

Теперь задачей будет не просто составить программу решаемой задачи в микропрограммной памяти, а разбить её на куски (подпрограммы). После чего из них мы составим полноценную программу.

И на сей раз обращение к каждой такой подпрограмме будет храниться в оперативной памяти.

Осуществляться это будет следующим образом:

1. Составляем подпрограмму выборки команд из оперативной памяти.
2. Записываем нужные нам подпрограммы подряд в **микропрограммную память**.
3. В таблице преобразования адресов записываем номера этих команд и строку в микропрограммной памяти, откуда подпрограмма начинается.
4. Записываем эти команды в **оперативную память**.
5. Начинаем выполнение программы.

Начнём с рассмотрения что такое «**выборка команд**».

Выборка команд должна будет указать на следующую команду в оперативной памяти, считать её и отправить на выполнение.

После выполнения команды указатель СМК должен вернуться к микрокоманде 0, для того чтобы опять запустить выборку команд.

Лучше всего её закодировать таким образом:

Таблица 3 - Микропрограмма выборки команд				
Адрес МК	Операция	Поле	Значение	Функция
00	ARAM := IP IP := IP + 2	B	C	IP
		WM	3	ARAM:= RGB
		SRC	5	CONST, RGB
		ALU	3	Сложение
		DST	4	Запись в РЗУ
01	Чтение ОП RGK := RGR Дешифрация	CONST	2	
		MEM	5	Чтение слова
		B	E	RGK
		DST	1	РЗУ[B]:= RGR
		CHA	2	JMAP

В данном случае регистр **IP** будет хранить номер следующей ячейки памяти, которую надо будет считать.

1. Заносим в **ARAM** значение из регистра **IP** и прибавляем к **IP** двойку, чтобы он начал указывать на следующую команду.
2. Вторым действием высчитываем команду и заносим её в регистр **RGK**. В этом регистре как раз и происходит расшифровка команды. Это значит, что

он ищет соответствующее значение в таблице преобразования адресов и если находит, то переходит к выполнению нужной подпрограммы.

Для выполнения расшифровки в поле **СНА** записываем 2.

Теперь, когда записали выборку команд, надо запрограммировать сами команды.

Делать это вы уже умеете из второй методы. Но в данном случае есть один нюанс. Теперь в конце вы должны писать не команду «стоп» (JFI=5), а команду **СНА=0** для обнуления СМК и возврату к выборке команд.

Заполнили микропрограммную память и теперь надо заполнить «таблицу преобразования адресов».

В этой таблице справа выбираем номер команды (по сути, это как название команды, только на машинном уровне). Слева выбираем, с какой ячейки микропрограммной памяти начинается команда.

Теперь нужно перевести эти номера из 2-ого в 16-ичный код и последовательно записать их в память.

Команда стоп должна быть записана как отдельная подпрограмма, записана в память и обозначаться там как 111111100000000 или же FF00.

Теперь вы можете запускать программу.

Для удобства лучше поставить режим «К» (покомандно). В этом режиме при нажатии «продолжить» микрокоманды будут выполняться автоматически до появления нуля в **СНА**. То есть до конца команды(подпрограммы).

«Как указывать номера регистров в самих кодах операций»

Записать номера регистров в самой команде значит, что вы сможете написать подпрограмму 1 раз и после этого указывать для неё каждый раз разные регистры.

В коде операции есть 3 места куда мы можем записать номер регистра (то есть за раз мы можем указать 3 регистра, с которыми мы будем проводить операции).

00000XXX 00XXXXXX

Для записи номера каждого регистра отведено только 3 бита. Следовательно, таким образом мы можем указать только на первые 8 регистров (0-7).

Поле	A	B	MA	MB	ME
Бит	4	4	2	2	3
Def	0	0	0	0	0

Поля **МА** и **МВ** отвечают за выбор регистра не из полей микрокоманды, а из самого кода операции.

Они указывают номера регистров, соответственно, для первого и второго операнда, таким образом что:

- если в них записан 0, то номера регистров берутся из полей А и В
- Если записана 1, то номер регистра считывается из первой тройки (00000XXX 00000000)
- Если 2, то из второй (00000000 00XXX000)
- Если 3, то из 3-ей (00000000 00000XXX)

Для того, чтобы вы могли указывать номера регистров в коде операции, нужно сообщить микре, что в данном коде операции вот такие то биты могут принимать любое значение (то есть они не участвуют в формировании «названия» операции). Делается это очень просто – нужно вместо 0 или 1 писать X.

Например, если вы напишите «11111111 00XXXXXX», то команды в памяти FF00,FF13,FF3F и т. д. будут расшифровываться как эта подпрограмма.