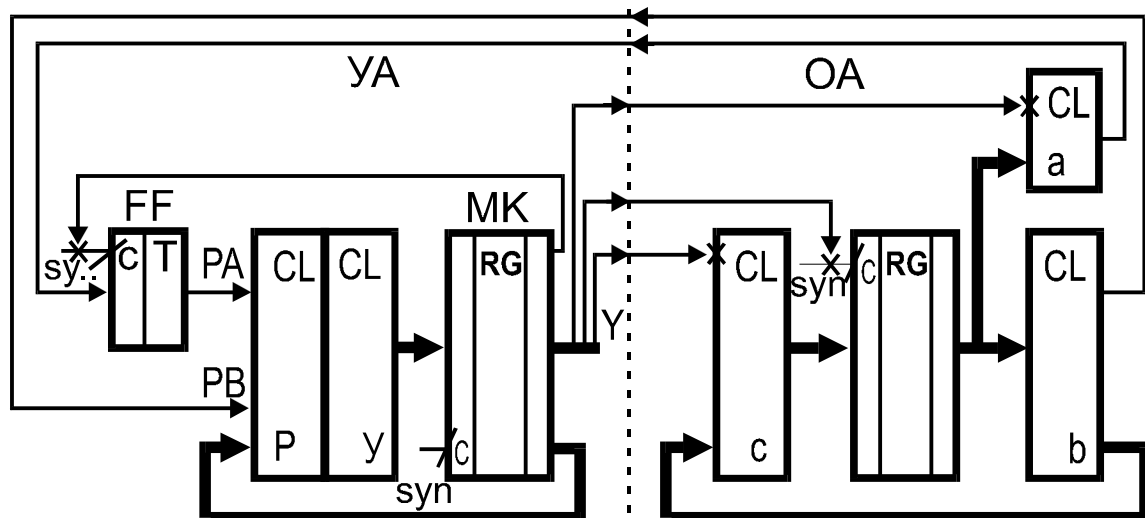


Рис.3. Конвейерный вариант взаимодействия

При конвейерном варианте взаимодействия $PA(t+1) := f(Y(t))$, т.е. и эти значения стали зависеть со сдвигом от сигналов управления. Тогда фрагмент микропрограммы

mS { ...; pA = f(...) }
«GO(pA; mi, mj)»,

выполняемый в последовательной схеме за один такт, в конвейерном варианте за один такт выполнен быть не может, и должен быть модифицирован следующим образом:

$$\begin{aligned} & mS \{ \dots, pA := f(\dots) \} \\ & mS' \{ \text{нет операции} \} \\ & \quad \ll GO(pA; m_i, m_j) \gg \end{aligned}$$

Таким образом, время выполнения этого фрагмента не только не уменьшилось, но даже возросло, несмотря на уменьшение продолжительности каждого из тактов. Зато, во всех остальных случаях (при безусловных переходах, при переходах по значению РВ) время выполнения микропрограммы уменьшается.

Особенностью реально существующих структур является большая разрядность микрокоманды и малая разрядность (чаще всего 1) анализируемых на каждом шаге признаков. Поэтому является оправданной такая структуризация комбинационной части схемы управляющего автомата, в которой выделяется функция мультиплексирования одного признака на каждом шаге выполнения алгоритма и реализация остальной значительной части логики в виде ПЗУ.

2. Основные способы адресации микрокоманд

В простейшем варианте, в микропрограмме переходы только безусловные. В таком случае УА является автономным синхронным автоматом.

В общем случае функция переходов УА зависит от признаков. Условимся о некоторых ограничениях, позволяющих упростить схему на начальных этапах проектирования (от которых легко впоследствии и отказаться):

- на каждом шаге процесса вычислений ветвление может осуществляться только по одной двузначной переменной - условию (т.е. ветвление возможно лишь на два направления);

- начальные значения всех регистров УА являются нулевыми. Впредь на схемах не будем показывать цепей установки нулевых начальных значений в регистрах УА.

Для реализации, в самом общем случае, микропрограмм произвольной структуры будем строить УА так, чтобы основным материальным носителем управляющей компоненты микропрограммы являлась бы управляющая память, реализованная, например, в виде ПЗУ. В этом случае слово управляющей памяти -

микроинструкция - состоит из двух составных частей: *микрокоманды* и *адресной части*.

Адресная часть микроинструкции содержит информацию, позволяющую в следующем такте работы выбрать (указать) новый адрес управляющей памяти. Реализация именно этого момента является основным предметом дальнейшего рассмотрения и определяет, в основном, структуру, объем аппаратуры и быстроедействие УА. При этом подлежит рассмотрению реализация следующих типов переходов как между шагами алгоритма, так, соответственно, и между микроинструкциями:

- безусловный переход,
- условный переход,
- функциональный переход,
- переход к микроподпрограмме с возвратом.

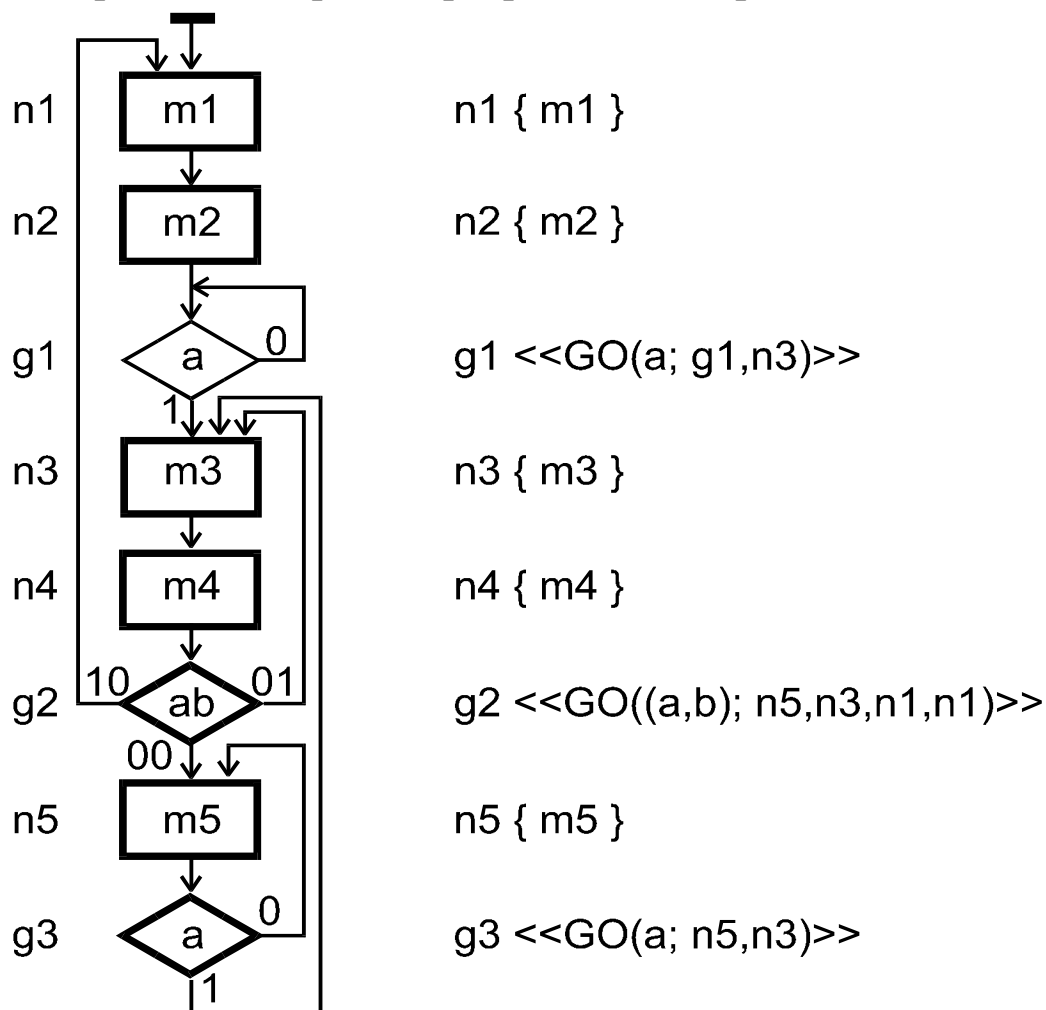


Рис.4. Блок-схема и блок-текст микропрограммы

Будем изучать работу управляющих автоматов различной структуры, демонстрирующие основные применяемые варианты адресации микроинструкций, на алгоритме, показанном на рис.4.

Укажем на некоторые особенности этого алгоритма: Оператор перехода (условная вершина), помеченный меткой g1, называют ждущим. Оператор, помеченный меткой g2, использует для перехода 4-значный признак, что не удовлетворяет вышеуказанному ограничению. Поэтому потребуются эквивалентные преобразования алгоритма для того, чтобы удовлетворить этому ограничению.

Алгоритмы эквивалентны, если они преобразуют информацию одинаковым образом, не обязательно за одно и тоже число тактов. Наиболее распространенным приемом эквивалентного преобразования алгоритмов и микропрограмм является включение микроблоков и, соответственно, тактов, в которых не выполняется модификация памяти ОА - "**нет операции**".

2.1. Схема с адресным ПЗУ

Начнем рассмотрение с управляющего автомата, который называют микропрограммным автоматом Уилкса. Функцию перехода и функцию выхода реализуем в виде ПЗУ.

В ПЗУ (ROM_1), реализующем функцию выхода, следует разместить микрокоманды; при этом их распределение по определенным адресам совершенно произвольно, за исключением начальной микрокоманды, которая должна располагаться по нулевому адресу в силу вышеуказанного ограничения (сброс в ноль RG УА в начальный момент времени). ПЗУ (ROM_2), реализующее функцию переходов автомата, можно трактовать как адресное ПЗУ. Ячеек в адресном ПЗУ в два раза больше, чем в ПЗУ микрокоманд. Каждой ячейке ПЗУ микрокоманд соответствуют две ячейки в адресном ПЗУ, в которых записываются два альтернативных адреса.

Микропрограмма для конвейерного варианта может измениться в силу замечаний, сделанных в разделе "Варианты взаимодействия ОА и УА". Для схемы на рис.7 ограничения на расположение стартовой микрокоманды иное.

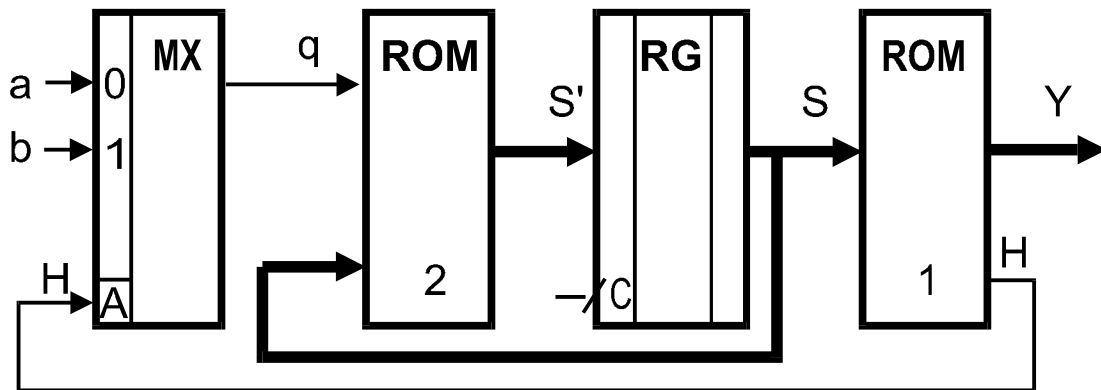


Рис.5. УА с адресным ПЗУ; последовательный вариант

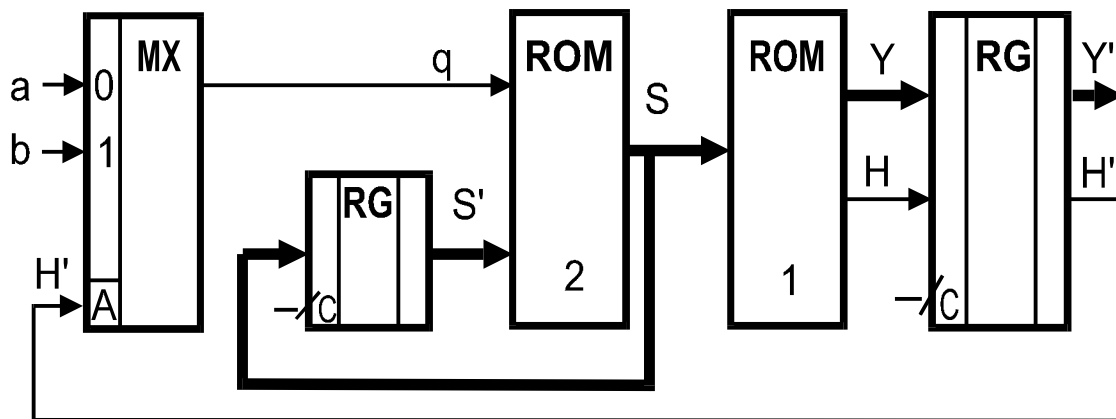


Рис.6. УА с адресным ПЗУ; конвейерный вариант

		ROM 1			ROM 2		
		S	Y	H	S'	q	S
n1	{ m1 }	0	m1	X	0	0	1
					0	1	1
n2	{ m2 }	1	m2	0	1	0	2
	«GO(a; d1,n3)»				1	1	3
d1	{ m0 }	2	m0	0	2	0	2
	«GO(a; d1,n3)»				2	1	3
n3	{ m3 }	3	m3	X	3	0	4
					3	1	4
n4	{ m4 }	4	m4	0	4	0	5
	«GO(a; d2,n1)»				4	1	0
d2	{ m0 }	5	m0	1	5	0	6
	«GO(b; n5,n3)»				5	1	3
n5	{ m5 }	6	m5	0	6	0	6
	«GO(a; n5,n3)»				6	1	3

2.2. Схема с явным указанием альтернативных адресов

Эта схема отличается от предыдущей тем, что, по существу, тот же способ адресации выполнен с использованием только одного ПЗУ. В этом варианте альтернативные адреса записываются в той же микроинструкции, что и микрокоманда.

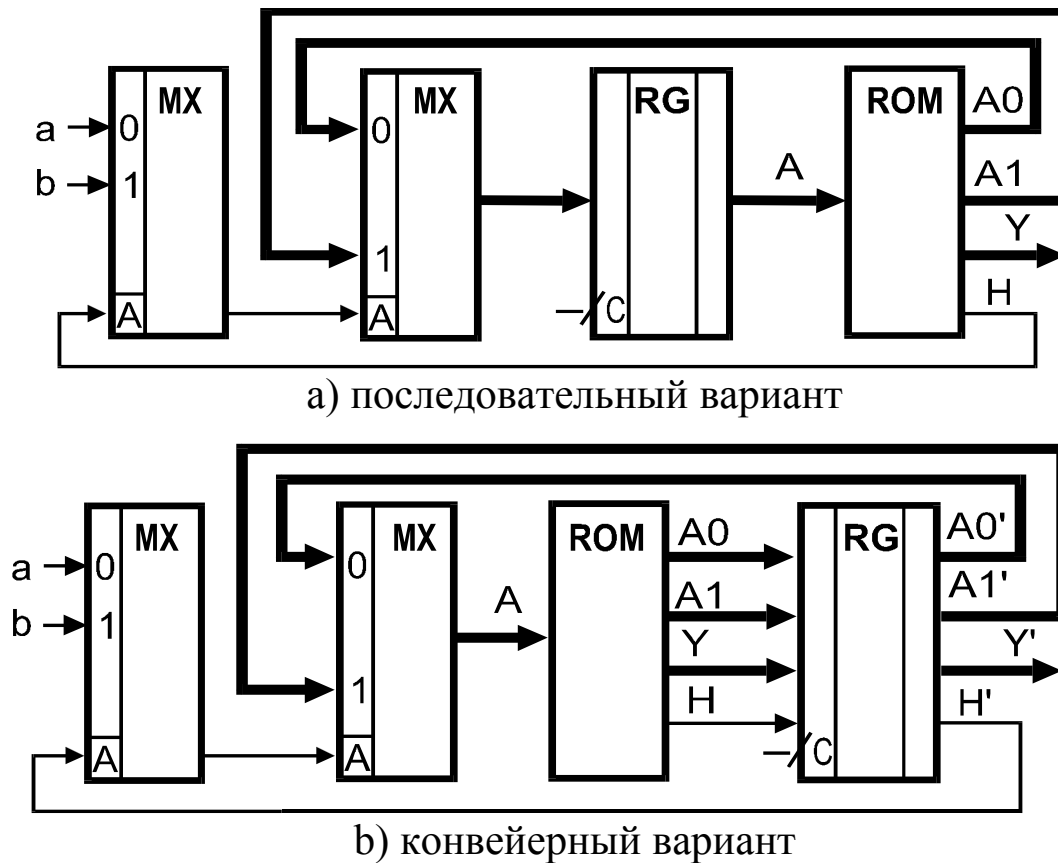
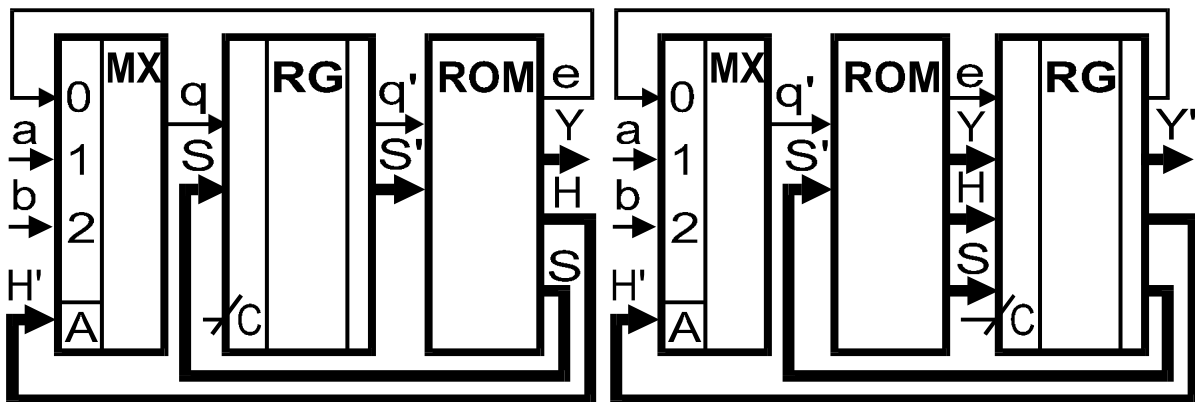


Рис.7. УА с явным указанием альтернативных адресов

		A	Y	H	A0	A1
n1	{ m1 }	0	m1	x	1	1
n2	{ m2 }	1	m2	0	2	3
	«GO(a; d1,n3)»	2	m0	0	2	3
d1	{ m0 }	3	m3	x	4	4
	«GO(a; d1,n3)»	4	m4	0	5	0
n3	{ m3 }	5	m0	1	6	4
n4	{ m4 }	6	m5	0	6	4
	«GO(a; d2,n1)»					
d2	{ m0 }					
	«GO(b; n5,n3)»					
n5	{ m5 }					
	«GO(a; n5,n3)»					

2.3. Схема с частичной записью адреса

При этом способе адресации альтернативные адреса отличаются только одним разрядом (в данном варианте - младшим), формируемым входным сигналом. Остальные разряды адреса указываются вместе с микрокомандой в одной и той же микроинструкции. При безусловном переходе в данном варианте схемы младший разряд также указывается в микроинструкции.



Последовательный вариант

Конвейерный вариант

Рис.8. УА с частичной записью адреса

		Адрес
n1	{ m1 }	(0,0),(2,1)
n2	{ m2 }	(4,0)
	«GO(a; d1,n3)»	1,X
d1	{ m0 }	(1,0)
	«GO(a; d1,n3)»	1,X
n3	{ m3 }	(1,1),(3,1)
n4	{ m4 }	(0,1)
	«GO(a; d2,n1)»	2,X
d2	{ m0 }	(2,0)
	«GO(b; n5,n3)»	3,X
n5	{ m5 }	(3,0)
	«GO(a; n5,n3)»	3,X

S'q'	Y	H	S	e
0 0	m1	0	4	0
0 1	m4	1	2	x
1 0	m0	1	1	x
1 1	m3	0	0	1
2 0	m0	2	3	x
2 1	m1	0	4	0
3 0	m5	1	3	x
3 1	m3	0	0	1
4 0	m2	1	1	x

При адресации одного и того же микроблока различными операторами условного перехода может возникнуть *конфликт адресации*. В этом случае одну и ту же микроинструкцию приходится располагать в различных ячейках управляющей памяти. Если различные операторы условного перехода одними и теми

же значениями признаков указывают на одни и те же микроблоки, то нет и конфликта адресации.

Распределять микроинструкции по ячейкам памяти удобно в следующем порядке:

- связать с различными операторами условного перехода, конфликтующими между собой по адресации, различающиеся между собой старшие разряды адреса;
- распределить микроблоки по ячейкам памяти с учетом назначенных старших разрядов адреса и входных значений, формирующих младший разряд адреса;
- оставшимся нераспределенным микроблокам назначить произвольные свободные адреса.

2.4. Схема с сокращенным тактом

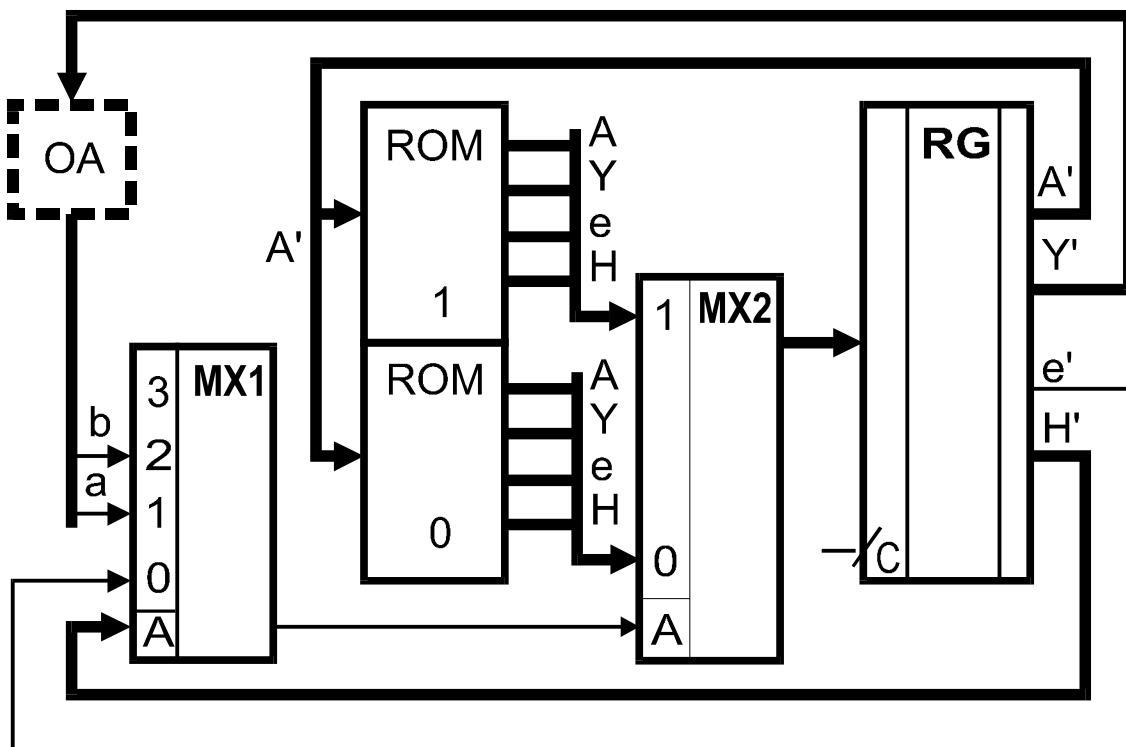


Рис.9. УА с сокращенным тактом

Использование этой схемы позволяет при сохранении преимуществ последовательного варианта взаимодействия сократить наиболее длинные цепи, общие для ОА и УА, до длины цепей конвейерного варианта.

ROM 0

A'	Y	H	A	e
0	m1	0	4	0
1	m0	1	1	x
2	m0	2	3	x
3	m5	1	3	x
4	m2	1	1	x

ROM 1

A'	Y	H	A	e
0	m4	1	2	x
1	m3	0	0	1
2	m1	0	4	0
3	m3	0	0	1

Части схемы критичные по длительности такта (ПЗУ и комбинационная часть операционного автомата) разнесены в разные контуры. Мультиплексор МХ'2, функционально необходимый, реально может отсутствовать, т.к. мультиплексирование может быть реализовано использованием высокоимпедансного состояния выхода ПЗУ. Эта функция реализуется в ПЗУ обычно в 3 - 4 раза быстрее, чем выбор содержимого ячейки по адресу. В этом случае объем аппаратуры остается таким же, как и в схеме с «частичной записью адреса» (рис.8).

Способ адресации, по существу, такой же, как и в предыдущей схеме. Только в рассматриваемой схеме входной сигнал управляет выбором одного из двух блоков ПЗУ (можно интерпретировать этот сигнал как старший разряд адреса).

2.5. Схема с регулярной адресацией

В этой схеме рис.1. при разветвлении процесса вычислений пара альтернативных адресов получается следующим образом: один адрес всегда на единицу больше, чем текущий (т.е. изменяется «регулярным» образом), второй адрес - произвольный и содержится вместе с микрокомандой в микроинструкции.

Элементом, «вычисляющим» адрес, служит счетчик, управляемый входным для УА сигналом. При различных значениях входного сигнала счетчик выполняет две функции: либо прибавляет единицу к значению, которое хранилось в счетчике и являлось текущим адресом, либо загружается значением адреса из управляющей памяти.

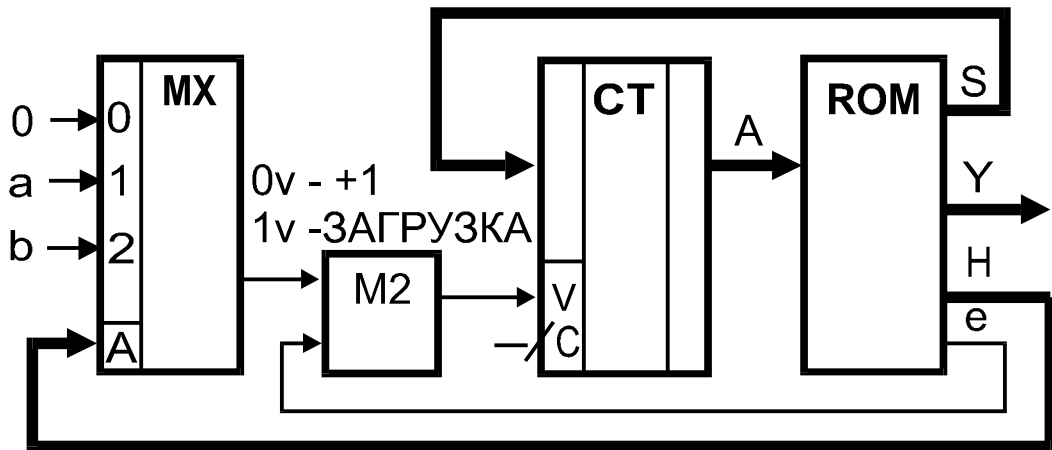


Рис.10. УА с регулярной адресацией; последовательный вариант

В схему введен элемент М2, позволяющий инвертировать значение входного сигнала, что облегчает распределение микроинструкций по ячейкам управляющей памяти.

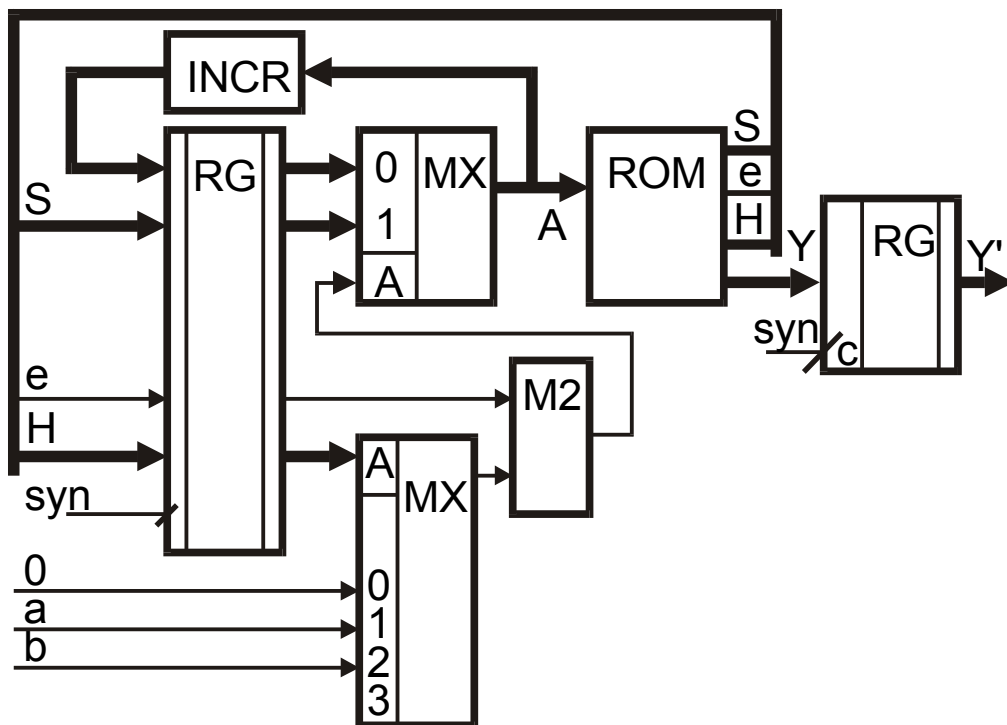


Рис.11. УА с регулярной адресацией; конвейерный вариант

В схеме для конвейерного варианта взаимодействия рис.7. регулярное изменение адреса приходится организовывать так, чтобы не увеличивать число мест конвейера.

		Адрес					
n1	{ m1 }	0	A	Y	H	S	e
n2	{ m2 }	1	0	m1	x	1	x
	«GO(a; d1,n3)»		1	m2	1	3	0
d1	{ m0 }	2	2	m0	1	2	1
	«GO(a; d1,n3)»		3	m3	x	4	x
n3	{ m3 }	3	4	m4	1	0	0
n4	{ m4 }	4	5	m0	2	3	0
	«GO(a; d2,n1)»		6	m5	1	6	1
d2	{ m0 }	5	7	m0	0	3	1
	«GO(b; n5,n3)»						
n5	{ m5 }	6					
	«GO(a; n5,n3)»						

2.6. Схема с естественной адресацией и совмещенным назначением разрядов ячейки ПЗУ

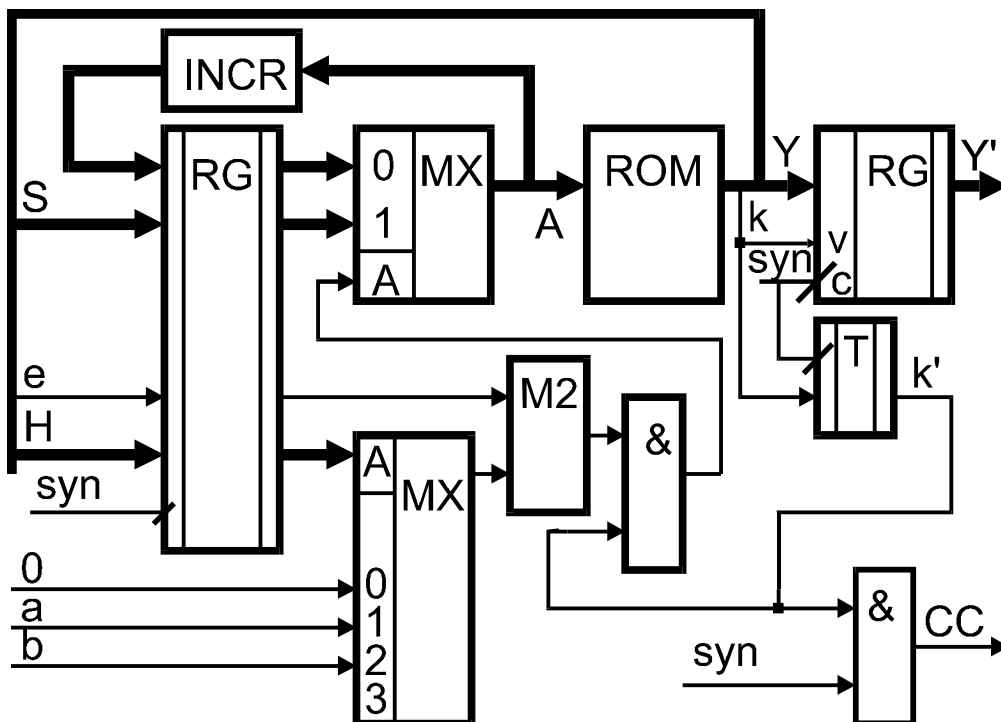


Рис.12. УА с естественной адресацией

Эта схема используется только в конвейерном варианте взаимодействия. Метод вычисления адреса для следующего такта такой же, как и в схеме с регулярной адресацией. (Другой термин

– «естественная» употреблен только ради различения самих схем.) В этой схеме, по сравнению с уже рассмотренными, разряд управляющей памяти с одним и тем же номером (разрядный срез) в различных микроинструкциях может быть использован различным образом.

Будем различать микроинструкции двух типов:

- операционные,
- адресации (выбора).

В данном варианте схемы тип микроинструкции устанавливается разрядом k . При $k=0$ выполняется микроинструкция операционного типа. Все остальные разряды ячейки загружаются в регистр микрокоманды и управляют выполнением микроопераций в ОА. Следующий адрес всегда на единицу больше.

При $k=1$ выполняется микроинструкция адресации. Все разряды микроинструкции могут быть использованы для вычисления следующего адреса. В данном варианте схемы, так же как и в схеме с регулярной адресацией, один из адресов явно записывается в микроинструкцию, другой альтернативный адрес на единицу больше текущего.

В этой схеме используется условная синхронизация, которая позволяет удлинить такт выполнения микрокоманды в ОА на время выполнения микроинструкций адресации.

A	k	Y		
		H	e	S
$n1 \{ m1 \}$	0	0	m1	
$n2 \{ m2 \}$	1	0	m2	
$g1 \ll GO(a; g1, n3) \gg$	2	1	1	2
$n3 \{ m3 \}$	3	0	m3	
$n4 \{ m4 \}$	4	0	m4	
$g2 \ll GO(a; g3, n1) \gg$	5	1	0	0
$g3 \ll GO(b; n5, n3) \gg$	6	1	0	3
$n5 \{ m5 \}$	7	0	m5	
$g4 \ll GO(a; n5, n3) \gg$	8	1	1	7
	9	1	0	3

2.7. Функциональный переход.

Переход на микроподпрограмму с возвратом

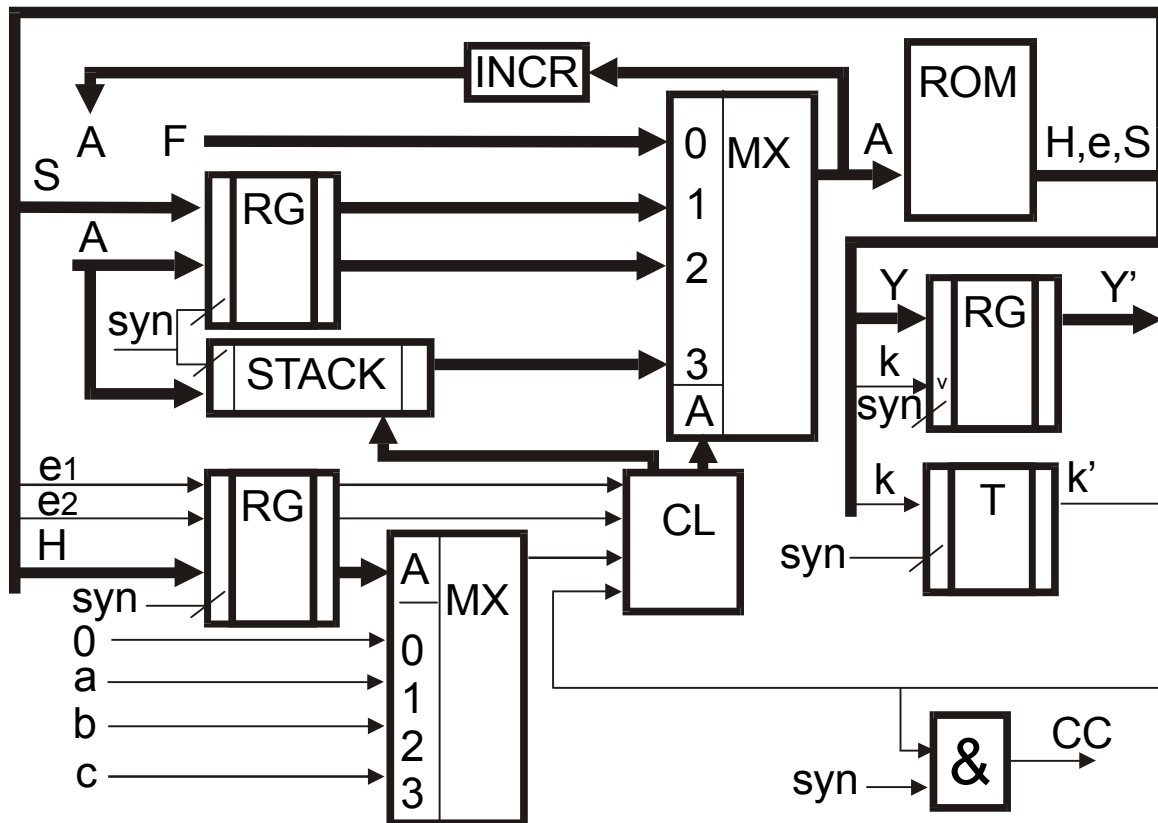


Рис.13. УА с вызовом микроподпрограмм

2.7.1. Функциональный переход

При необходимости выполнения нескольких вычислительных функций, управление которыми представляется в виде независимых микропрограмм, необходимо организовать независимый вызов этих микропрограмм.

Начальные адреса микропрограмм, управляющих вычислениями различных функций, обычно существуют вне управляющей памяти. В УА достаточно предусмотреть механизм коммутации, позволяющий сделать начальный адрес текущим. Это можно осуществить в любой из рассмотренных схем. (К механизму коммутации относятся, кроме аппаратуры, специальные разряды управляющей памяти и специальные микроинструкции.)

2.7.2. Переход к микроподпрограмме с возвратом

При реализации достаточно сложных вычислений удобно

воспользоваться механизмом микроподпрограмм.

Одна и та же микроподпрограмма может быть вызвана из разных точек вызывающих микропрограмм. Поэтому при вызове микроподпрограммы необходимо запомнить адрес, с тем, чтобы восстановить его при возвращении из микроподпрограммы. Чтобы запомнить и восстановить адреса возврата от вложенных вызовов, используется безадресная память - стек (stack).

Принцип (дисциплина) работы стека описывается условием «последним вошел - первым вышел» (Last In - First Out, LIFO). Чтобы описать этот принцип будем считать, что слова, хранящиеся в стеке, перенумерованы с помощью первых натуральных чисел 1,2,...,N. Слово с наибольшим номером называют вершиной стека.

Стек может выполнять следующие действия:

- «чтение» слова с наибольшим номером,
- «выталкивание» (стирание) из памяти слова с наибольшим номером,
- «запись» нового слова с присваиванием ему наибольшего номера.

При вызове микроподпрограммы выполняется операция «записи» в стек адреса возврата. При возвращении из микроподпрограммы выполняется операция «чтения» адреса возврата, затем «выталкивания» его же из стека.

Операция «чтения» без «выталкивания» выполняется при использовании стека для организации циклов.

Разряды с именем (k,e1,e2) определяют тип выполняемой микроинструкции. В связи с тем, что теперь в схеме существует 4 источника адреса для управляющей памяти, возможны 4 типа безусловных переходов, кроме того, возможны различные условные переходы в разных сочетаниях комбинирующие эти источники адреса и входные переменные.

С помощью комбинационной схемы CS разряды микроинструкции (k,e1,e2) преобразуются в сигналы управления стеком и мультиплексором.

3. Управление с предвосхищением

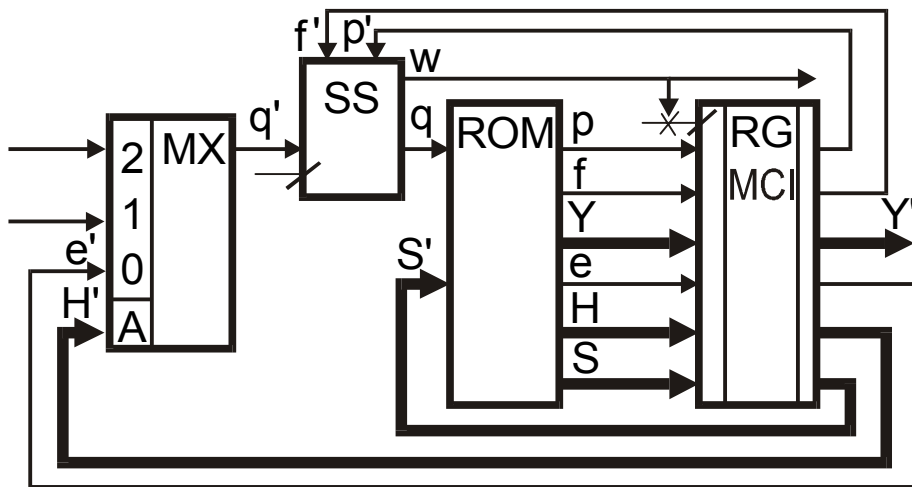


Рис.14. УА с предвосхищением

В конвейерном варианте при выполнении условных переходов по флагам, зависящим без сдвига от сигналов управления, приходится добавлять еще один такт для того, чтобы "увидеть" значение переменной, по которой выполняется ветвление, и выбрать нужную микроинструкцию. Можно построить управляющий автомат, в котором для ускорения выполнения микропрограммы выполняются следующие действия. Предварительно выбирается из ПЗУ одна из двух альтернативных микроинструкций, соответствующая наиболее вероятному значению флага. Это значение должно храниться в той микроинструкции, после которой выполняется ветвление. В конце такта выработанное реальное значение переменной сравнивается с предсказанным, если они совпадают, то выбранная из ПЗУ микроинструкция записывается в выходной регистр, если нет, то предыдущий такт продлевается, т.е. не синхронизируются операционный автомат и RG'MCI.

Микропрограмма будет такой же, как и для последовательного варианта взаимодействия, но в микроинструкцию добавляются два разряда:

$f = \{ 1, \text{если используется предвосхищение}; 0, \text{если нет} \};$

p - наиболее вероятное значение переменной ветвления.

В схему включается автомат SS, который вырабатывает два сигнала:

q - разряд адреса ПЗУ;

w - управляет условной синхронизацией операционного автомата и $RG'MCI$; эта переменная должна зависеть без сдвига от входных переменных автомата SS , т.к. установившееся значение w должно предшествовать рабочему фронту сигнала синхронизации.

Итак, SS - автомат, который может находиться в одном из двух состояний $s0$ и $s1$:

состояние	f'	p', q'	q	w	переход
$s0$	0	x	q'	1	$s0$
$s0$	1	$p' = q'$	p'	1	$s0$
$s0$	1	$p' \neq q'$	p'	0	$s1$
$s1$	x	x	q'	1	$s0$

Аппаратура микропрограммного управляющего автомата несколько увеличилась. При этом основная доля прироста приходится на ПЗУ, увеличенное на два разряда в каждой ячейке.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гилл А. Введение в теорию конечных автоматов. –М.: Наука, 1966 –272с.
2. Гилл А. Линейные последовательностные машины. –М.: Наука, 1974 –288с.
3. Кузин Л.Т. Основы кибернетики в двух томах. Т.2 Основы кибернетических моделей. –М.: Энергия, 1979 –583с.
4. Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов. –М.: Наука, 1985 –320с.
5. Фридман А., Менон П. Теория и проектирование переключаемых схем. – М.: Мир, 1978 –580с.