**Задание 8б**

**Синтез микропрограммного автомата с программируемой логикой   
по граф-схеме алгоритма**

**Цель** – приобретение практических навыков по проектированию микропрограммных автоматов с программируемой логикой (МПА ПЛ).

**Постановка задачи**

В качестве исходного материала для задания предлагаются алгоритмы реализации отдельных арифметических операций ЭВМ, представленные в виде закодированных граф-схем алгоритмов (ГСА). Индивидуальное задание представляет собой ГСА, приведенные в Приложении задания 8а. В процессе выполнения задания необходимо разработать микропрограммный автомат с программируемой логикой, реализующий функцию, заданную исходной граф-схемой.

**Подготовка к выполнению практического задания**

Ознакомиться с лекционным материалом по данной тематике и соответствующими разделами в литературных источниках [1, 2, 3].

**Порядок выполнения задания**

1. В соответствии с номером варианта выбрать ГСА.
2. Определить формат микрокоманды.
3. Выполнить разметку вершин микропрограммы и предварительно закодировать адреса микрокоманд.
4. Закодировать микрооперации.
5. Закодировать логические условия.
6. Закодировать микропрограмму.

**Пример.**

**Структура МПА.**

Общая структура МПА с ПЛ представлена на рисунке 1.

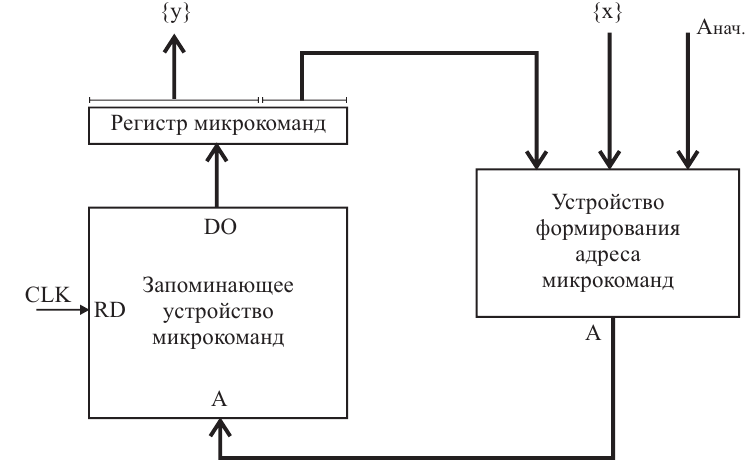


Рис. 1. Микропрограммный автомат с программируемой логикой

Будем использовать структуру МПА с естественной адресацией и смешанным кодированием в поле микроопераций (поле Y).

**1. Определение формата микрокоманды.**

Формат микрокоманды в общем виде приведен на рисунке 2.

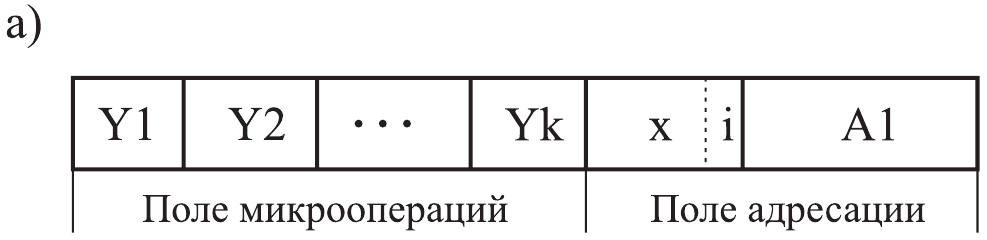


Рис. 2. Формат микрокоманды

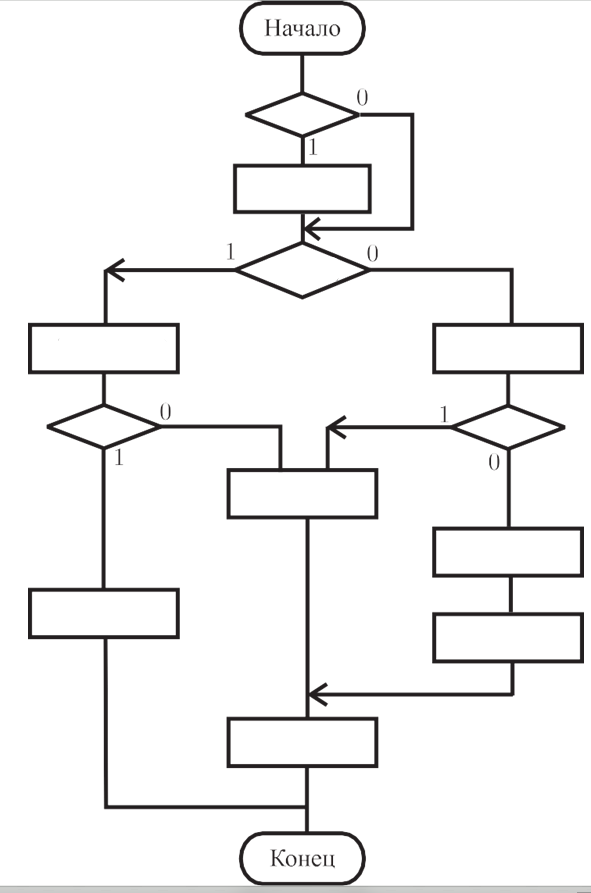
На разрядность полей микрокоманды влияют следующие параметры:

* количество различных микроопераций, формируемых УА, в конечном итоге определяет (с учётом выбранного способа кодирования) длину поля ***y*** (микроопераций);
* количество различных логических условий определяет длину поля ***x*** (логических условий и адресации);
* количество вершин ГСА связано с общим числом микрокоманд, а, следовательно, с объёмом памяти микропрограмм и разрядностью поля адреса микрокоманды.

Кодирование микрокоманд включает кодирование микроопераций, кодирование адресов микрокоманд (поле А1) и кодирование логических условий (поле Х).

**2. Разметка вершин микропрограммы.**

Выполним разметку вершин микропрограммы. Присваиваем каждой операторной вершине числовой номер.



***7***

***6***

***y7, y2***

***y5, y1***

***y5, y6***

***y1, y6***

***5***

***4***

***3***

***8***

***9***

***2***

***1***

***0***

Рис. 3. Граф-схема алгоритма   
(граф-схема микропрограммы)

***y8***

***y4***

***x3***

***x3***

***y3***

***y2***

***x2***

***x1***

Этот номер в дальнейшем будет являться предварительным адресом соответствующей микрокоманды. Каждой операторной вершине в микропрограмме ставится в соответствие микрокоманда с набором микроопераций, отмеченных в этой вершине. Вершина ***Начало*** отмечается номером **0**, если после этой вершины следует логическое условие, иначе номером **0** помечается первая операторная вершина после вершины ***Начало***. Вершина ***Конец*** также отмечается как операторная вершина.

В связи с использованием естественной адресации (если логическое условие равно 0, то к адресу прибавляется 1) нумерация вершин ведётся с номера 0 по ветви алгоритма, в которой логические условия равны 0: последовательность вершин и условий

**0 – (*х*1=1) – 1 – (*х*2=0) – 2 – (*х*2=0) – 3 – 4 –5 – 6**.

Далее аналогично нумеруются последовательности оставшихся вершин (в примере **7 – (*х*3=1) – 8** и **9**).

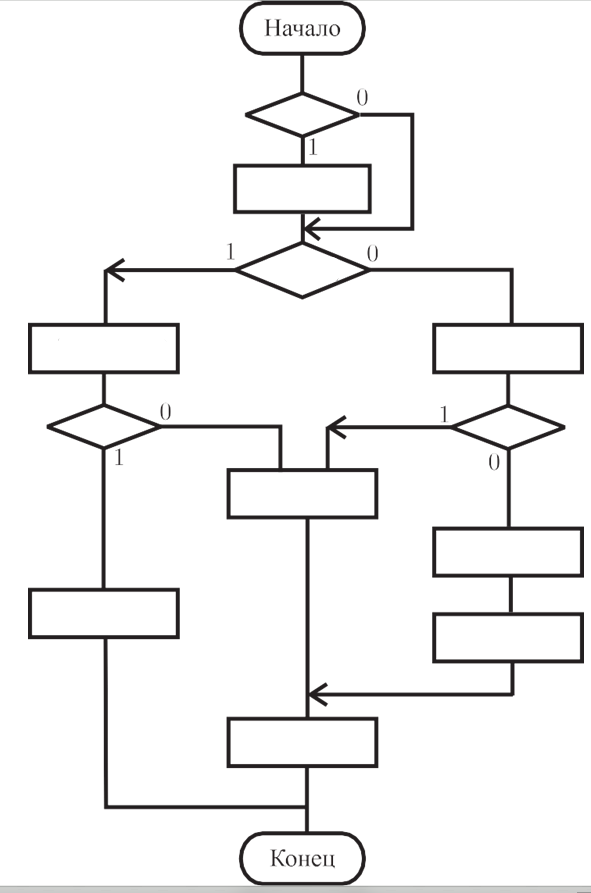
Всего получилось 9 микрокоманд. Для их кодирования достаточно 4-х двоичных разрядов (табл. 1).

Таблица 1 – Таблица кодирования адресов микрокоманд

|  |  |
| --- | --- |
| **А** | **Код** |
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

**3. Кодирование микроопераций.**

УА обычно должен вырабатывать кроме показанных на рисунке 1 микроопераций ещё одну микрооперацию, свидетельствующую об окончании выполнения алгоритма и предназначенную для использования не в ОА, а в управляющем автомате верхнего уровня иерархии, который запускает операцию, выполняемую операционным устройством и получает сигнал об окончании этой операции. Назовём эту микрооперацию ***yk*** и разместим ее в вершине ***Конец***.



***4***

***7***

***yk***

***6***

***y7, y2***

***y5, y1***

***y5, y6***

***y1, y6***

***5***

***3***

***8***

***9***

***2***

***1***

***0***

Рис. 4. Граф-схема микропрограммы

***y8***

***y4***

***x3***

***x3***

***y3***

***y2***

***x2***

***x1***

Множество микроопераций ***Y***, используемых в заданной ГСА —

***Y* = {*y*1*, y*2*, . . . y*8, *yk*}**,

мощность множества |***Y*|= 9**.

При горизонтальном кодировании необходимо в поле микроопераций ***n*=0** разрядов, при вертикальном кодировании ***n*>*log*2 9** и ***n*=4.** Вертикальный способ кодирования микроопераций к заданной ГСА неприменим, поскольку ГСА содержит вершины с двумя микрооперациями.

Мы рассматриваем смешанное кодирование.

Разбиваем множество ***Y***на подмножества несовместимых микроопераций.

Микрооперации, которые могут выполняться одновременно, называются ***совместимыми***. Совместимые микрооперации помещаются в различные подмножества. Поскольку в нашем примере максимальное число одновременно выполняемых микроопераций 2, выделяем два подмножества несовместимых операций (***s* = 2**): ***Y*1** и ***Y*2**. Это минимальное число подполей в поле микропераций, которое может быть.

Начинаем размещение микроопераций по этим полям с микрокоманд с наибольшим числом микроопераций. Выбираем микрокоманду 1: (***y1, y6***). Помещаем произвольно микрооперации в подмножества:

***Y*1 = {*y1*}**, ***Y*2 = {*y6*}**.

Теперь разместим по множествам микрооперации следующей микрокоманды **3**, содержащей также две микрооперации (***y5, y6***). ***y6*** уже размещена в ***Y*2**, размещаем ***y5*** в ***Y*1**:

***Y*1 = {*y*1*, y*5}*, Y*2 = { *y6*}*.***

Выбираем далее микрокоманду 4: (***y7, y2***). Обе микрооперации могут быть размещены произвольно:

***Y*1 = {*y*1*, y*5*, y7*}*, Y*2 = {*y6, y2*}*.***

Выбираем далее микрокоманду 8: (***y5, y1***). Обе микрооперации уже размещены, но размещены в одном множестве ***Y*1**. Этого не должно быть. В множестве ***Y*2** их также нельзя разместить, поскольку в этом множестве есть совместимые с ними микрооперации. Поэтому необходимо выделить еще одно подмножество ***Y*3** и одну из микроопераций (произвольно) поместить в это подмножество. Например, таким образом:

***Y*1 = {*y*1*, y7*}*, Y*2 = {*y6, y2*}*, Y*3 = {*y*5}*.***

Остальные микрокоманды 2, 5, 6, 7 и 9 содержат по одной микрооперации. Их можно размещать произвольным образом по выделенным выше множествам. При размещении необходимо учитывать, что каждое подмножество должно также содержать «пустую» микрооперацию, то есть код микрооперации (**00…0**), по которому в данном поле не выдается управляющий сигнал в операционный автомат (не выполняется ни одна микрооперация данного поля). Добавим «пустые» микрооперации в множества:

***Y*1 = {**∅***, y*1*, y7*}*, Y*2 = {**∅***, y6, y2*}*, Y*3 = {**∅***, y*5}*.***

Далее при размещении несовместимых микроопераций необходимо учитывать оптимизацию возможной разрядности полей. Разрядность поля ***li*** зависит от числа микроопераций ***ki*** в этом поле ***li* = *Log*2 *ki***.

В данный момент для кодирования поля ***Y*1** достаточно 2-х разрядов, поля ***Y*2** – также 2-х разрядов, поля ***Y*3** – одного разряда.

Микрооперация ***y2*** уже размещена в поле ***Y*2**. Остаются четыре микрооперации ***y3***, ***y4***, ***y8*** и ***yk***. «Свободных» мест в подмножествах ***Y*1**и ***Y*2** только два, поэтому придется увеличивать разрядность одного из полей, например, следующим образом:

***Y*1 = {**∅***, y*1*, y3, y7***, ***y8***, ***yk*}*, Y*2 = {**∅***, y2, y4***, ***y6*}*, Y*3 = {**∅***, y*5}*.***

В первом подмножестве 6 микроопераций, во втором – 4, в третьем – 2. Следовательно, разрядность поля ***Y*1** – 3 разряда, поля ***Y*2** – 2, поля ***Y*3** – 1. Суммарная разрядность поля микроопераций ***Y*** равна 6, что меньше разрядности 9 при горизонтальном кодировании микроопераций.

**Кодирование микроопераций** может осуществляться произвольно, например, так, как показано в таблице 2.

Таблица 2 – Таблица кодирования микропераций

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Код** | ***Y*1** | **Код** | ***Y*2** | **Код** | ***Y*3** |
| 000 | ∅ | 00 | ∅ | 0 | ∅ |
| 001 | ***y*1** | 01 | ***y*2** | 1 | ***y*5** |
| 010 | ***y*3** | 10 | ***y*4** |  |  |
| 011 | ***y*7** | 11 | ***y*6** |
| 100 | ***y*8** |  |  |
| 101 | ***yk*** |
| 110 | **–** |
| 111 | **–** |

**4. Кодирование логических условий.**

При кодировании логических условий к множеству условий добавляются два условия (***Const* «0»** и ***Const* «1»**) необходимых для микропрограммирования переходов между микрокомандами.

**Кодирование логических условий** может осуществляться произвольно, например, так, как показано в таблице 3.

Таблица 3 – Таблица кодирования логических условий

|  |  |
| --- | --- |
| **Код** | ***x*** |
| 000 | ***Const* «0»** |
| 001 | ***x*1** |
| 010 | ***x*2** |
| 011 | ***x*3** |
| 100 | ***Const* «1»** |
| 101 | **–** |
| 110 | **–** |
| 111 | **–** |

**5. Кодирование микропрограммы**

Таким образом, формат микрокоманды для нашего примера будет такой, как приведено на рисунке 5.

**A**

**Y1**

**Y2**

**Y3**

**X**

**i**

**5**

**6**

**1**

**3**

**4**

**76**

**976**

**1076**

**1176**

**1476**

Рис. 5. Формат микрокоманды

***y8***

***y4***

***x3***

***x3***

***y3***

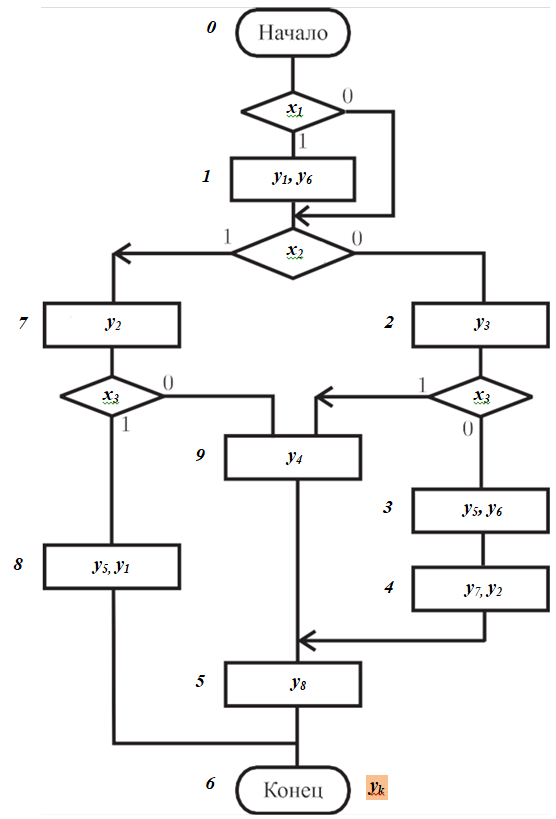
***y2***

***x2***

***x1***

Выбрав кодирование запишем микропрограмму микропрограмму в машинных микрокодах. Фактически мы формируем содержимое ПЗУ микропрограмм (табл. 4).

Таблица 4. Содержимое ПЗУ микропрограмм



**2**

**2**

**2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Адрес** | ***Y*1** | ***Y*2** | ***Y*3** | ***x*** | ***i*** | ***A*1** |
| **0** | 000 | 00 | 0 | 001 | 1 | 0010 (2) |
| **1** | 001 | 11 | 0 | 000 | 0 | xxxx |
| **2** | 000 | 00 | 0 | 010 | 0 | 1000 (8) |
| **3** | 010 | 00 | 0 | 011 | 0 | 1010 (10) |
| **4** | 000 | 11 | 1 | 000 | 0 | xxxx |
| **5** | 011 | 01 | 0 | 000 | 0 | xxxx |
| **6** | 100 | 00 | 0 | 000 | 0 | xxxx |
| **7** | 101 | 00 | 0 | 000 | 0 | xxxx |
| **8** | 000 | 01 | 0 | 011 | 1 | 1010 (10) |
| **9** | 001 | 00 | 1 | 100 | 0 | 0111 (7) |
| **10** | 000 | 10 | 00 | 100 | 0 | 0110 (6) |

**10**

**9**

**8**

**5**

**4**

**3**

**6**

**7**

Поскольку при значении ***x*1 = 0** осуществляется переход к проверке второго условия подряд, что невозможно при выбранном формате микрокоманды, то добавляется «пустая» микрокоманда 2 (выделена красным цветом) для проверки условия ***x*2**. В связи с этим произведена перенумерация микрокоманд. Новые номера (адреса) помечены красным цветом.

**Состав отчета по заданию**

1. Постановка задачи.

2. Исходная ГСА.

3. Все этапы синтеза МПА ПЛ по ГСА.

4. Выводы по работе.

**Литература**

1. Бабкин Е.А. Курс лекций. Раздел 7. – 2022 г.

2. Баранов С.И.Синтез микропрограммных автоматов (граф-схемы и автоматы). Ленинград. – «Энергия». Ленинградское отделение. – 1979 г.

3. Ожиганов А.А. Теория автоматов. Учебное пособие. – Санкт-Петербург: НИУ ИТМО, – 2013. – 84 с.