

Лабораторная работа №3

Разработка цифрового управляющего автомата

1.1 Введение

Большинство цифровых вычислительных устройств состоит, как правило, из трех частей: запоминающей, операционной части и управляющей части.

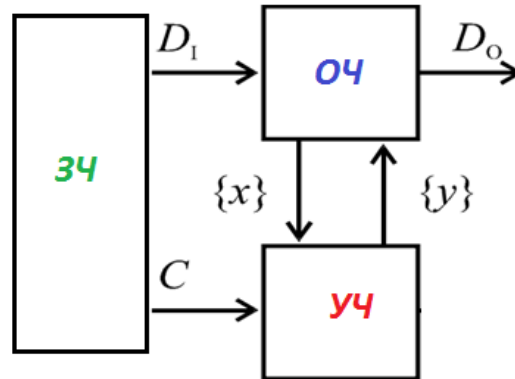


Рисунок 1- Структура вычислительного устройства

Операционная часть выполняет обработку цифровых данных под управлением управляющей части (управляющего автомата).

При разработке управляющих автоматов используются понятия абстрактного и структурного автомата.

Абстрактный автомат (рисунок 2) это математическая модель не учитывающая его внутреннюю структуру и описывает только поведение автомата во времени. Сама структура автомата не известна.



Рисунок 2-Абстрактный автомат

Абстрактный цифровой автомат задается вектором

$$S=(Q,Z,W,\delta,\lambda,q_0),$$

где:

- $Q=\{q_1,\dots,q_m\}$ – множество внутренних состояний абстрактного автомата,
- $Z=\{z_1,\dots,z_k\}$ – множества входных слов
- $W=\{w_1,\dots,w_l\}$ – множества выходных слов
- δ – функция переходов,
- λ – функция выходов,
- q_0 – начальное состояние автомата.

Структурный автомат представляет конкретную реализацию управляющего автомата, в зависимости от его архитектуры и заданного набора элементов используемых для его реализации.

Существуют две архитектуры управляющих автоматов:

- с жёсткой (фиксированной) логикой,
- с хранимой в памяти микропрограммой.

Принцип функционирования автомата с жесткой логикой жёстко задан его схемой. Для внесения даже незначительных изменений в алгоритм необходимо полностью (или почти полностью) пересинтезировать всю схему автомата.

Микропрограммный автомат, работает под управлением микропрограммы, записанной в постоянном запоминающем устройстве (ПЗУ) автомата.

В зависимости выходных слов от значений входных слов управляющие автоматы делятся на два основных типа : автомат Мили и автомат Мура.

1.2 Разработка структурного цифрового управляющего автомата Мили

Автомат Мили имеет структуру, приведенную на рис. 1 в.

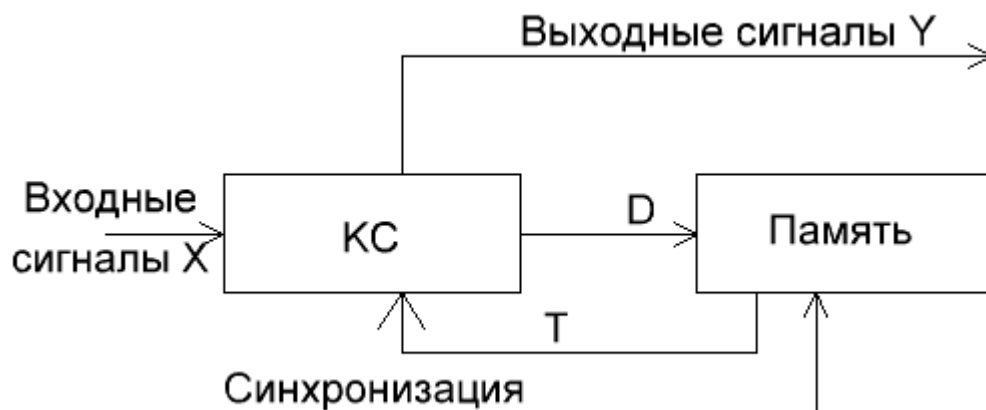


Рисунок 3 - Обобщённая структурная схема УА с жёсткой логикой

где: X – множество входных сигналов автомата,

Y – множество выходных сигналов,

D – сигналы управления памятью,

T – сигналы состояния.

УА состоит из 2-х функциональных блоков:

1. КС – комбинационная схема, формирующая выходные сигналы автомата и сигналы управления памятью.

2. Память автомата – просто набор триггеров (регистр). Кол-во триггеров N определяется количеством Q требуемых состояний автомата. N равно ближайшему целому (в большую сторону) $N = \lceil \log Q \rceil$

. Например, если у нас 5 состояний, то мы должны поставить 3 триггера ($2^3=8>5$), а если 8, то 4 ($2^4=16>8$).

УА с жёсткой логикой бывают 2-х видов – Мили и Мура.

Поведение работы автомата описывается общими формулами $Y=\lambda(X,T)$, $D=\delta(X,T)$, поэтому иногда говорят, что этот автомат генерирует (в смысле изменяет) выходные сигналы при переходах из одного состояния в другое. Здесь подчёркивается тот факт, что Y непосредственно зависит от X .

Цифровой автомат может задаваться как правило двумя способами:

в виде граф - схемы алгоритма (графа);
 виде таблицы переходов.

Рассмотрим синтез цифрового управляющего автомата для операционной части сложения двух двоичных чисел в дополнительном коде. Описание работы операционной части приведено в лабораторной работе 2. Схема алгоритма операции приведена на рисунке 4

Синтез управляющего автомата Мили

Шаг 1. Разметка ГСА

1. Символом Q0 помечаем вход вершины, следующий за начальной и вход конечной вершины
2. Входы всех вершин, следующих за операторными, помечаем символами состояний Q1-Q5.
3. Вход вершины отмечаются одним символом.

Схема алгоритма приведена на рисунке:

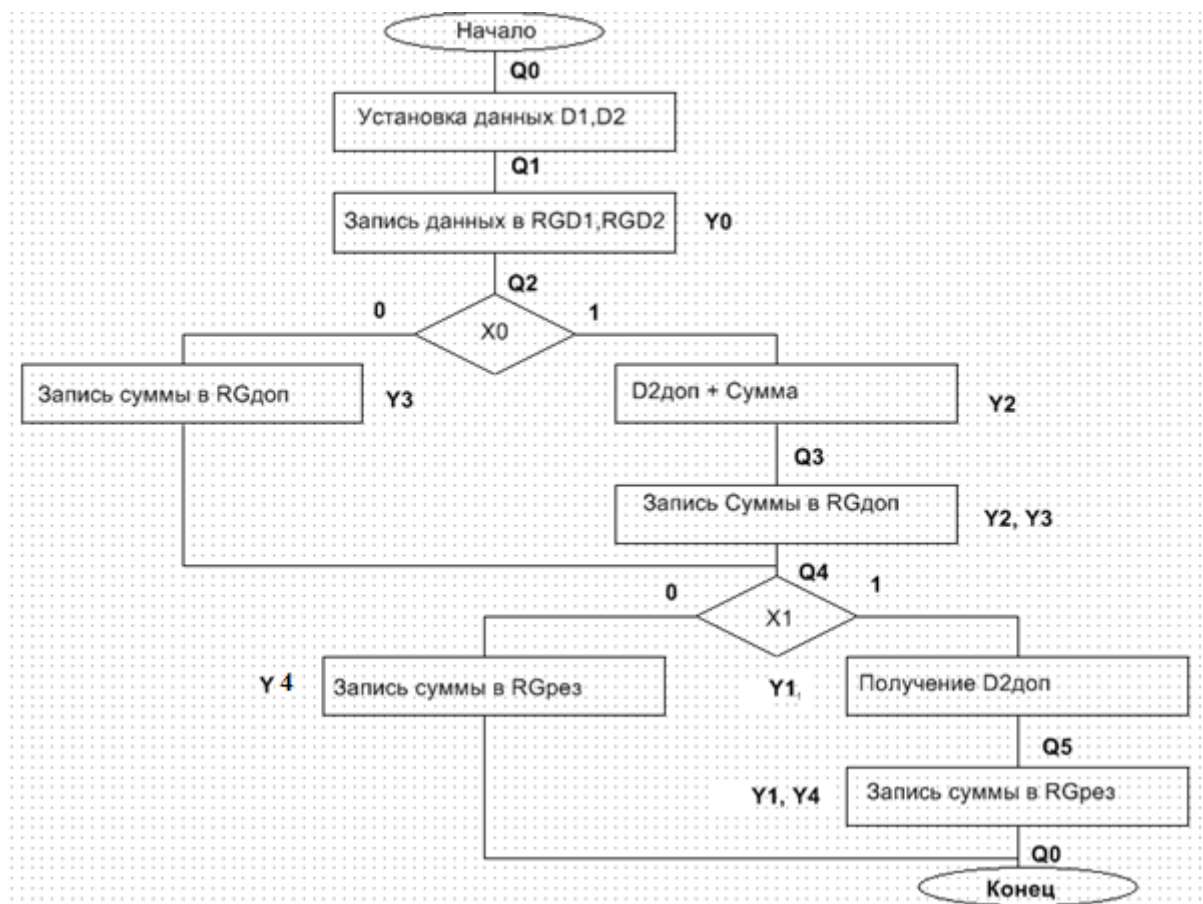


Рисунок 4 - Размеченная ГСА автомата Мили

Как видим наш автомат должен иметь $Q = 6$ состояний (Q0-Q5). Для его построения необходимо $N = \lceil \log_2 6 \rceil = 3$ элемента памяти. В качестве элемента памяти будем использовать три D триггера, входящие в четырехразрядный параллельный регистр.

Шаг 2 Кодирование состояний

Зададим таблицу кодировки состояний

Номер состояния	Код состояния		
	T1	T2	T3
Q0	0	0	0
Q1	1	0	0
Q2	0	1	0
Q3	1	1	0
Q4	0	0	1
Q5	1	0	1

Шаг 3 Построение полной таблицы переходов автомата Мили

Исх. сост.	Код исходного состояния			Условия перехода		След. Сост.	Код следующего состояния			Выходные сигналы					
	T1	T2	T3	X0	X1		D1	D2	D3	Y0	Y1	Y2	Y3	Y4	
Q0	0	0	0	*	*	Q1	1	0	0	0	0	0	0	0	
Q1	1	0	0	*	*	Q2	0	1	0	1	0	0	0	0	
Q2	0	1	0	0	*	Q4	0	0	1	0	0	0	1	0	
Q2	0	1	0	1	*	Q3	1	1	0	0	0	1	0	0	
Q3	1	1	0	1	*	Q4	0	0	1	0	0	1	1	0	
Q4	0	0	1	*	0	Q0	0	0	0	0	0	0	0	1	
Q4	0	0	1	*	1	Q5	1	0	1	0	1	0	0	0	
Q5	1	0	1	*	1	Q0	0	0	0	0	1	0	0	1	

Состояние Q0 необходимо для начальной установки данных D1, D2.

В состоянии Q1 данные записываются в входные регистры.

Шаг 4 Построение логических выражений функций выходных сигналов и сигналов управления элементами памяти.

Запишем выражения (СДНФ) для логических функций Y0, Y1, Y2, Y3, Y4 и D1, D2, D3.

$$Y0 = T1 \cdot \overline{T2} \cdot \overline{T3}$$

$$Y1 = \overline{T1} \cdot \overline{T2} \cdot T3 \cdot x1 + T1 \cdot \overline{T2} \cdot T3 \cdot x1$$

$$Y2 = \overline{T1} \cdot T2 \cdot \overline{T3} \cdot x0 + T1 \cdot T2 \cdot \overline{T3} \cdot x0$$

$$Y3 = T1 \cdot T2 \cdot \overline{T3} \cdot \overline{x0} + T1 \cdot T2 \cdot \overline{T3} \cdot X0$$

$$Y4 = \overline{T1} \cdot \overline{T2} \cdot T3 \cdot \overline{x1} + T1 \cdot \overline{T2} \cdot T3 \cdot x1$$

$$D1 = \overline{T1} \cdot \overline{T2} \cdot \overline{T3} + \overline{T1} \cdot T2 \cdot \overline{T3} \cdot x0 + \overline{T1} \cdot \overline{T2} \cdot T3 \cdot x1$$

$$D2 = T1 \cdot \overline{T2} \cdot \overline{T3} + \overline{T1} \cdot T2 \cdot \overline{T3} \cdot x0$$

$$D3 = \overline{T1} \cdot T2 \cdot \overline{T3} \cdot \overline{x0} + T1 \cdot T2 \cdot \overline{T3} \cdot x0 + \overline{T1} \cdot \overline{T2} \cdot T3 \cdot x1$$

Для лучшей минимизации вместо звездочек можно записать конъюнкции с их возможными значениями.

$$D1 = \overline{T1} \cdot \overline{T2} \cdot \overline{T3} \cdot \overline{x0} \cdot \overline{x1} + \overline{T1} \cdot \overline{T2} \cdot \overline{T3} \cdot x0 \cdot \overline{x1} + \overline{T1} \cdot \overline{T2} \cdot \overline{T3} \cdot \overline{x0} \cdot x1 + \overline{T1} \cdot \overline{T2} \cdot \overline{T3} \cdot x0 \cdot x1 + \\ \overline{T1} \cdot T2 \cdot \overline{T3} \cdot \overline{x0} \cdot \overline{x1} + \overline{T1} \cdot T2 \cdot \overline{T3} \cdot x0 \cdot \overline{x1} + \overline{T1} \cdot T2 \cdot \overline{T3} \cdot \overline{x0} \cdot x1 + \overline{T1} \cdot T2 \cdot \overline{T3} \cdot x0 \cdot x1$$

$$D2 = T1 \cdot \overline{T2} \cdot \overline{T3} \cdot \overline{x0} \cdot \overline{x1} + T1 \cdot \overline{T2} \cdot \overline{T3} \cdot x0 \cdot \overline{x1} + T1 \cdot \overline{T2} \cdot \overline{T3} \cdot \overline{x0} \cdot x1 + T1 \cdot \overline{T2} \cdot \overline{T3} \cdot x0 \cdot x1 + \\ T1 \cdot T2 \cdot \overline{T3} \cdot \overline{x0} \cdot \overline{x1} + T1 \cdot T2 \cdot \overline{T3} \cdot x0 \cdot \overline{x1} + T1 \cdot T2 \cdot \overline{T3} \cdot \overline{x0} \cdot x1 + T1 \cdot T2 \cdot \overline{T3} \cdot x0 \cdot x1$$

$$D3 = \overline{T1} \cdot T2 \cdot \overline{T3} \cdot \overline{x0} \cdot \overline{x1} + \overline{T1} \cdot T2 \cdot \overline{T3} \cdot x0 \cdot \overline{x1} + T1 \cdot T2 \cdot \overline{T3} \cdot \overline{x0} \cdot \overline{x1} + T1 \cdot T2 \cdot \overline{T3} \cdot x0 \cdot \overline{x1} + \\ T1 \cdot T2 \cdot \overline{T3} \cdot \overline{x0} \cdot x1 + T1 \cdot T2 \cdot \overline{T3} \cdot x0 \cdot x1$$

Шаг 5. Минимизация логических функций.

На следующем шаге необходимо провести минимизацию логических функций любым подходящим методом. Будем использовать карты Карно. Приведем пример минимизации функции Y1. Строим карту Карно.

	T2T3			
X1T1	00	01	11	10
00				
01				
11		1		
10		1		

После склеивания получим выражение для Y1

$$Y1 = X1 \overline{T2} T3$$

Проведя аналогичные действия над Y2, D3 получим соответственно

$$Y2 = \overline{T1} \cdot T2 \cdot \overline{T3} \cdot x0 + T1 \cdot T2 \cdot \overline{T3} \cdot x0 = T2 \cdot \overline{T3} \cdot x0 (\overline{T1} + T1) = T2 \cdot \overline{T3} \cdot x0$$

Значения Y3, Y4 остаются без изменения.

$$Y3 = \overline{X0} \overline{T1} T2 \overline{T3} + X0 T1 T2 \overline{T3}$$

$$Y4 = \overline{X1} \overline{T1} \overline{T2} T3 + X1 T1 \overline{T2} T3$$

Произвести минимизацию значений D1, D2, D3

Шаг 6. Построение принципиальной схемы управляющего автомата Мили.

Принципиальная схема управляющего автомата Мили строится на основании полученных логических выражений для Y и D.

Каждое переключение автомата из одного состояния в другое происходит по кнопке "ТАКТ".

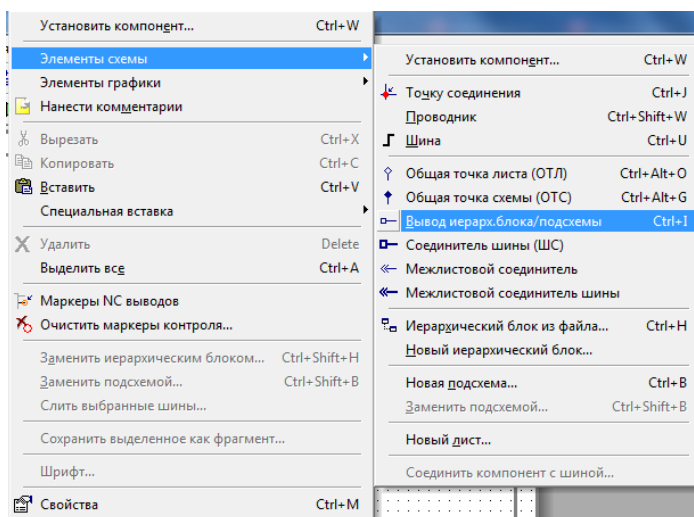
Разрядность данных желательно выбирать не более 4 с учетом знака. Если под знак необходимо использовать два разряда (модифицированный код), то можно выбирать соединене основных элементов, управляемых параллельно одими сигналами. Например,

для хранения 5 - разрядных чисел можно использовать два 4-х разрядных регистра управляемых одними и теми же сигналами. Тоже относится к мультиплексорам и сумматорам).

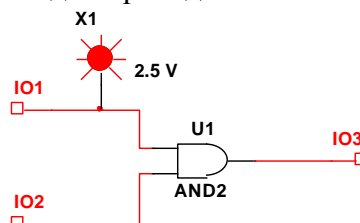
Сигналы управления (Y_n) с управляющего автомата необходимо подать на разработанную в лабораторной работе « операционную часть».

Ссовмещенное устройство приведено на рисунке 5 (в среде Multisim и 5а в среде Proteus).

Операционная часть представлена в виде подсхемы (прямоугольник с входами и выходами). Для создания подсхемы операционной части в среде Multisim необходимо ко всем внешним входам и выходам предварительно подсоединить выводы подсхемы (правой кнопкой мыши на рабочем поле\элемент схемы\выводы подсхемы)



Пример расположения выводов приведен ниже



Далее необходимо выделить схему (нажать правую кнопку мыши и выбрать пункт "Заменить подсхемой" , присвоить имя подсхеме (операционная часть). После этого наша схема заменяется прямоугольником с входами и выходами к которым подводятся сигналы с управляющей части.

Для просмотра содержимого подсхемы надо кликнуть по ней два раза мышью и выбрать пункт «редактировать» после чего она появится на отдельном листе схемы.

Пример создания подсхемы в среде Proteus приведен в соответствующем файле лабораторной работы 3.

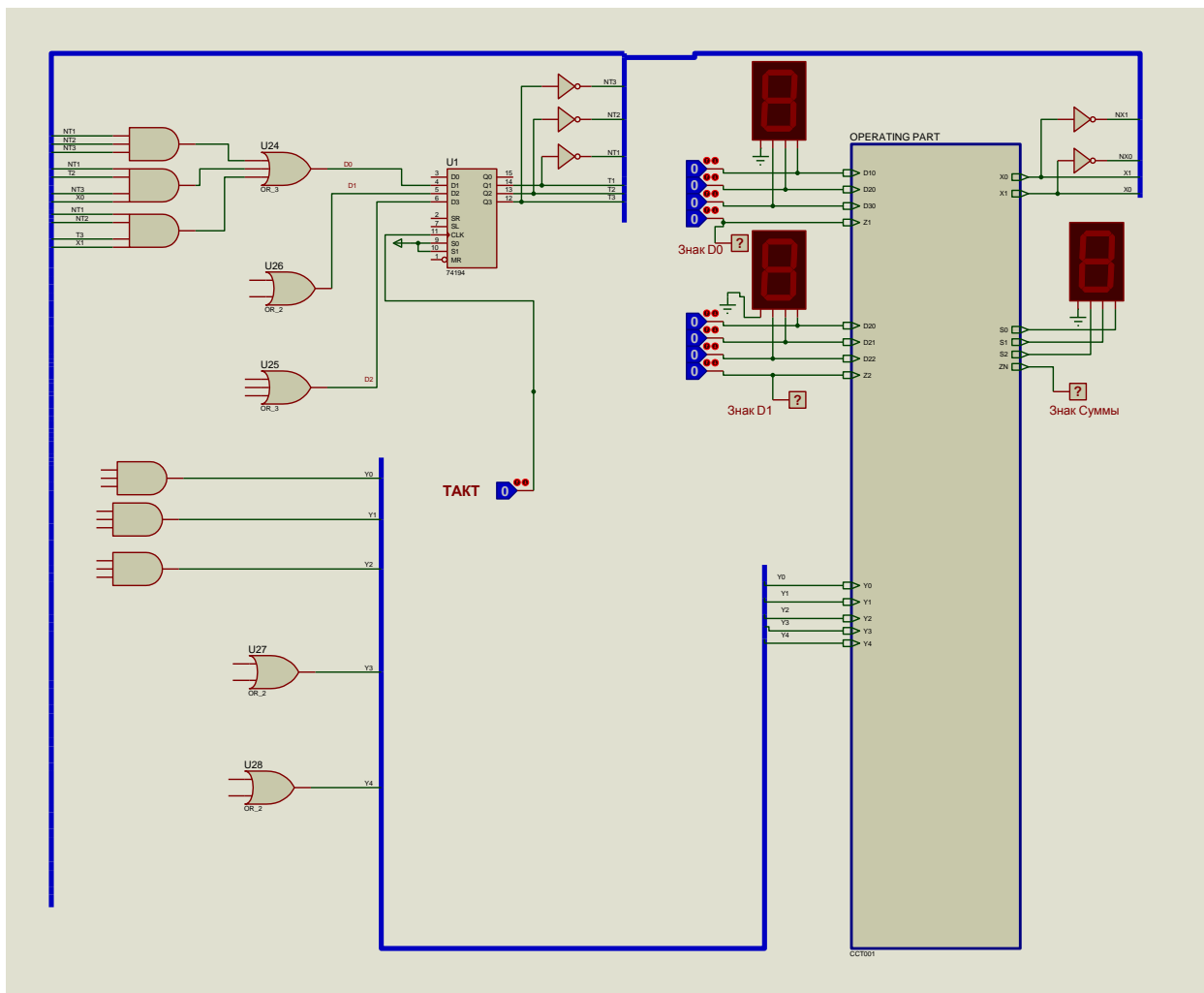


Рисунок 5а - Фрагмент схемы управляющего автомата Мили

1.3 Синтез управляющего автомата Мура

Автомат Мура отличается от Мили тем, что он описывается формулами $Y=f1(T)$, $D=f2(X,T)$. Т.е. его выходные сигналы зависят только от состояния триггеров. Поэтому его КС фактически распадается на 2 независимые КС – рис. 6.

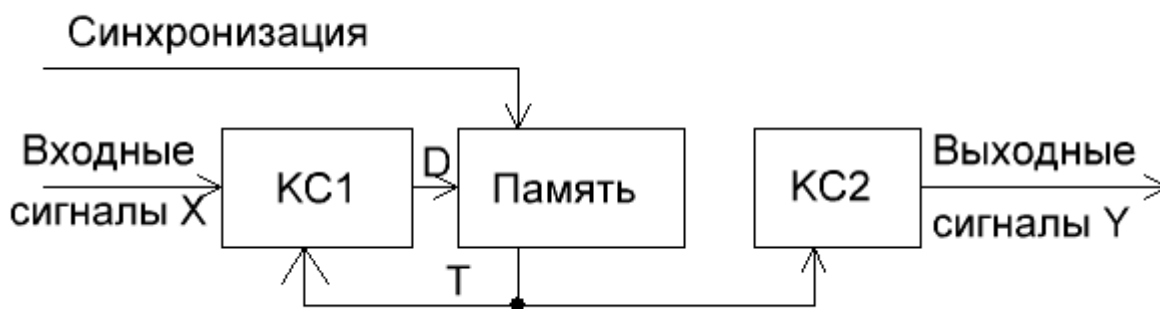


Рисунок 6 - Структура автомата Мура

КС1 реализует функцию $D=f_2(X,T)$, а КС2 - $Y=f_1(T)$.
Для каждой комбинационной схемы строится своя таблица состояний.

Шаг 1. Разметка ГСА

- 1. Символом Q0 отмчается начальная и кнечная вершина ГСА.
 - 2. Символами Q1-Q5 отмчаются все **операторные вершины** ГСА.
 - 3. Каждая опраторная вершина помечается одним символом.
- Рамеченная ГСА автомата Мура приведена на рисунке 7.

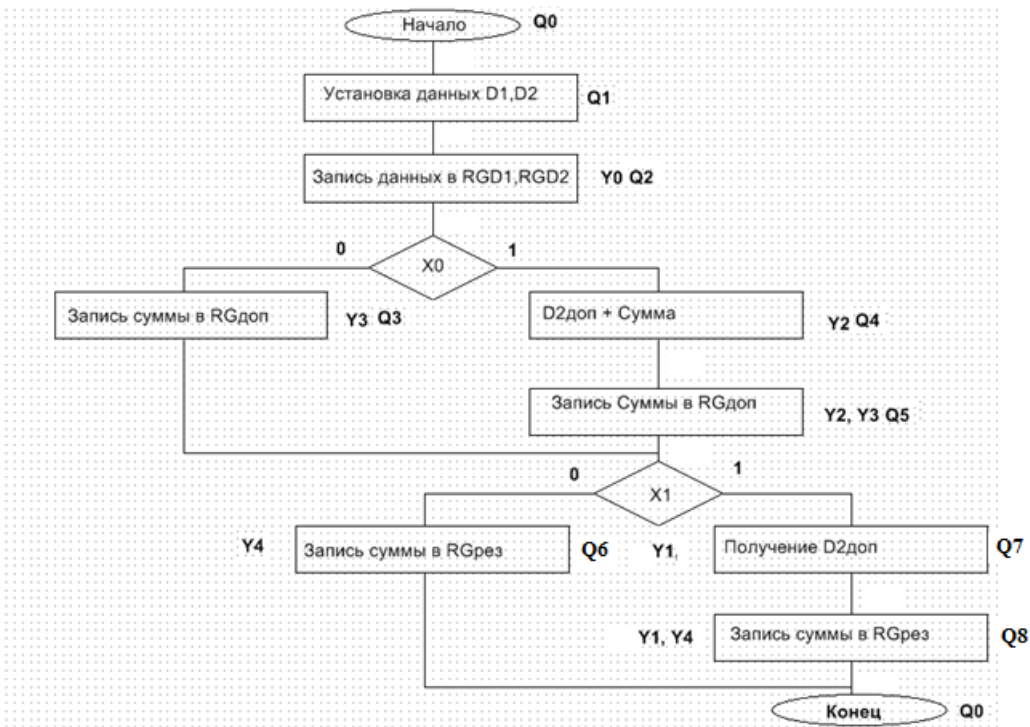


Рисунок 7 Размеченная ГСА автоиата Мура

Шаг 2. Кодирование состояний и таблица выходов для построения КС2

Для КС2

Номер состояния	Код состояния				Значения выходных сигналов Y
	T1	T2	T3	T4	
	T1	T2	T3	T4	Y0-Y4=0
Q0	0	0	0	0	Y0-Y4=0
Q1	1	0	0	0	Y0-Y4=0
Q2	0	1	0	0	Y1-Y4=0 Y0=1
Q3	1	1	0	0	Y0,Y1,Y2,Y4=0 Y3=1
Q4	0	0	1	0	Y0,Y1,Y3,Y4=0 Y2=1
Q5	1	0	1	0	Y0,Y1,Y4=0 Y2,Y3=1
Q6	0	1	1	0	Y0,Y1,Y2,Y3=0 Y4=1
Q7	1	1	1	0	Y0,Y2,Y3,Y4=0 Y1=1
Q8	0	0	0	1	Y0,Y2,Y3=0 Y1,Y4=1

Шаг 3Построение таблицы переходов автомата Мура для КС1

Исх. сост.	Код исходного состояния				Условия перехода		След. Сост.	Код следующего состояния			
	T1	T2	T3	T4	X0	X1		D1	D2	D3	D4
Q0	0	0	0	0	*	*	Q1	1	0	0	0
Q1	1	0	0	0	*	*	Q2	0	1	0	0
Q2	0	1	0	0	1	*	Q4	0	0	1	0
Q2	0	1	0	0	0	*	Q3	1	1	0	0
Q3	1	1	0	0	*	0	Q6	0	1	1	0
Q3	1	1	0	0	*	1	Q7	1	1	1	0
Q4	0	0	1	0	1	*	Q5	1	0	1	0
Q5	1	0	1	0	*	0	Q6	1	0	1	0
Q5	1	0	1	0	*	1	Q7	1	1	1	0
Q6	0	1	1	0	*	*	Q0	0	0	0	0
Q7	1	1	1	0	*	*	Q8	0	0	0	1
Q8	0	0	0	1	*	*	Q0	0	0	0	0

Шаг 4. По таблицам записываем значения функций Y0-Y5 и D1-D4 аналогично автомату Мили.

Например,

$$Y2 = \overline{T1} \overline{T2} T3 \overline{T4} + T1 \overline{T2} T3 \overline{T4}$$

$$D4 = T1 T2 T3 \overline{T4} \overline{X0} \overline{X1} + T1 T2 T3 \overline{T4} X0 \overline{X1} + T1 T2 T3 \overline{T4} \overline{X0} X1 + T1 T2 T3 \overline{T4} X0 X1$$

Шаг 5 Минимизируем полученные значения.

Шаг 6 Строим принципиальную схему в среде Multisim

Шаг 7 Отлаживаем её.

1.4 Синтез управляющего автомата на основе микропрограммного управления.

Основное достоинство рассмотренных УА с жёсткой логикой — их высокое быстродействие, определяемое быстродействием используемой элементной базы.

Однако есть и большие недостатки:

1. При необходимости внесения любых, даже небольших изменений алгоритма работы схему автомата надо полностью пересинтезировать.

2. При большом числе входных и выходных сигналов схема автомата сильно разрастается, а синтез становится сложным. Так, например, карты Карно уже при 5 аргументах становятся трудночитаемыми и труднопонимаемыми. Неизбежным итогом этого может стать неполная минимизация и, как следствие, излишне сложная и избыточная схема полученного автомата. На работоспособность схемы это, правда, не повлияет.

Второй недостаток особенно ярко проявляется при разработке различных вычислительных структур, где есть много операционных узлов, для которых требуется очень много выходных сигналов и много состояний управляющего автомата.

В таких случаях используют принципиально другие УА — УА с микропрограммным управлением.

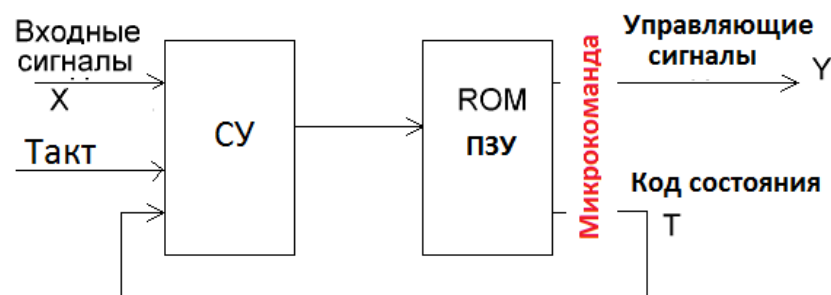


Рис. 8. Структура УА с микропрограммным управлением

Основа такого управляющего аппарата – ROM – ПЗУ. Каждая ячейка ПЗУ хранит микрокоманду (МК) – набор выходных сигналов Y для каждого состояния автомата и набор управляющих сигналов T для своего сугубо внутреннего устройства управления УУ.

В плане генерации выходных сигналов все микропрограммные автоматы идентичны автомату Мура – Y зависят только от состояния памяти автомата.

УА с микропрограммным управлением бывают 2-х типов – с естественной адресацией микрокоманд и с принудительной. В каждом случае структура УУ разная.

Пример МП автомата с естественной адресацией приведен на – рис 9.



Рис 9. Структура МП УА с естественной адресацией.

Микропрограмма записана в микропрограммное постоянное запоминающее устройство МПЗУ. Каждая микрокоманда состоит из четырех полей $V: A: N_x: Y$. Существует два типа микрокоманд, различающихся старшим битом V :

$V=1$ – ОМК – операционная микрокоманда, содержащая в поле Y значения бит соответствующие сигналам $Y_1 - Y_n$.

$V=0$ – УМК – управляющая микрокоманда. По этой команде происходит переход на микрокоманду адрес которой хранится в поле A управляющей команды.

Поле A – адрес перехода, подаётся на входы предзагрузки данных D регистра счётчика адреса микрокоманд **РАМК** и в зависимости от значения анализируемой переменной X_i , поступающей с мультиплексора **MS**, формирует адрес перехода.

N_x – закодированный номер проверяемой входной переменной. Это значение подается на адресные входы мультиплексора **MS** и разрешает прохождение на его выход значения соответствующей переменной X_i . Безусловный переход реализуется путем фиксирования лог. 0 на первом входе данных **MS**. Т.е. автомат всегда выполняет переход, если $X_i=0$.

$Y_1 - Y_n$ – управляющие сигналы формируемые управляющим автоматом.

Счётчик-РАМК при $V=0$ выполняет загрузку с входов **D** адреса следующей команды (для команды перехода). При $V=1$ выполняется счёт (инкремент – увеличение значения на 1) по тактовым импульсам на входе +1, по которым происходит выполнение микрокоманд по последовательным адресам.

РМК (регистр микрокоманды) осуществляет промежуточное хранение микрокоманды на время её выполнения.

Функционирование автомата состоит в следующем.

Если автомат выполняет ОМК, то 1 ($V=1$) из старшего бита МК блокирует мультиплексор входных сигналов **MS** элементом ИЛИ и запрещает счётчику-РАМК выполнять предзагрузку адреса перехода. На выходе ПЗУ формируются управляющие сигналы Y_i . По приходу очередного тактового импульса счётчик увеличится на 1, последовательно выбирая следующую МК.

Если выполняется УМК ($V=0$), то выходные значения Y_i сохраняют предыдущее значение, а элемент ИЛИ разрешает прохождение сигнала данных с мультиплексора. Далее, если $X_i=0$, то прохождение тактовых импульсов на счётчик запрещается, но разрешается предзагрузка значением поля адреса **A** микрокоманды и выполняется переход. Если $X_i=1$, то предзагрузка не происходит, а разрешен проход тактового импульса на счетный вход +1, в результате перехода по адресу нет, и счетчик последовательно формирует по следующий адрес микрокоманды.

Рассмотрим разработку МП для такого автомата на примере.

Шаг1 Построение схемы-алгоритма функционирования микропрограммного автомата. Для этого будем использовать схему алгоритма автомата Мура.

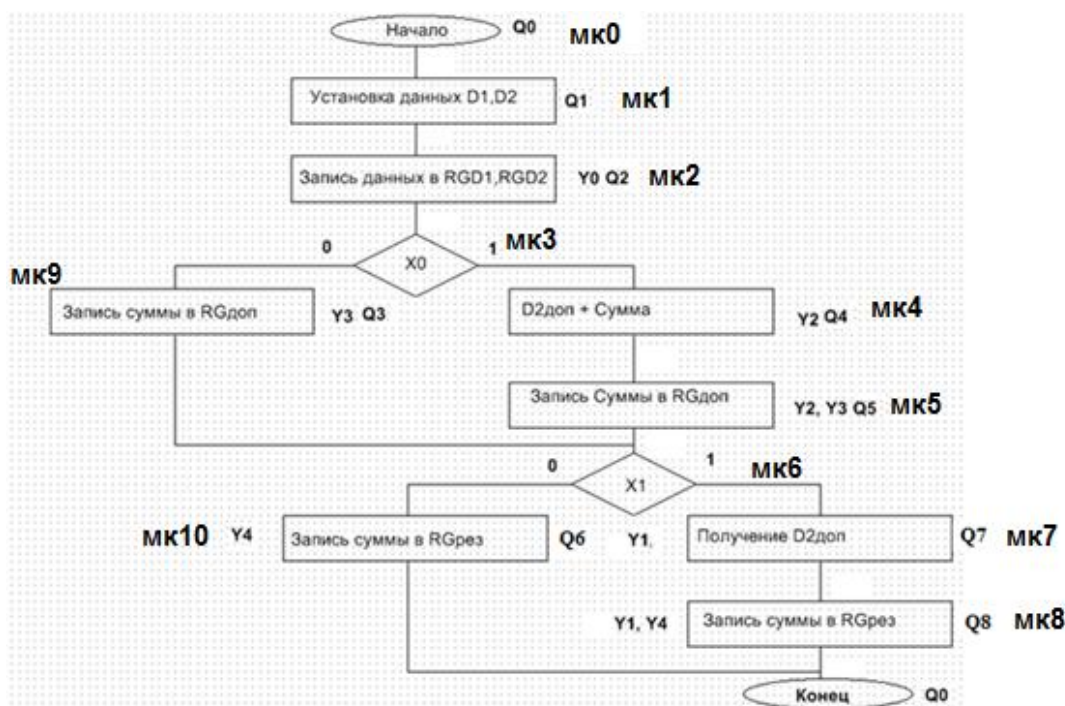


Рис. 10 Схема-алгоритма работы микропрограммного автомата

Как видно из схемы в каждой вершине вырабатывается микрокоманда (управляющая или операционная).

Шаг2. Кодировка входных сигналов автомата x_0, x_1 (знак слагаемого и знак результата) и постоянный лог. 0 (для безусловных переходов) на нулевом входе мультиплексора MS1:

Входы данных мультиплексора	Адресные входы мультиплексора
Вход \square	Поле N_x микрокоманды \square
0 (лог. 0) Безусловный переход	00 \square
1 (x_0) \square	01 \square
2 (x_1) \square	10 \square

Шаг 3. Разработка микропрограммы.

№ МК	Тек. сост.	Адрес на входе ПЗУ				Адрес следующей МК				Поле N_x		V	hex	Выходные сигналы				
		A3	A2	A1	A0	A3	A2	A1	A0	NX1	NX0			Y4	Y3	Y2	Y1	Y0
Мк0	Q0	0	0	0	0	0	0	0	0	1	1	1	F	0	0	0	0	0
Мк1	Q1	0	0	0	1	0	0	0	0	1	1	1	17	0	0	0	0	0
Мк2	Q2	0	0	1	0	0	0	0	0	1	1	1	1F	0	0	0	0	1
Мк3	Q2	0	0	1	1	0	0	0	0	0	1	0	52	0	0	0	0	1
Мк4	Q4	0	1	0	0	0	0	0	0	1	1	1	2F	0	0	1	0	0
Мк5	Q5	0	1	0	1	0	0	0	0	1	1	1	37	0	1	1	0	0
Мк6	Q5	0	1	1	0	0	0	0	0	1	0	0	64	0	1	1	0	0
Мк7	Q7	0	1	1	1	0	0	0	0	1	1	1	47	0	0	0	1	0
Мк8	Q8	1	0	0	0	0	0	0	1	1	1	1	4F	1	0	0	1	0
Бп	Q8	1	0	0	1	0	0	0	1	0	0	0	08	1	0	0	1	0
Мк9	Q3	1	0	1	0	0	0	0	1	1	1	1	5F	0	1	0	0	0
Бп	Q3	1	0	1	1	0	0	0	1	0	0	0	30	0	1	0	0	0
Мк10	Q6	1	1	0	0	0	0	0	1	1	1	1	6F	1	0	0	0	0
Бп	Q6	1	1	0	1	0	0	0	1	0	0	0	08	1	0	0	0	0

На начальном этапе счетчик команд формирует нулевой адрес, который поступает на адресные входы ПЗУ. По этому считывается Мк0. Это ОМК, поэтому $V=1$. Далее по следующему тактовому происходит увеличение содержимого счетчика на 1. Из ПЗУ считывается Мк1. По следующему такту считывается Мк2. По третьему такту считывается Мк3. В Мк3 происходит анализ переменной X_0 (знак слагаемого), поэтому данная команда, является командой условия (перехода) и $V=0$. Если значение $X_0=0$, то по сигналу $V=0$ происходит запись адреса перехода(1001) в регистр-счетчик адреса микрокоманд и формируется адрес перехода на (1001) и из ПЗУ считывается Мк9. Желтым цветом выделены команды безусловного перехода.

Микропрограмма занимает 14 ячеек памяти. Разрядность выходных данных ПЗУ определяется: $V+N_x+A+Y=1+2+4+5=12$. Так как разрядность данных ячейки ПЗУ составляет 8 бит, то нам необходимо использовать два Пзу. При этом биты, относящиеся к полям $V+N_x+A$, записываются в одно ПЗУ, а биты, отвечающие за Y записываются в другое ПЗУ в формате hex файла. На следующем рисунке приведены фрагменты начала hex-файлов для обоих ПЗУ.

: 0100000000FF0	: 0100000000FF
: 0100010017E7	: 0100010000FE
: 010002001FDE	: 0100020001FC
: 0100030052AA	: 0100030001FB
: 010004002FCC	: 0100040004F7

Шаг4. Разработка принципиальной схемы микропрограммного автомата .
Принципиальная схема автомата приведена на рисунке 11. , 12,13.

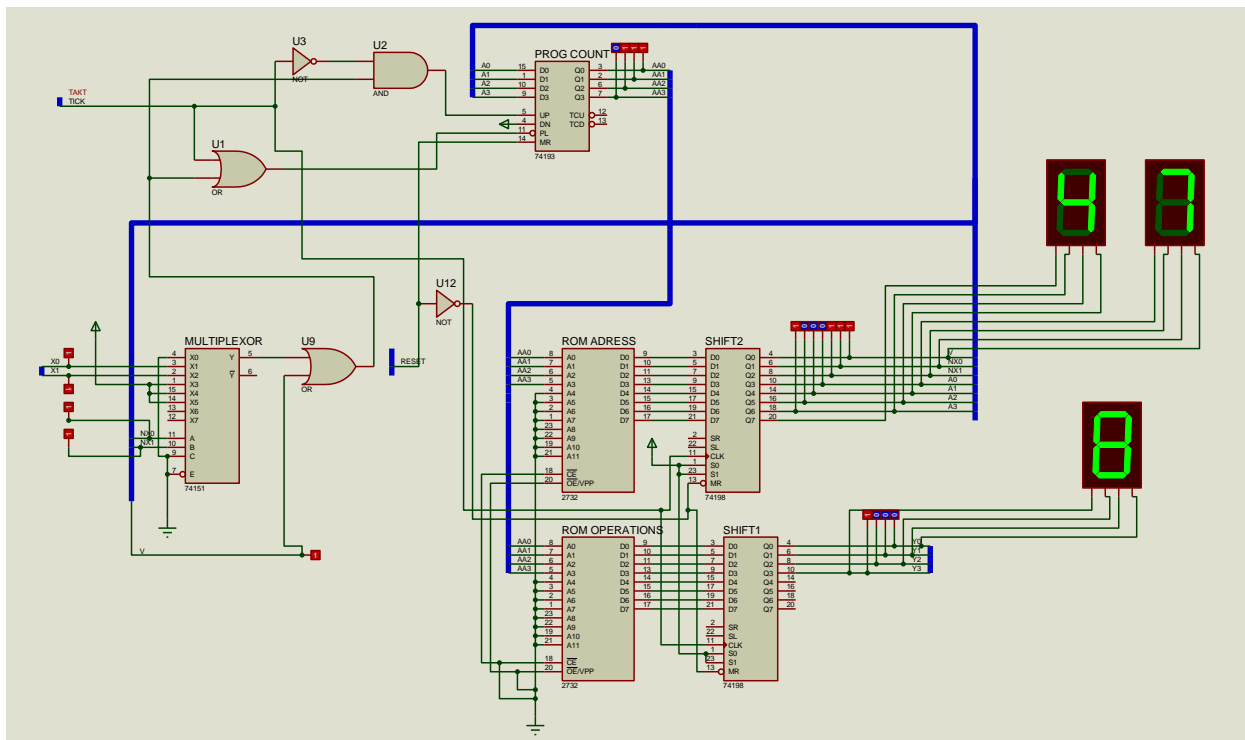


Рисунок 11. Базовая принципиальная схема микропрограммного автомата

На рисунке 12 приведена схема автомата при количестве микрокоманд более 16. В данном случае используются два счетчика, соединенных последовательно.

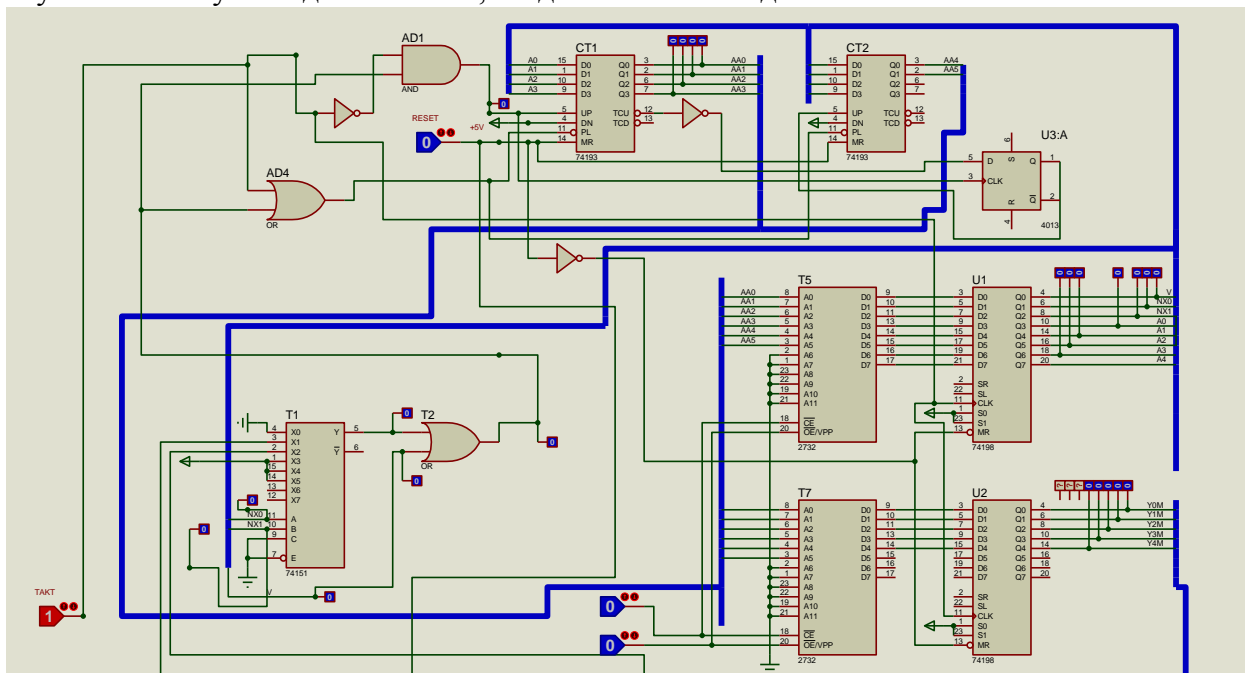


Рисунок 12 Базовая принципиальная схема микропрограммного автомата с количеством микрокоманд более 16

В качестве элемента ПЗУ используется ПЗУ 8К *8 типа 27C128 компонентное значение 2732. Так как количество считываемых бит данных составляет 12, то используется два ПЗУ, управляемые параллельно по одним и тем же адресам. На неиспользуемые адресные входы подается уровень логического нуля.

Для чтения информации из ПЗУ на управляющие входы необходимо подать последовательно уровень логического нуля на входы CE, OE.

Для прошивки ПЗУ надо кликнуть по нему левой кнопкой мышки два раза и в открывшемся окне выбрать файл прошивки в формате hex (рис. 13).

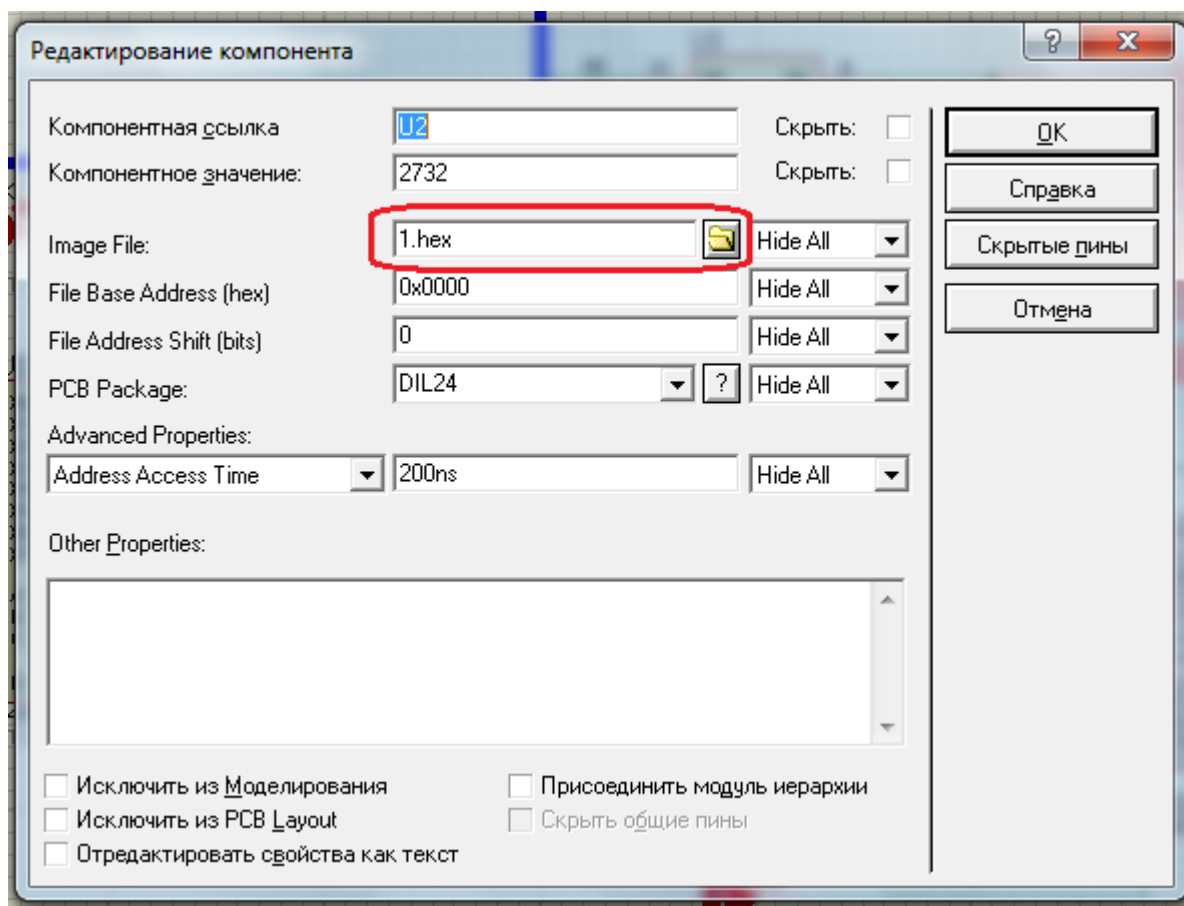


Рис 13 Запись прошивки в ПЗУ

В качестве счетчика микропрограмм можно использовать счетчик с параллельной записью информации 74193. Так как всего используется 12 микрокоманд, то для их адресации достаточно 4-х разрядного счетчика. При большем количестве микрокоманд счетчики соединяются последовательно (выход переноса первого счетчика соединяется со входом UP следующего счетчика). Перед началом работы счетчик обнуляется подачей единицы на вход сброса MR. Тактирование микропрограммного автомата осуществляется с помощью кнопки "ТАКТ".

В качестве мультиплексора выбрана микросхема 74151. Сигналами на адресных входах A, B входные значения X0, X1 последовательно поступают на выход мультиплексора.

Работа автомата :

- 1) обнулить счетчик;
- 2) подать сигналы X0, X1;
- 3) Кнопкой "ТАКТ" подать серию импульсов для синхронизации работы автомата.

Для прошивки ПЗУ, составленную программу необходимо представить в hex-формате. Правила составления hex-файла приведено в дополнительном файле.

Отчет должен содержать:

3 Граф - схему алгоритма

4 Таблицы состояний для управляющих автоматов (Мили/Мура);

5 СДНФ/СКНФ для логических выражений (Мили/Мура);

6 Минимизацию логических выражений (Мили/Мура);

7 Микропрограмму(для микропрограммного автомата);

8 Hex-файл (для микропрограммного автомата);

7 Принципиальную схему управляющего автомата.

8. Продемонстрировать работу полного (совмещенного)устройства в среде Multisim или Proteus.

9. Предусмотреть счетчик количества тактовых импульсов.

Отчет оформить в бумажном варианте

Варианты заданий:

№ вариант а	Разрядность данных	Зна к D1	Зна к D2	Знак результата	Сложение доп.код	Сложен ие обратны й код	Тип управляющего автомата
1	3+1знак=4	+	-/+	-/+		+	Мура
2	3+1знак=4	+	-/+	-/+		+	Мили
3	3+1знак=4	+	-/+	-/+		+	М П
4	3+1знак=4	+	-/+	-/+	+		Мура
5	3+1знак=4	+	-/+	-/+	+		Мили
6	3+1знак=4	+	-/+	-/+	+		М П
7	4	Умножение с мл.разрядов со сдвигом суммы вправо					Мили
8	4	Умножение с мл.разрядов со сдвигом суммы вправо					МП
9	4	Умножение с мл.разрядов со сдвигом суммы вправо					Мура
10	4	Умножение с ст. разрядов со сдвигом суммы влево					МП
11	4	Умножение с ст. разрядов со сдвигом суммы влево					Мили
12	4	Умножение с ст. разрядов со сдвигом суммы влево					Мура
13	4	Умножение с мл. разрядов со сдвигом множимого влево					Мура
14	4	Умножение с мл. разрядов со сдвигом множимого влево					МП
15	4	Умножение с мл. разрядов со сдвигом множимого влево					Мили

Контрольные вопросы:

1. Что такое абстрактный и структурный управляющий автомат.
2. Приведите структурную схему управляющего автомата Мили.
3. Приведите структурную схему управляющего автомата Мура.
4. Приведите структурную схему микропрограммного автомата.
5. Приведите сравнительный анализ управляющих автоматов.
6. Какой формат у hex-файла.