

5. Методы низкоуровневого микропрограммирования.

Формирование управляющих функциональных сигналов может быть осуществлено несколькими способами:

Горизонтальное микропрограммирование.

Каждому разряду операционной части микрокоманды ставится в соответствие определенный управляющий функциональный сигнал, вызывающий определенную микрооперацию в операционном блоке.



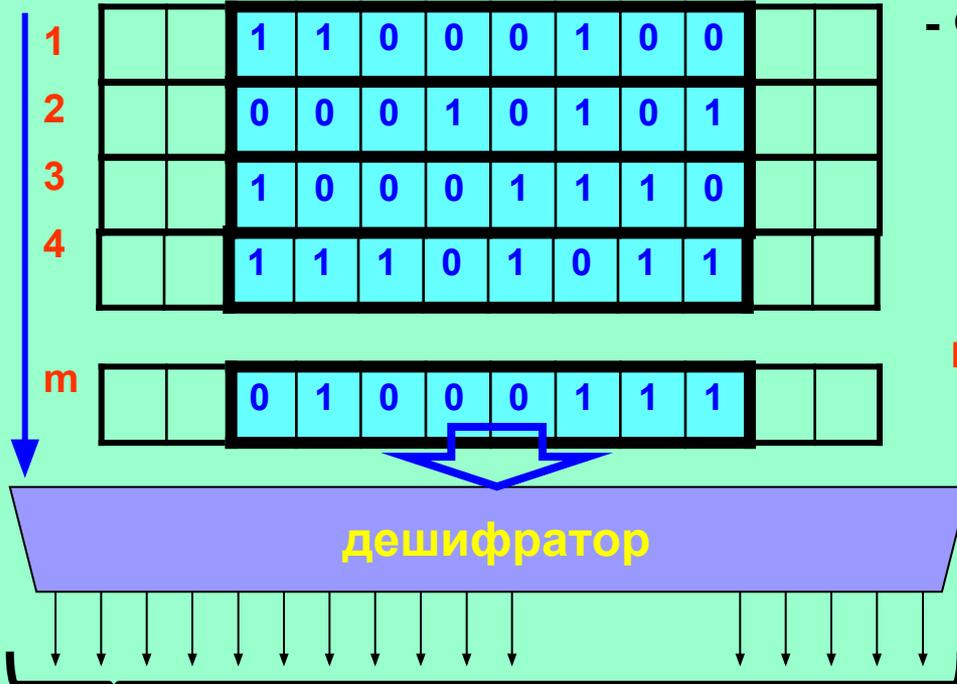
- Если в разряде «1», то операция выполняется независимо от других разрядов операционной части

m – число всех возможных микроопераций

Достоинства метода: возможность в одном такте выполнить одновременно любой набор из m микроопераций, а также простота формирования управляющих сигналов, возбуждаемых непосредственно от регистра.

Недостатки: требуется большая длина микрокоманды и, как следствие, число разрядов регистра, т.к. число сигналов может быть больше 100.

Вертикальное микропрограммирование.



- Функциональному сигналу ставится в соответствие не разряд, а двоичный код операционной части микрокоманды (есть даже код отсутствия микрооперации).

m – число функциональных сигналов

n – число разрядов операционной части микрокоманды

m \approx 100 сигналов

$$n = \log_2 (m+1) = 8$$

Достоинства метода: малая длина микрокоманды (8-16 разрядов)

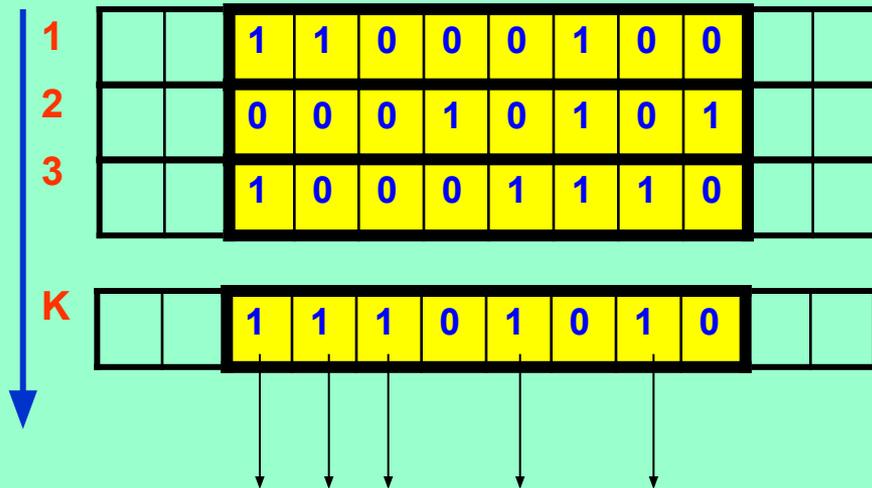
Недостатки: 1. сложность дешифрации кода сигнала микрооперации
2. Одной микрокоманде соответствует всего один функциональный сигнал поэтому число микрокоманд в пакете должно соответствовать числу функциональных сигналов.

Вертикально-горизонтальное микропрограммирование.

Сочетает достоинства горизонтального и вертикального методов.

Множество микроопераций разбивается на несколько подмножеств.

Микрооперации внутри подмножеств кодируются горизонтально.



- В каждое подмножество сводят типовые наборы микроопераций, встречающиеся в одном такте.

Подмножества строят равно-объемными, чтобы уравнять разрядность микрокоманд.

Достоинства метода: 1. Малая длина микрокоманды (8-32 разряда)

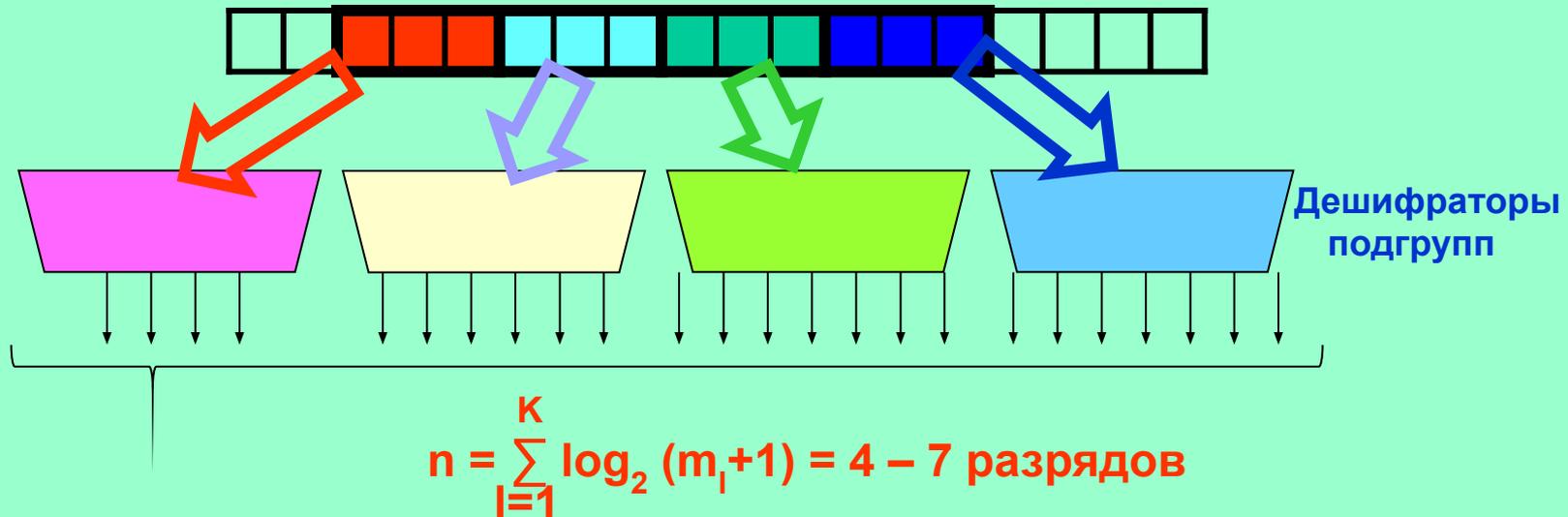
2. Сигналы представлены в явной форме и не требуют дешифрации.

Недостатки: 1. Вся совокупность микроопераций реализуется

несколькими микрокомандами за несколько тактов, при этом необходимо соблюдать порядок следования микрокоманд в пакете.

Горизонтально-вертикальное микропрограммирование.

Здесь также, как и в вертикально-горизонтальном методе все функциональные сигналы разбиваются на несколько подмножеств, которые располагаются горизонтально, т.е. в одной микрокоманде. Но внутри каждого подмножества сигналы кодируются (вертикально) и, следовательно, реализоваться в каждом такте будет только один сигнал из подмножества. Поэтому в подмножества сводят сигналы, которые не могут появляться одновременно с другими сигналами подмножества.



Достоинства метода: микрокоманда выполняется за один такт

Недостатки: 1. сложность дешифрации кода сигнала микрооперации

2. Сложность сведения функциональных сигналов в подмножества.

6. Проектирование и программирование микропрограммного автомата (МПА).

Принцип микропрограммного управления.

Для выполнения операций над информацией используются операционные устройства (ОУ) – регистры, АЛУ, мультиплексоры, шифраторы, дешифраторы, контроллеры и т.д..

Функцией операционного устройства является выполнение заданного множества операций $F = \{f_1, f_2, \dots, f_k\}$ над входными словами из множества D_{IN} с целью вычисления множества D_{OUT} , представляющих результаты операций $D_{OUT} = f_k(D_{IN})$.

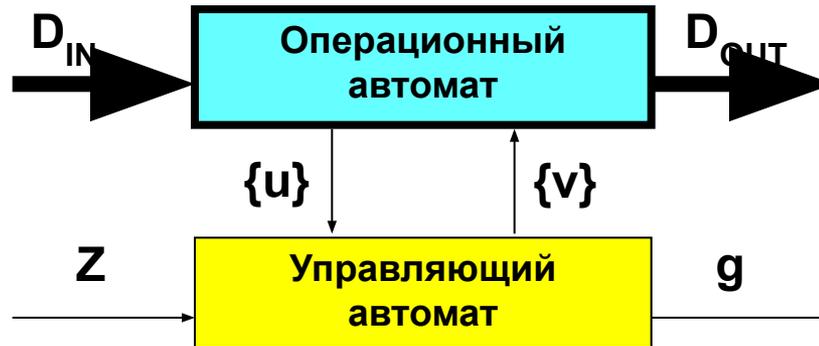
Организация архитектуры МПА базируется на следующих принципах:

1. Любая операция f_k , реализуемая ОУ, рассматривается, как сложный процесс, состоящий из последовательности элементарных действий над словами информации, называемых микрооперациями.
2. Порядок выполнения микроопераций может определяться наличием логических условий (сигналов оповещения).
3. Процесс выполнения операции f_k описывается в форме алгоритма и называется микропрограммой.
4. Микропрограмма используется как форма представления (активации) функций устройства, на основе которой определяется структура и порядок функционирования устройства во времени.

Строится тактовая диаграмма сигналов управления.

Концепция операционного и управляющего автоматов.

В функциональном отношении любое операционное устройство, входящее в состав ЭВМ, удобно представлять разделенным на две части: операционный и управляющий автоматы.



Операционный автомат служит для приема слов информации D_{IN} в различном формате, их хранения, преобразования и выдачи во внешнюю среду D_{OUT} . Этот процесс сопровождается выработкой множества осведомительных сигналов $\{u\}$, используемых в качестве логических условий для управляющего автомата.

Управляющий автомат генерирует последовательность управляющих сигналов $\{u\}$, распределенных во времени, обеспечивающих выполнение в операционном автомате заданную последовательность элементарных действий, которая реализуется алгоритмом выполняемой операции.

Обычно управляющие автоматы объединяют в единый блок, называемый центральным устройством управления, а конкретный алгоритм (микропрограмма) активируется кодом операции, который дешифруется в конкретный сигнал Z , определяющий тип выполняемой операции.

После завершения операции управляющий автомат активирует сигнал g , отмечающий завершение операции и готовность выходных данных D_{OUT} .

Пример проектирования операционного автомата.

В качестве исходных данных для разработки структуры операционного автомата являются:

- Описание формата входных и выходных слов (множеств $\{D_{IN}\}$ и $\{D_{OUT}\}$),
- Список множества операций из F , которые должны выполняться над словами.
- Разработка общей структуры операционного автомата.
- Тактовая диаграмма сигналов управления.

В качестве примера операционного автомата разработаем схему сумматора простых чисел.

Определение формата данных.

Будем считать, что в операции сложения участвуют два 32-х разрядных числа.

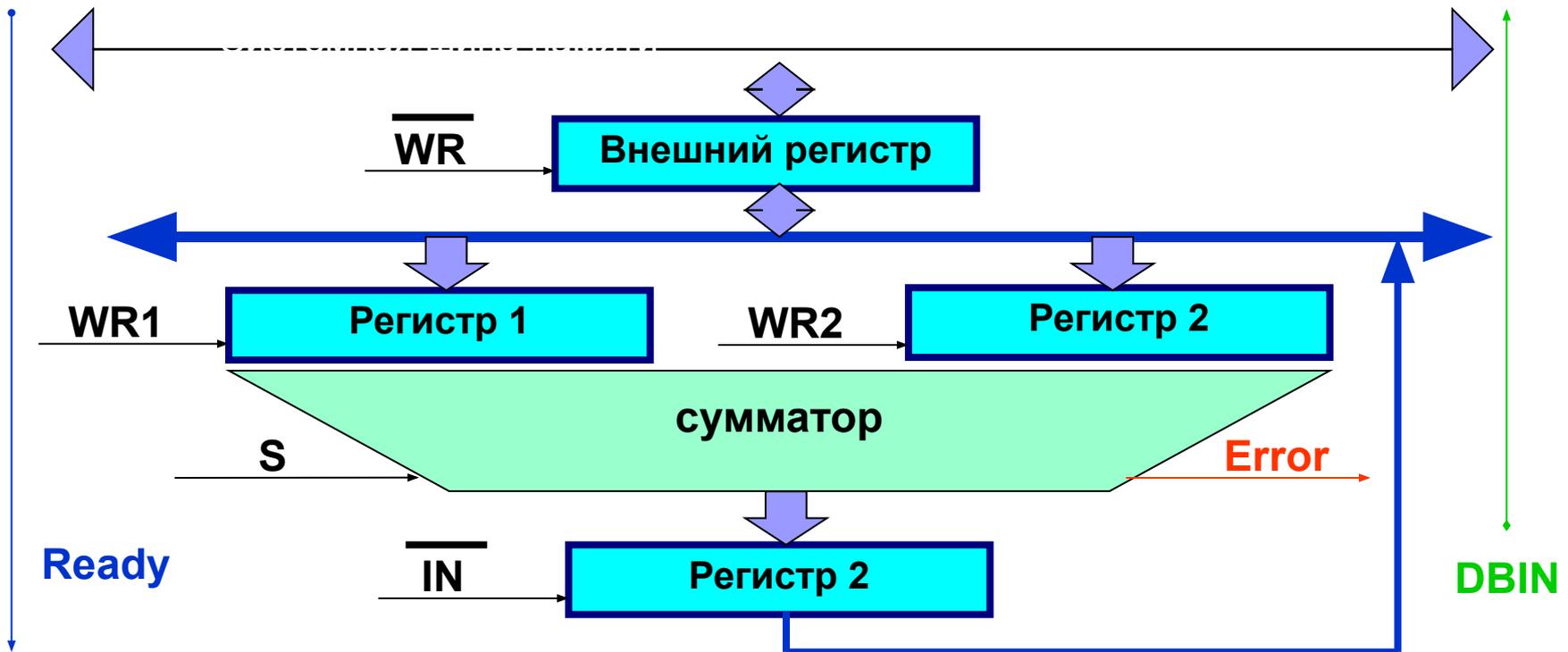
Слагаемые поступают с системной шины памяти по адресам памяти, определенной схемой страничной адресации, которая в данной структуре операционного автомата не рассматривается.

Сумма помещается в постоянно открытый аккумулятор, а из него во внешний буфер обмена системной шины для размещения по адресу второго операнда.

Основным логическим условием состоявшейся операции сложения будет служить отсутствие сигнала переполнения разрядной сетки - **Error**.

Признаком завершения – сигнал – **DBIN** (открытие внешнего регистра на чтение).

Разработка структуры операционного автомата



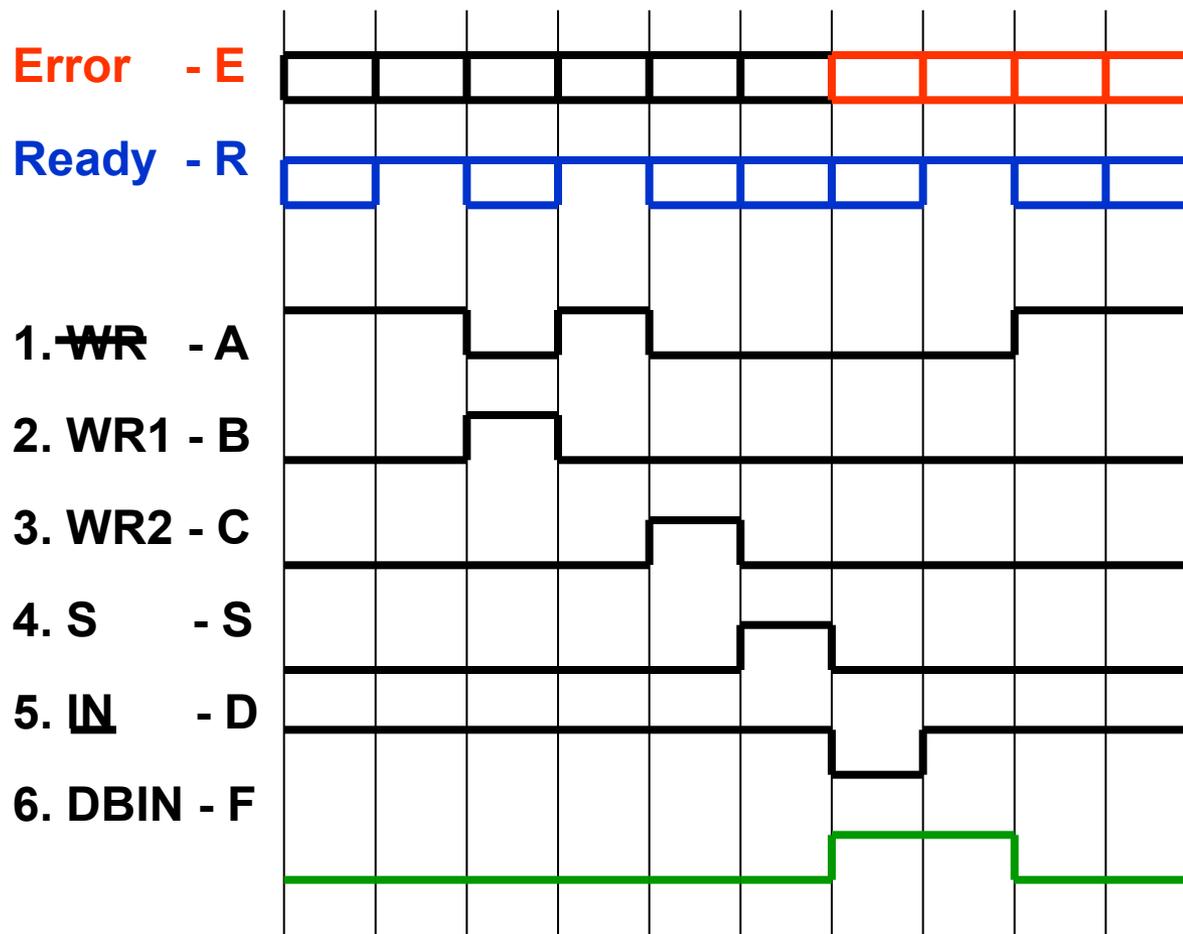
Основные сигналы управления;

1. \overline{WR} – открытие внешнего регистра системной шины на запись (сигнал постоянно активен, т.е. регистр постоянно открыт на шину данных).
2. $WR1$ – открытие первого буферного регистра на запись.
3. $WR2$ – открытие второго буферного регистра на запись.
4. S – сигнал активации сдвигового сумматора АПУ.
5. \overline{IN} - открытие регистра - аккумулятора на запись (сигнал постоянно активен, т.е. регистр постоянно открыт в сторону сумматора).
6. **DBIN** – открытие внешнего регистра на чтение данных в память.

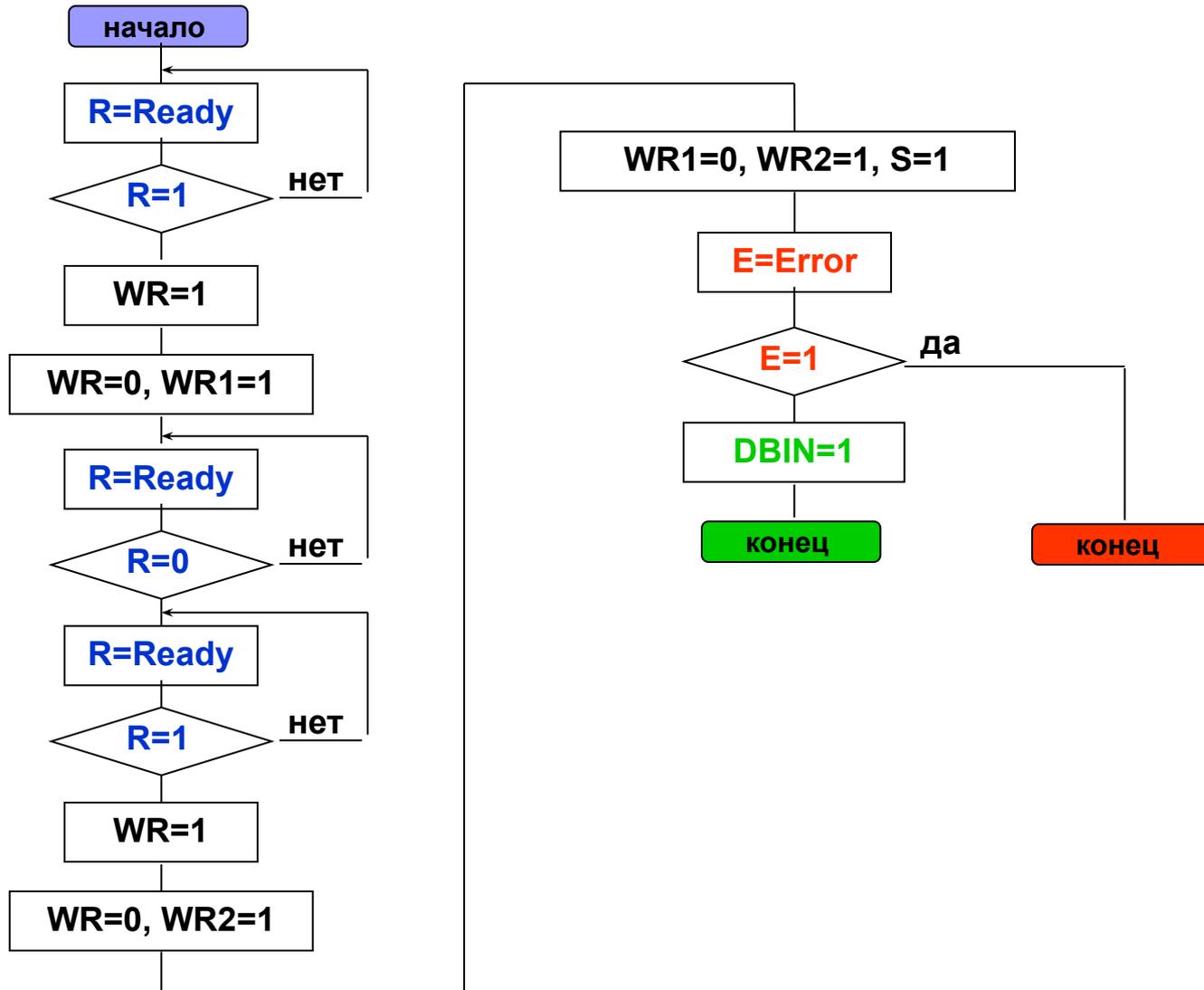
Оповещающие сигналы:

1. **Ready** – сигнал готовности контроллера памяти к операции чтение-запись.
2. **Error** – сигнал, источником которого является единица в старшем (дополнительном) разряде сумматора (переполнения разрядной сетки).

Тактовая диаграмма сигналов управления:



Разработка алгоритма выполнения операции сложения двух чисел.



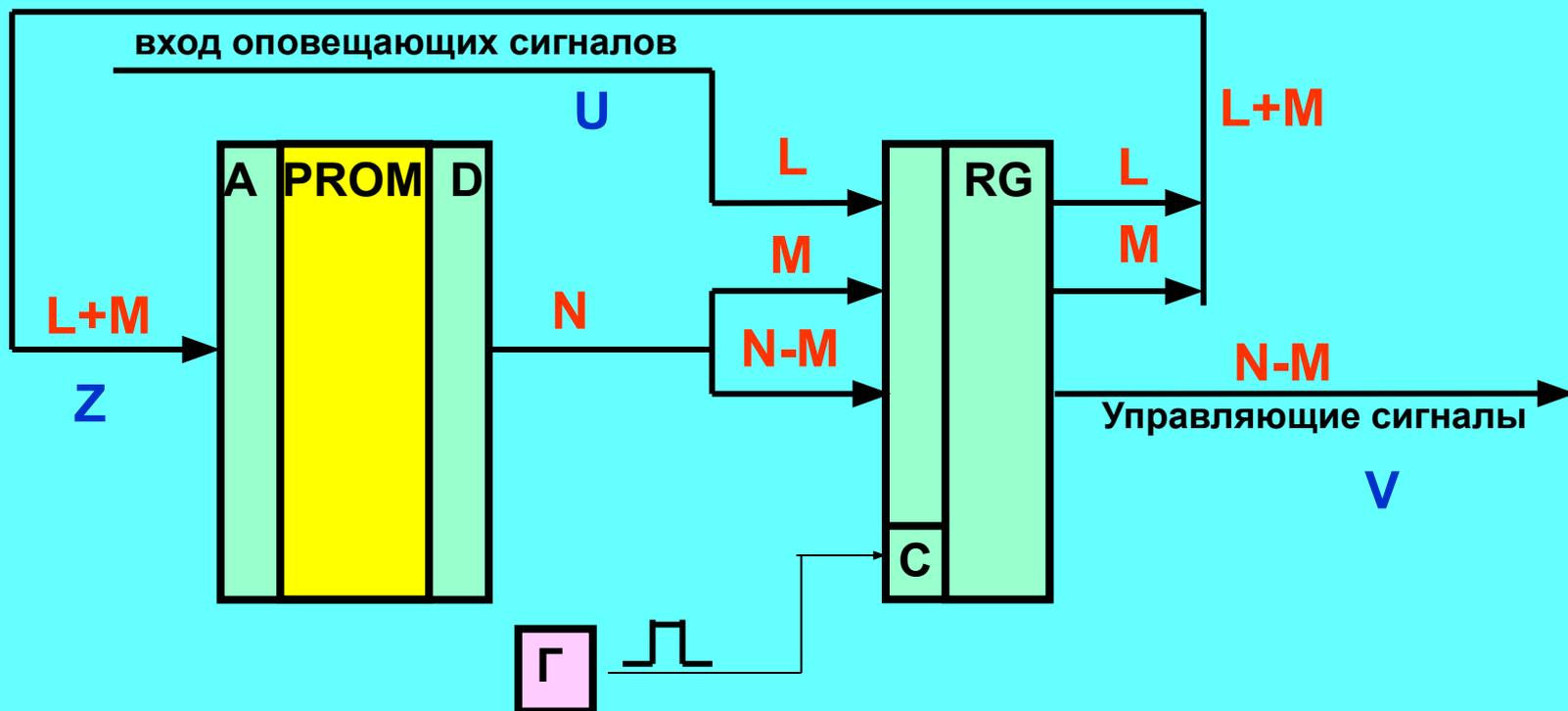
7. Разработка структуры управляющего микропрограммного автомата.

Принципиальная схема управляющего микропрограммного автомата.

Схема микропрограммного автомата синхронизируется с элементами операционного автомата с помощью генератора синхроимпульсов «Г».

ППЗУ является энергонезависимой долговременной памятью и синхронизации не подлежит, т.к. выдает данные на выход при появлении на входе адреса памяти.

Для сопряжения ППЗУ с остальной схемой используется регистр.





Порядок построения простой горизонтальной микропрограммы для микропрограммного автомата схемы генерации сигналов.

В ответ на положительный уровень сигнала, определяемого кодом операции сложения двух чисел, управляющим автоматом вырабатывается последовательность выходных сигналов, которые в нужные моменты времени открывают или закрывают буферные регистры, активируют сумматор.

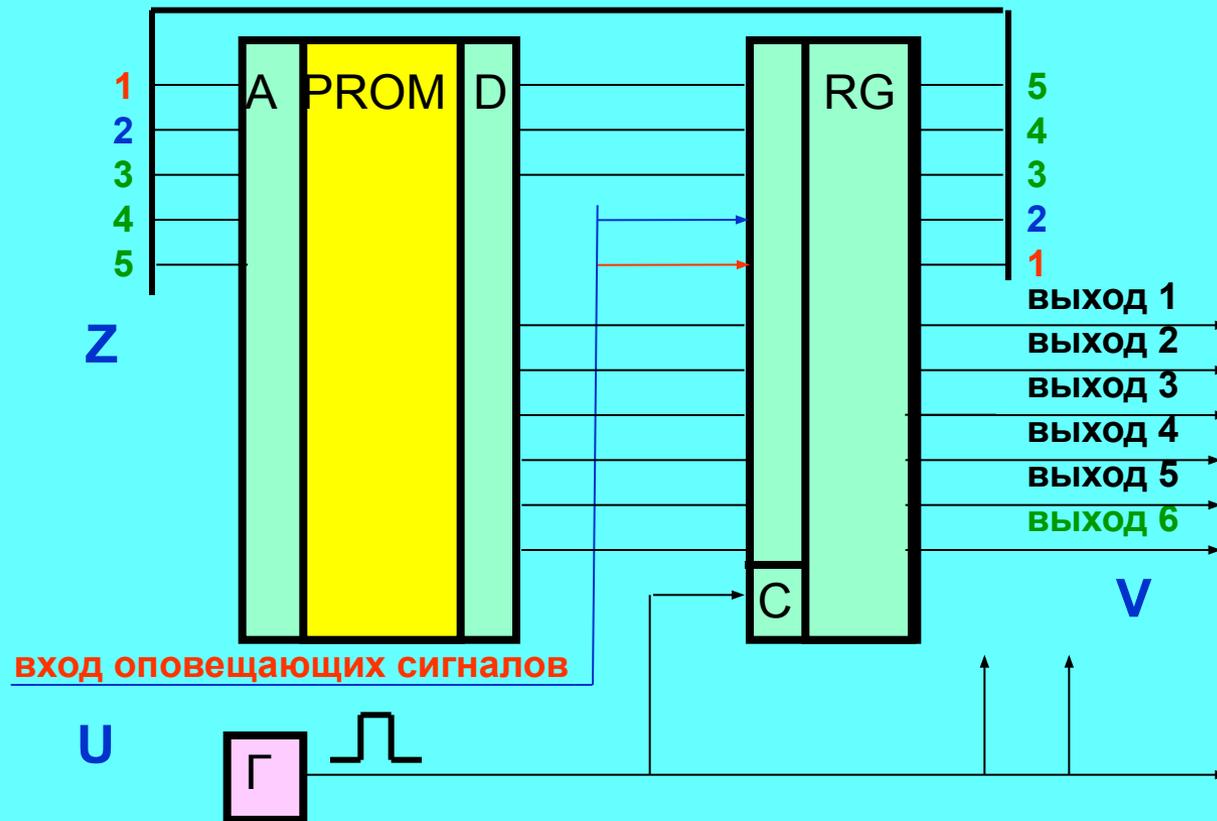
Во некоторые моменты времени при формировании выходной последовательности автомат не должен реагировать на входной сигнал, а после ее окончания должен ожидать следующего положительного уровня оповещающего сигнала готовности контроллера памяти.

Для упрощения задачи примем все временные сдвиги равными 1 мкс. Поэтому частота тактового генератора должна быть 1 МГц.

Как видно из временной (тактовой) диаграммы сигналов управления, выходная последовательность состоит из девяти тактовых моментов (включая все единичные уровни). Поэтому требуется, как минимум, 3 адресных входа ППЗУ (8 возможных состояний).

Кроме этого адресная часть формируется двумя входными сигналами, следовательно количество адресных разрядов ППЗУ должно быть не менее пяти, а количество разрядов регистра должно быть равно 9 (к трем разрядам адреса добавляется 6 разрядов данных, определяющих выходные сигналы).

Разработка схемы управляющего микропрограммного автомата.



В микропрограмме реализованы следующие режимы:

1. Последовательный перебор адресов (при отработке выходной последовательности).
2. Останов с ожиданием.
3. Отключение реакции на входной сигнал (путем дублирования последовательного перебора в зоне адресов, соответствующей изменению входного сигнала).

Общий вид микропрограммы в ППЗУ УМП

АДРЕСНАЯ ЧАСТЬ					СИГНАЛЫ						СЛ. АДРЕС			КОММЕНТАРИЙ
E	R	3	4	5	A	B	C	S	D	F	3	4	5	
0	0	0	0	0	1	0	0	0	1	0	0	0	0	Цикл ожидания сигнала готовности контроллера памяти
0	1	0	0	0	0	1	0	0	1	0	0	0	1	Заполнение буфера 1
0	1	0	0	1	1	0	0	0	1	0	0	0	1	Цикл ожидания снятия сигнала готовности контроллера памяти
0	0	0	0	1	1	0	0	0	1	0	0	1	0	Переход на ожидание сигнала готовности
0	0	0	1	0	1	0	0	0	1	0	0	1	0	Цикл ожидания сигнала готовности контроллера памяти
0	1	0	1	0	0	0	1	0	1	0	0	1	1	Заполнение буфера 2
0	1	0	1	1	0	0	0	1	1	0	1	0	0	Суммирование и сброс результата в аккумулятор
0	1	1	0	0	0	0	0	0	0	1	1	0	1	Заполнение внешнего буферного регистра и установка флага DBIN
0	0	0	1	1	0	0	0	1	1	0	1	0	0	Суммирование и сброс результата в аккумулятор
0	0	1	0	0	0	0	0	0	0	1	1	0	1	Заполнение внешнего буферного регистра и установка флага DBIN
0	0	1	0	1	0	0	0	0	0	1	1	0	1	Цикл ожидания сигнала готовности контроллера памяти
0	1	1	0	1	1	0	0	0	1	1	1	1	0	Запись результата в память на место второго операнда
0	0	1	1	0	1	0	0	0	1	0	0	0	0	Переход в начало программы (для цепочечных операций)
0	1	1	1	0	1	0	0	0	1	0	1	1	0	Цикл ожидания снятия сигнала готовности контроллера памяти
1	0	0	0	0	1	0	0	0	1	0	0	0	0	Повтор операций заполнения буферов и сложения при не критичное ошибке сумматора
1	1	0	0	0	0	1	0	0	1	0	0	0	1	
1	0	0	0	1	1	0	0	0	1	0	0	1	0	
1	1	0	0	1	1	0	0	0	1	0	0	0	1	
1	0	0	1	0	1	0	0	0	1	0	0	1	0	
1	1	0	1	0	0	0	1	0	1	0	0	1	1	
1	0	0	1	1	0	0	0	1	1	0	1	0	0	
1	1	0	1	1	0	0	0	1	1	0	1	0	0	
1	0	1	0	0	0	0	0	0	0	0	1	0	0	Обработка ошибки исполнения – перевод в цикл ожидания снятия сигнала
1	1	1	0	0	0	0	0	0	0	0	1	0	0	