**Лабораторная работа №2**

**Разработка простейшего цифрового вычислительного устройства на основе управляющего автомата**

**1.1 Введение**

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

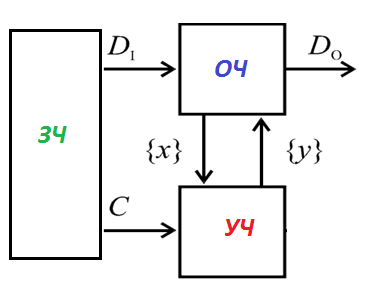


Рисунок 1а Структура вычислительного устройства

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 1 б) это математическая модель не учитывающая его внутреннюю структуру и описывает только поведение автомата во времени. Сама структура автомат не известна.



Абстрактный цифровой автомат задается вектором

S=(Q,Z,W,δ,λ,q0),

где:

* + Q={q1,…,qm} – множество внутренних состояний абстрактного автомата,
  + Z=[z1,…,zk} – множества входных слов
  + W={w1,…,wl} – множества выходных слов
  + δ – функция переходов,
  + – функция выходов,
  + q0 – 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 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 цифрового устройства на основе управляющего автомата Мили**

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

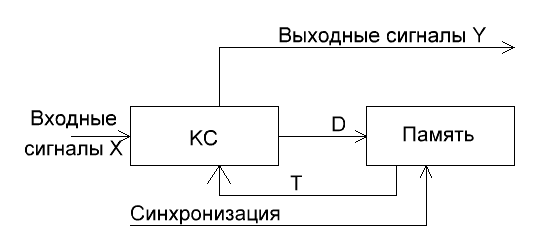


Рис. 1. Обобщённая структурная схема УА с жёсткой логикой

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 k требуемых состояний автомата. k равно ближайшему целому (в большую сторону) числу из значений выражения 2n. Т.е. n=]log2k[. Например, если у нас 5 состояний, то мы должны поставить 3 триггера (23=8>5), а если 8, то 4 (24=16>8).

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

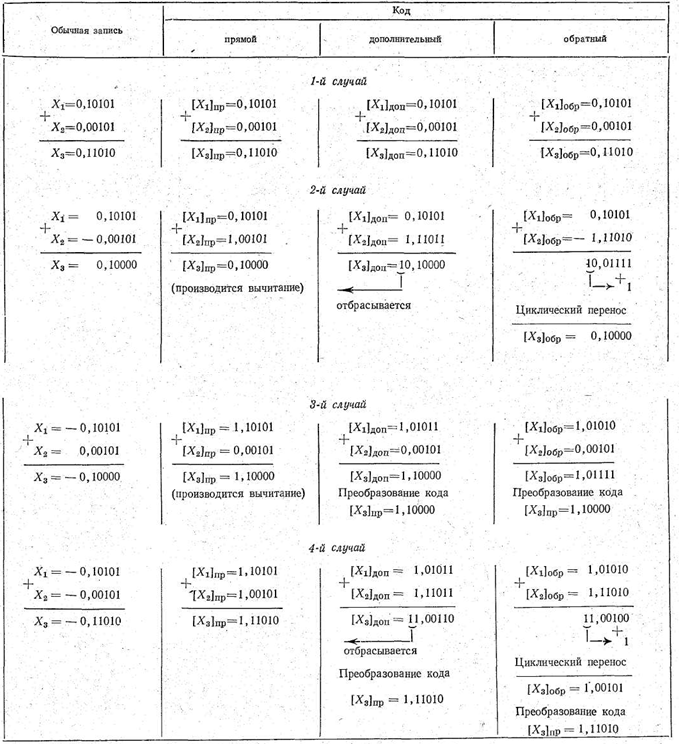
Поведение работы автомата описывается е общими формулами Y=λ (X,T),

D=δ (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 простейшего цифрового устройства сложения двух двоичных чисел в дополнительном коде. Условимся, что одно из чисел D1, будет всегда положительным, а второе может быть как положительным так и отрицательным. Правило сложения чисел в различных кодах приведено на рисунке 2.

Рисунок 2 - Правила сложения двоичных чисел

Структурная схема одного из вариантов операционной части устройства приведена на рисунке 3.

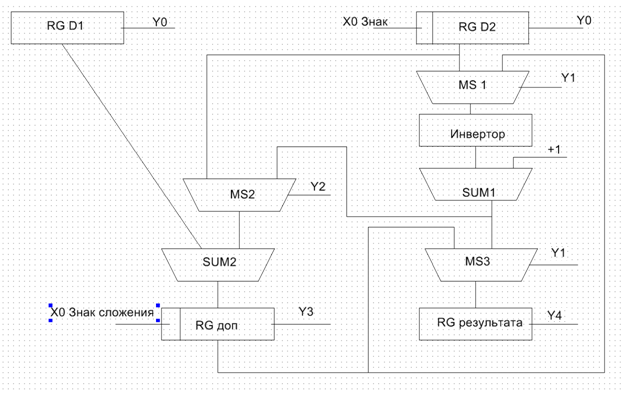


Рисунок 3 - Структурная операционной части

где: RG D1,RGD2 - входные регистры для хранения входных данных;

MS1, MS2, MS3 - цифровые мультиплексоры 8 на 4;

Инвертор - для получения обратного кода;

SUM1, SUM2 - сумматоры;

RGдоп - регистр для хранения промежуточной суммы в дополнительном коде;

RG результата - регистр для хранения результата сложения.

)

**Работа устройства**.

Данные по переднему фронту сигнала У0 D1 и D2 записываются в входные регистры( см. схему "Регистр" в папке лабораторной работы). Если оба числа положительные, знак Х0 равен 0 и D2 проходит через открытые по умолчанию мультиплексоры MS1, MS2(см. схему "Мультиплексор" в папке лабораторной работы).Результат сложения записывается в RG доп.

Если результат сложения положительный, то значение суммы через нормально открытый MS3 сигналом Y5 записывается в RG результата.

Если D2 отрицательно (Х0=1), то D2 проходит через нормально открытый MS1, далее инвертируется, на сумматоре к нему прибавляется единица, и дополнительный код D2, через открытый сигналом Y2 мультиплексор MS2 поступает на сумматор.

Если результат сложения в RG доп положительный (Х1=0), результат сложения записывается в RG результата сигналом Y5.

Если результат отрицательный (Х1=1), то результат сложения по сигналу Y1через мультиплексор MS1 поступает на инвертор, после к нему добавляется единица и результат сложения по сигналу Y4 пройдя мультиплексор MS3 по сигналу Y5 записывается в RG результата.

**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:

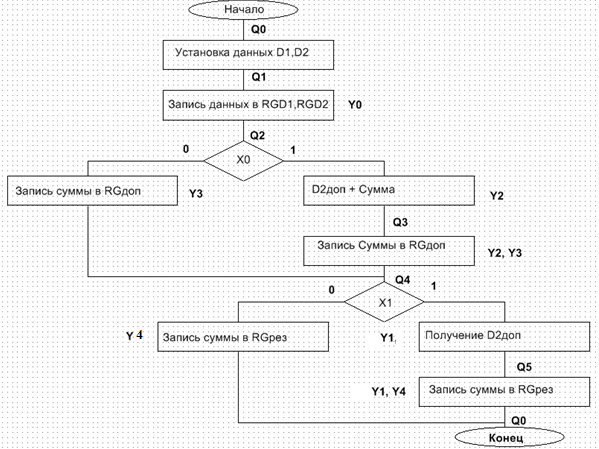


Рисунок 4 - Размеченная ГСА автомата Мили

Как видим наш автомат должен иметь k = 6 состояний (Q0-Q5). Для его построения необходимо  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 | | |
|  | Т1 | Т2 | Т3 |
| 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 | | | Выходныесигналы | | | | | |
|  | 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,Y4и D1,D2,D3.





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

**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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | T1T2 |  |  |  |
| X1Т1 | 00 | 01 | 11 | 10 |
| 00 |  |  |  |  |
| 01 |  |  |  |  |
| 11 |  | 1 |  |  |
| 10 |  | 1 |  |  |

После склеивания получим выражение для Y1

Проведя аналогичные действия над Y2, D3 получим соответственно

Значения Y3, Y4 остаются без изменения.

Возможные значения D1-D3:

**Шаг 6. Построение принципиальной схемы управляющего автомата Мили.**

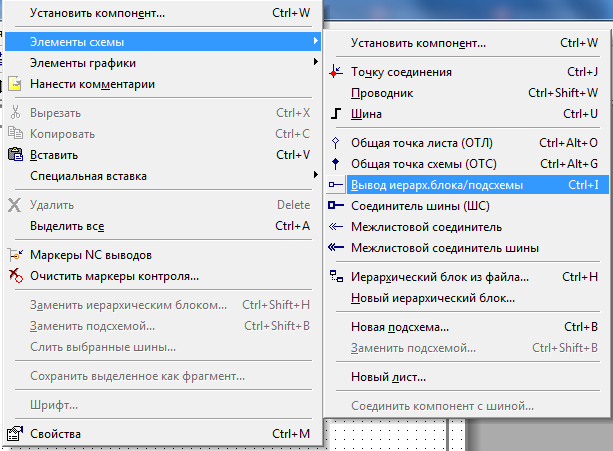
**Управляющая часть**.

Управляющая часть строится на основании полученных логических выраженй 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 параллелно 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).

Управляющая часть приведена на рисунке 5.

 Операционная часть представлена в виде подсхемы (прямоугольник с входами и выходами). Для создания подсхемы операционной части необходимо ко всем внешним входам и выходам предварительно подсоедигить 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.

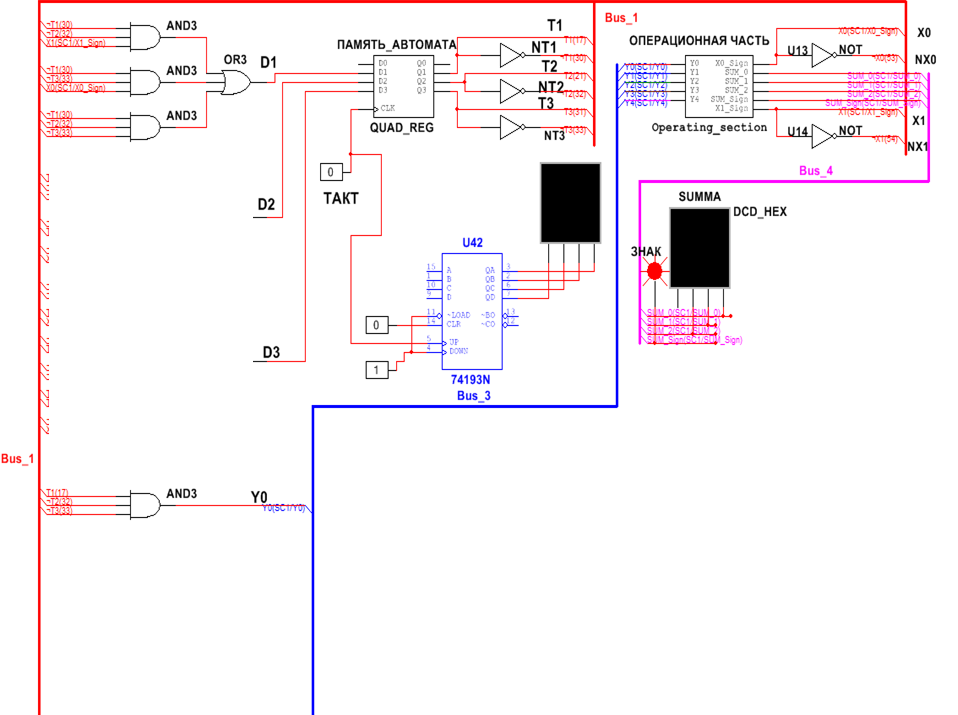
****

Рисунок 5 - Фрагмент схемы управляющего автомата Мили

**Шаг 7- Построение принципиальной хемы операционной части**

Далее по структурной схеме на рисунке 3 строим принципиальную схему операционной части

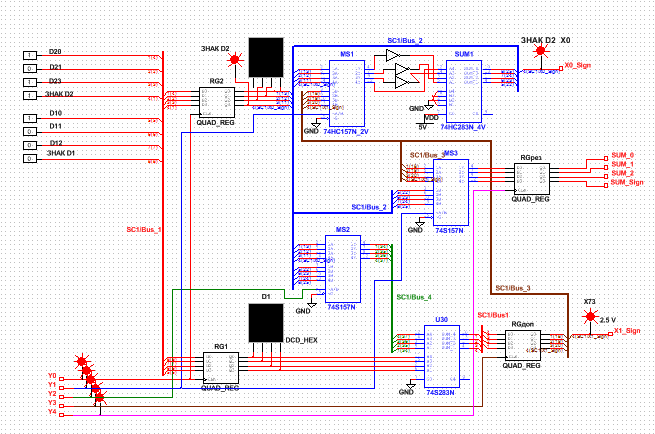


Рисунок 6 - Принципиальная схема операционной части

Для построения схемы надо ознакомиться со схемами включения основных элементов. Схемы приведены в папках «Использование микросхем Multisim» «Использование микросхем Proteus» лабораторной работы.

Для устранения загромождения чертежа необходимо использовать элемент групировки проводников "Шина" (Вставить\Шина). С правой стороны в шину подключаюься входные сигналы с левой выходные. Все сигналы в шине имеют собственный идентификатор, который для удобства может задавать ползователь.

Как видно в операционную часть с переключателей D11-D13 D21-D23 поступают два числа D1, D2 и их знаки. В операционную часть также поступают сигналы управления узлами операционной части Y0 - Y5 с управляющего блока. Выходными сигналами операционной части являются переменные определяющие знаки X0, X1 и значения разрядов результирущей суммы SUM1 - SUM2.Эти сигналы поступают в управляющую часть.

Для контроля входных данных и результата удобно использовать HEX индикаторы.

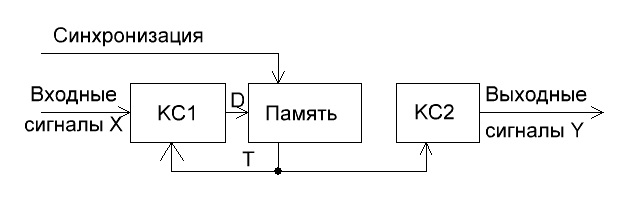
Вся операционная часть оформляется в виде подсхемы, входящей в управляющую часть.

Сумматор SUM1 служит для добавления единицы к обратному коду для получения дополнительного кода. Для этого на самый младший вход подается 1, а на остальные входы сигнал 0. Все непонятные места удобно проверять с помощью логического пробника - светодиода.

Шаг **8 Проверка работоспособности схемы в среде Multisim**

**1.3 Синтез управляющего автомата Мура**

Автомат Мура отличается от Мили тем, что он описывается формулами Y=ƒ1(T), D=ƒ2(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=ƒ2(X,T), а КС2 - Y=ƒ1(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 отмечаются все **операторные вершины** ГСА.

3. Каждая опраторная вершина помечается одним символом.

Рамеченная ГСА автомата Мура приведена на рисунке 7.

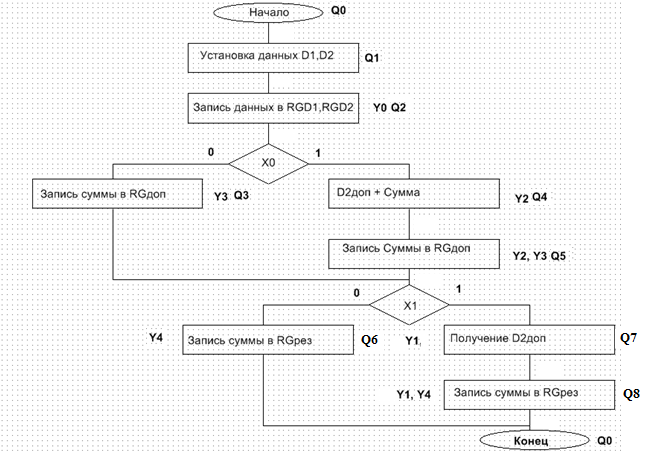


Рисунок 7 Размеченная ГСА автоиата Мура

**Шаг 2. Кодирование состояний и таблица выходов для построения КС2**

Для КС2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер  состояния | Код состояния | | | | Значения выходных сигналов  Y |
|  | Т1 | Т2 | Т3 | Т4 | 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 аналогичноавтомату Мили.

Например,

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.

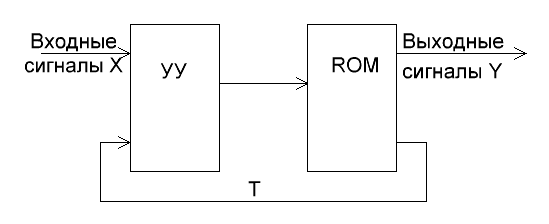
Однако есть и большие недостатки:

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 и тяжёлым занятием. Так, карты Карно уже при 5 аргументах становятся трудночитаемыми и труднопонимаемыми, т.к. не все клетки, которые можно склеить и минимизировать, являются физически соседними. 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.

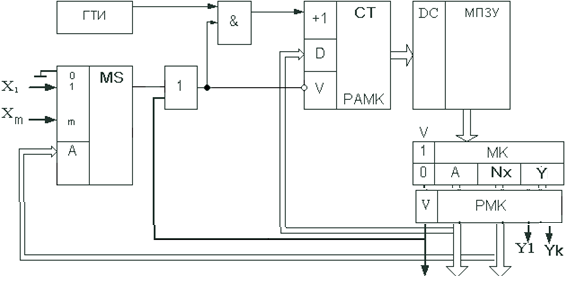


Рис 9. Структура МП УА с естественной адресацией.

Микропрограмма зашита в микропрограммное постоянное запоминающее устройство МПЗУ. Каждая микрокоманда состоит из четырех полей **V: A : Nx : Y**

Существует два типа микрокоманд, различающихся старшим битом V:

V=1 – ОМК – операционная микрокоманда, содержащая в поле Y значения бит соответствующие сигналам   Y1 - Yn.

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

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

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

Y1 – Yn – 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 Yi. По приходу очередного тактового импульса счётчик увеличится на 1, последовательно выбирая следующую МК.

Если выполняется УМК (V=0), то выходные значения Yi сохраняют предыдущее значение, а элемент ИЛИ разрешает прохождение сигнала данных с мультиплексора. Далее, если Xi=0, то прохождение тактовых импульсов на счётчик запрещается, но разрешается предзагрузка значением поля адреса A микрокоманды и выполняется переход. Если Xi==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.

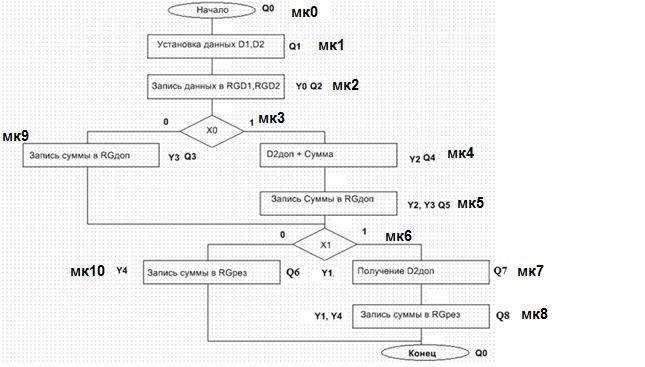


Рис. 10 Схема-алгоритма работы микропрограммного автомата

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

Шаг2. Кодировка входных сигналов автомата x0, x1(знак слагаемого и знак результата) и постоянный лог. 0(для безусловных переходов) на нулевом входе мультиплексора MS1:

|  |  |
| --- | --- |
| Вход данных мультиплексора | **Поле Nx микрокоманды** |
| 0 вход – логический 0 | 00 |
| 1 вход – x0 (знак слагаемого) | 01 |
| 2 вход – x1 (знак результата) | 10 |

Шаг 3. Разработка микропрограммы.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №  МК | Исх.  сост. | Адрес  на входе ПЗУ | | | | Адрес  перехода | | | |  |  |  |  | Выходные сигналы | | | | |
|  |  | А3 | А2 | А1 | А0 | А3 | А2 | А1 | А0 | X1 | X0 | V | hex | Y4 | Y3 | Y2 | Y1 | Y0 |
| Мк0 | Q0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | F | 0 | 0 | 0 | 0 | 0 |
| Мк1 | Q1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 17 | 0 | 0 | 0 | 0 | 0 |
| Мк2 | Q2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1F | 0 | 0 | 0 | 0 | 1 |
| Мк3 | Q2 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 52 | 0 | 0 | 0 | 0 | 1 |
| Мк4 | Q4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 2F | 0 | 0 | 1 | 0 | 0 |
| Мк5 | Q5 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 37 | 0 | 1 | 1 | 0 | 0 |
| Мк6 | Q5 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 64 | 0 | 1 | 1 | 0 | 0 |
| Мк7 | Q7 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 47 | 0 | 0 | 0 | 1 | 0 |
| Мк8 | Q8 | 1 | 0 | 0 | 0 | 1 | 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 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 5F | 0 | 1 | 0 | 0 | 0 |
| Бп | Q3 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 30 | 0 | 1 | 0 | 0 | 0 |
| Мк10 | Q6 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 6F | 1 | 0 | 0 | 0 | 0 |
| Бп | Q7 | 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 Х0 (знак слагаемого), поэтому данная команда, является командой условия (перехода) и V=0. Если значение Х0=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+Nx+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+Nx+A, записываются в одно ПЗУ, а биты, отвечающие за Y записываются в другое ПЗУ в формате hex файла. На следующем рисунке приведены фрагменты начала heх-файлов для обоих ПЗУ .

Шаг4. Разработка принципиальной схемы микропрограммного автомата .

Принципиальная схема автомата приведена на рисунках 11, 12,13.



Рис 11. Базовая принципиальная схема управляющей части (микропрограммный автомат вариант1)



Рис. 12 Принципиальная схема операционной части сложения чисел в дополнительном коде

**

Рис. 13 Общая принципиальная схема устройства сложения чисел в дополнительном коде

*(Правило рисования шин и нумерации входящих и выходящих проводников в шины приведено на стр. 28 документа "Описание Proteus.pdf".*

*Если схема устройства получается громоздкой, то её часть можно оформить в виде субсхемы и разместить на дочернем листе. Правило создания субсхем приведено на стр. 131 документа "Описание Proteus.pdf".)*

Если количество микрокоманд не превышает 16 (в большинстве заданий так и есть) , то можно использовать упрощенную схему микропрограммного автомата, приведенную на рисунке 14. Здесь используется только один счетчик, который может адресовать 16 микрокоманд:

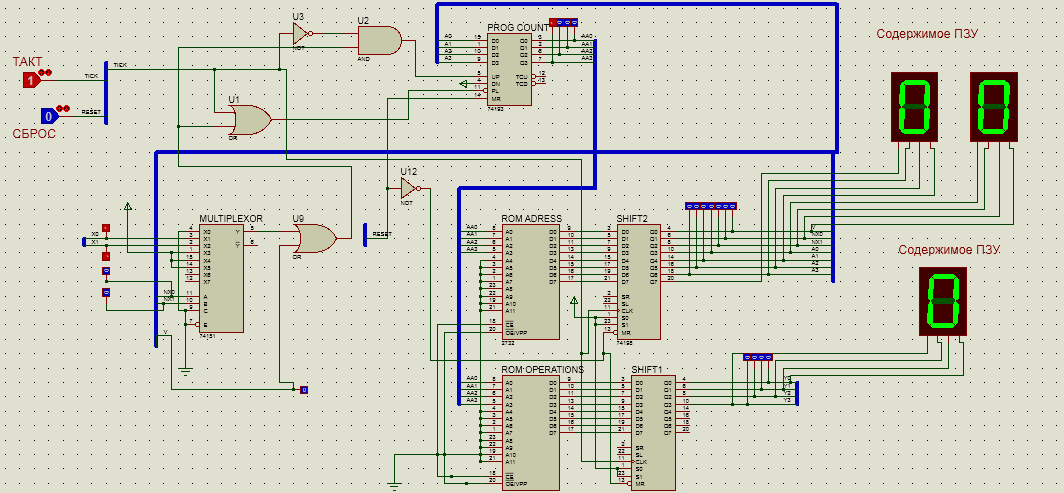


Рисунок 14 Принципиальная схема управляющей части (микропрограммный автомат вариант 2)

В качестве элемента ПЗУ используется ПЗУ 8К \*8 типа 27С128 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. 15).

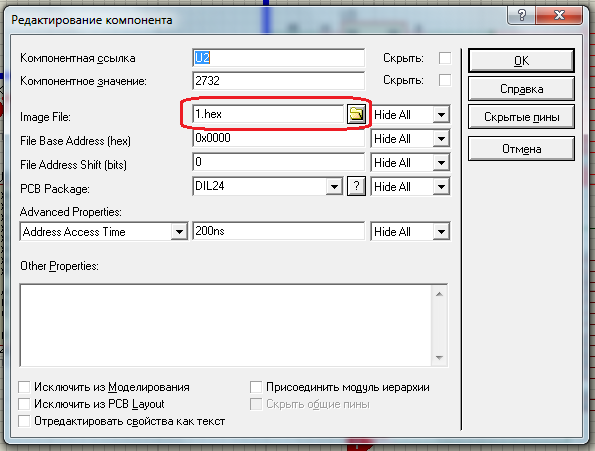


Рис. 15 Запись прошивки в файл

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 А,В входные значения Х0, Х1 последовательно поступают на выход мультиплексора.

Работа автомата :

1. обнулить счетчик;
2. подать сигналы Х0, Х1;
3. Кнопкой "ТАКТ" подать серию импульсов для синхронизации работы автомата.

Для прошивки ПЗУ, составленную программу необходимо представить в hex-формате.

Правила составления hex-файла и пример прошивки (1.hex-запись единицы по первому и нулевому адресу ) приведены в соответствующих файлах.

## 1.4.1Машинные методы умножения чисел в прямых кодах

Операция умножения состоит из ряда последовательных сложений. Сложением управляют разряды множителя: если в очередном разряде множителя содержится единица, то к сумме добавляется множимое. При этом, в зависимости от метода умножения, выполняется сдвиг либо множимого, либо частичной суммы. Наряду с этим умножение можно начинать как с младших, так и со старших разрядов множителя.

На рисунке 16 изоражена структурная схема операционной части умножения с младших разрядов множителя со сдвигом суммы вправо.

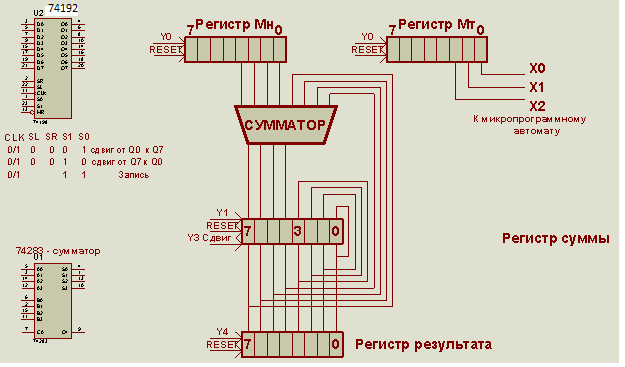


Рис. 16 Структурная схема операционной части умножения начиная с младших разрядов множителя со сдвигом частичной суммы вправо)

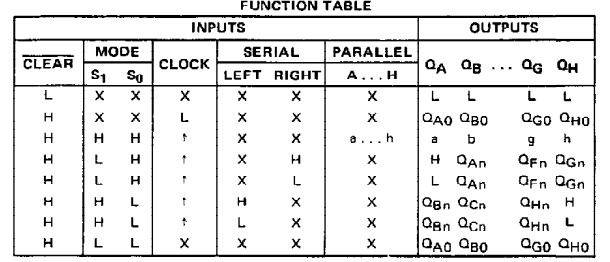
На рисунке 16а приведен один из вариантов принципиальной схемы операционной части умножения со сдвигом частичной суммы вправо. Множитель и множимое записываются в регистры данных RGD1 RGD4 соответственно. Для анализа значений битов множителя используется схема, состоящая из мультиплексора, счетчика количества сдвигов и компаратора. Выходы счетчика поступают на мультиплексор и задают номер анализируемого бита множителя **Х0**. Значение счетчика увеличивается после каждого сдвига да тех пор, пока текущее значение счетчика не сравняется со значением разрядности умножаемых чисел, задаваемым переключателями на цифровом компараторе U5. При достижении поседнего анализируемого разряда на выходе компаратора вырабатывается сигнал **Х1**. Сигналы Х0 и Х1 поступают в управляющую часть для анализа.

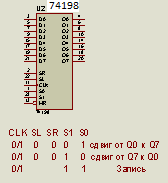


Рис 16а Принципиальная схема операционной части умножения с младших разрядов множителя со сдвигом частичной суммы вправо

В качестве основного регистра сдвига используется 8-битный универсальный последовательно-параллельный регистр 74198. Схема его работы приведена в таблице.

Режим работы задается с помощью управляющих бит S1 и S0





Режимы работы универсального регистра 74198

**

Рис. 16б Схема алгоритма умножения с младших разрядов множителя со сдвигом частичной суммы вправо

**Умножения начиная с младших разрядов множителя со сдвигом множимого влево**



Рис. 16е Структурная схема операционной части умножения начиная с младших разрядов множителя со сдвигом множимого влево



Рис. 16д Базова принципиальная схема операционной части умножения с младших разрядов множителя со сдвигом множимого влево

**

Рис. 16ж Схема алгоритма умножения начиная с младших разрядов множителя со сдвигом множимого влево

**Умножение со старших разрядов множителя со сдвигом суммы частичных произведений влево**



Рис. 16г Базовая принципиальная схема операционной части умножения начиная с старших разрядов множителя со сдвигом суммы частичных произведений влево

**

Рис. 16д Схема алгоритма умножения начиная со старших разрядов множителя со сдвигом суммы частичных произведений влево

Задание к лабораторной работе.

Устройства на основе микропрограммного автомата разрабатываются в среде моделирования Proteus. Устройства с управляющим автоматами Мура или Мили разрабатываются либо в среде Proteus или Multisim. При выполнении операций умножения для автоматов Мили или Мура, в качестве операционных частей, можно взять схемы, приведенные в лабораторной работе для Proteus и синтезировать к ним соответствующий автомат Мили или Мура. При работе в Multisim в качестве начальных схем, также можно использовать эти схемы, только модифицировать их, используя те же элементы только в среде Multisim.

Отчет должен содержать:

1 Структурную схему операционной части;

2 Принципиальную схему операционной части;

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 устройства в среде Multisim или Proteus.

9. Предусмотреть счетчик количества тактовых импульсов.

Отчет оформить в бумажном варианте

Варианты заданий:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № варианта | Разрядность  данных | Знак D1 | Знак D2 | Знак результата | Сложениедоп.код | Сложение  обратный код | Тип автомата |
| 1 | 3+1знак=4 | + | -/+ | -/+ |  | + | Мили |
| 2 | 3+1знак=4 | + | -/+ | -/+ |  | + | Мура |
| 3 | 3+1знак=4 | + | -/+ | -/+ |  | + | М П |
| 4 | 3+1знак=4 | + | -/+ | -/+ | + |  | Мили |
| 5 | 3+1знак=4 | + | -/+ | -/+ | + |  | Мура |
| 6 | 3+1знак=4 | + | -/+ | -/+ | + |  | М П |
| 7 | 3 | Умножение с мл.разрядов со сдвигом суммы вправо | | | | | МП |
| 8 | 3 | Умножение с мл.разрядов со сдвигом суммы вправо | | | | | Мили |
| 9 | 3 | Умножение с мл.разрядов со сдвигом суммы вправо | | | | | Мура |
| 10 | 3 | Умножение с ст. разрядов со сдвигом суммы влево | | | | | МП |
| 11 | 3 | Умножение с ст. разрядов со сдвигом суммы влево | | | | | Мура |
| 12 | 3 | Умножение с ст. разрядов со сдвигом суммы влево | | | | | Мили |
| 13 | 3 | Умножение с мл. разрядов со сдвигом множимого влево | | | | | Мура |
| 14 | 3 | Умножение с мл. разрядов со сдвигом множимого влево | | | | | МП |
| 15 | 3 | 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.
7. Обобщенная архитектура персонального компьютера Фон-Неймановской архитектуры.
8. Обобщенная архитектура процессора
9. Назначение компьютерной шины(аппаратного интерфейса), классификация шин.
10. Назначение контроллеров ввода-вывода.
11. Три способы адресации регистров(портов) контроллеров ввода вывода.
12. Архитектура процессора х8086, назначение его основных выводов.
13. Временная диаграмма работы процессора х8086 при выполнении циклов записи/чтения.