

Архитектура ЭВМ. Специальный раздел.

Недельная аудиторная нагрузка:

лекции – 2 часа;

лабораторные занятия – 1 час;

практические занятия – 1 час;

курсовое проектирование (СРС – 2-4 часа).

***Консультация: пятница, чет., А-101,
15:40 (после лекции).***

Volkova T.V.

Тема курсового проектирования:

**СИНТЕЗ ЦЕНТРАЛЬНОГО
ОБРАБАТЫВАЮЩЕГО
УСТРОЙСТВА (ЦОУ) ЦВМ**

График выполнения курсового проекта

Номер этапа	Название этапов курсового проекта	Срок выполнения этапов проекта (неделя)
1.	Получение задания. Оформление листа задания	1
2.	Изучение литературы и уточнение постановки задачи. Определение структуры и разрядности основных элементов ЦОУ.	2
3.	Разработка форматов команд и уточнение форматов обрабатываемых данных.	3
4.	Изучение особенностей алгоритма выборки команды для заданной ширины выборки.	4
5.	Разработка отдельных ГСА исполнения команд.	5-6
6.	Построение общей ГСА функционирования процессора.	7
7.	Построение структурной схемы ЦОУ и архитектуры внешних выводов процессора.	8
8.	Разработка структурной схемы управляющего устройства (УУ).	9
9.	Разработка микропрограммы функционирования УУ.	10-11
10.	Разработка принципиальной схемы заданного узла УУ.	12-13
11.	Оформление пояснительной записки и чертежей	14-15
12.	Защита проекта (в соответствии с графиком защиты)	16-17

Лекция 1

**Структура учебной ЭВМ.
Типы и форматы данных. Способы
адресации операндов.
Разработка системы команд
процессора.**

Обобщенная структурная схема ЦОУ компьютера:

ОП – оперативная память, МВВ – модуль ввода/вывода,
А – адрес, Д – данные, УС ∨ ОС – управляющие или
осведомительные сигналы

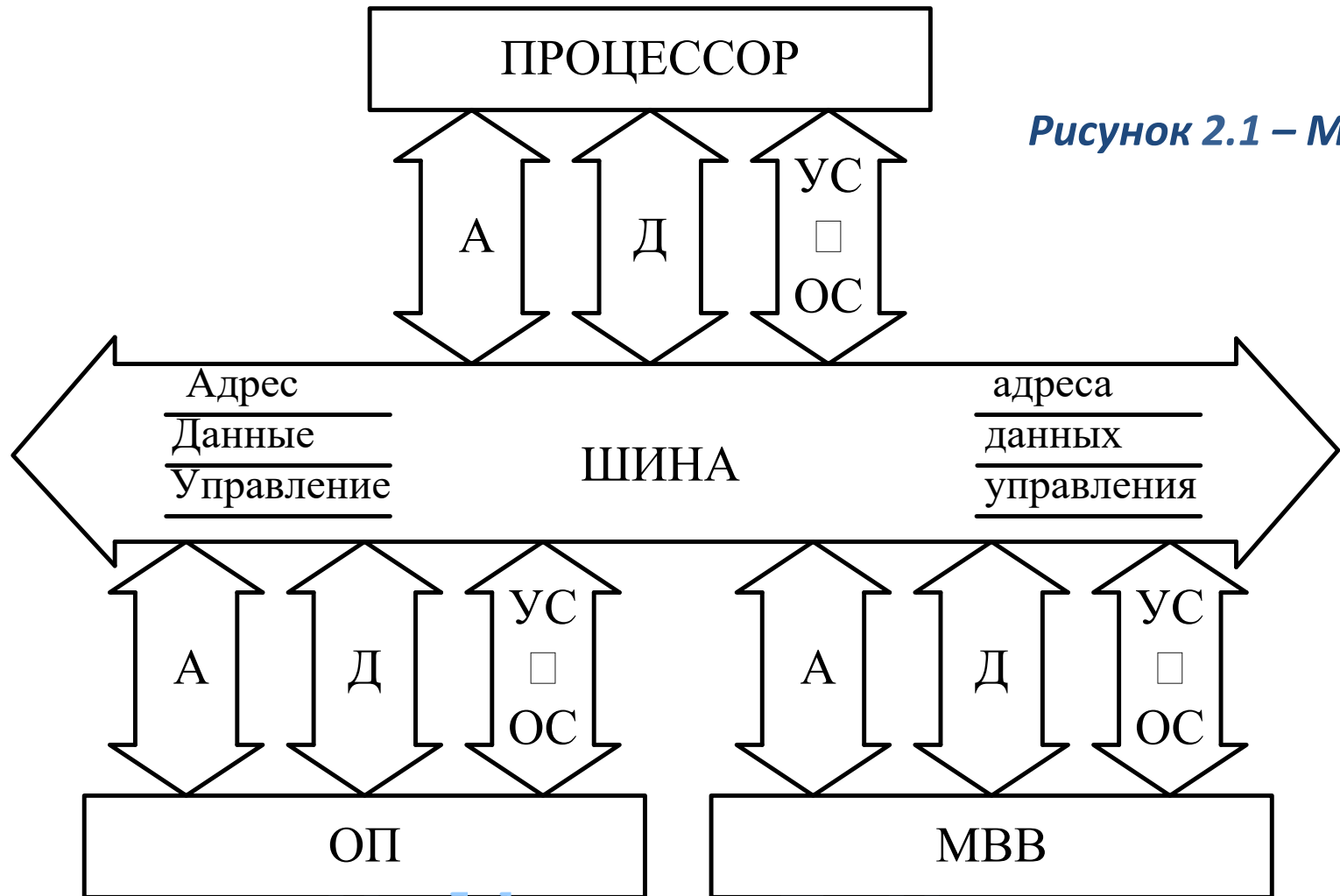


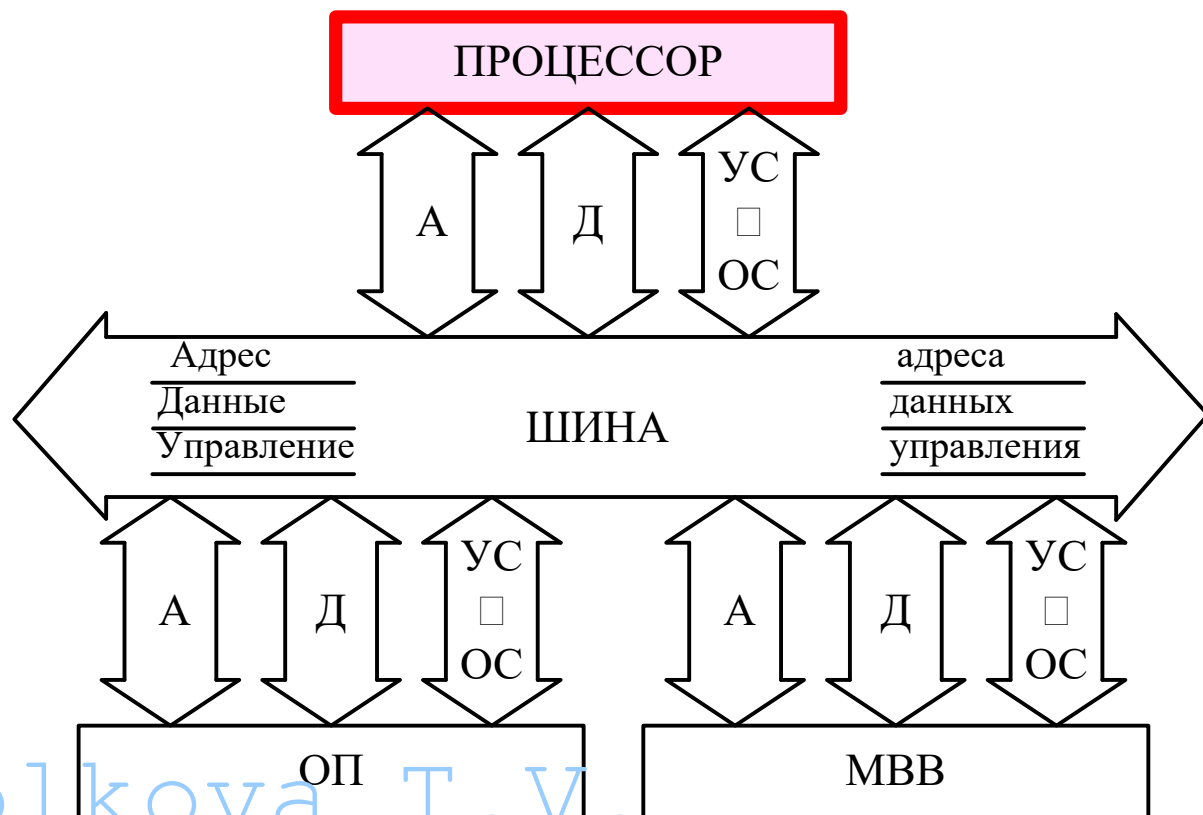
Рисунок 2.1 – МУ к КП

Предполагается, что архитектура ЦОУ – **принстонская** (команды и данные хранятся в одном и том же устройстве ОП).

Функции процессорного блока:

- управление потоком обработки команд исполняемой компьютером программы;
- управление процессом исполнения команды;
- управление процессом взаимодействия всех блоков ЦОУ.

*Распараллеливание
вычислительного
процесса в ЦОУ не
используется
(ограничение задачи КП!).*



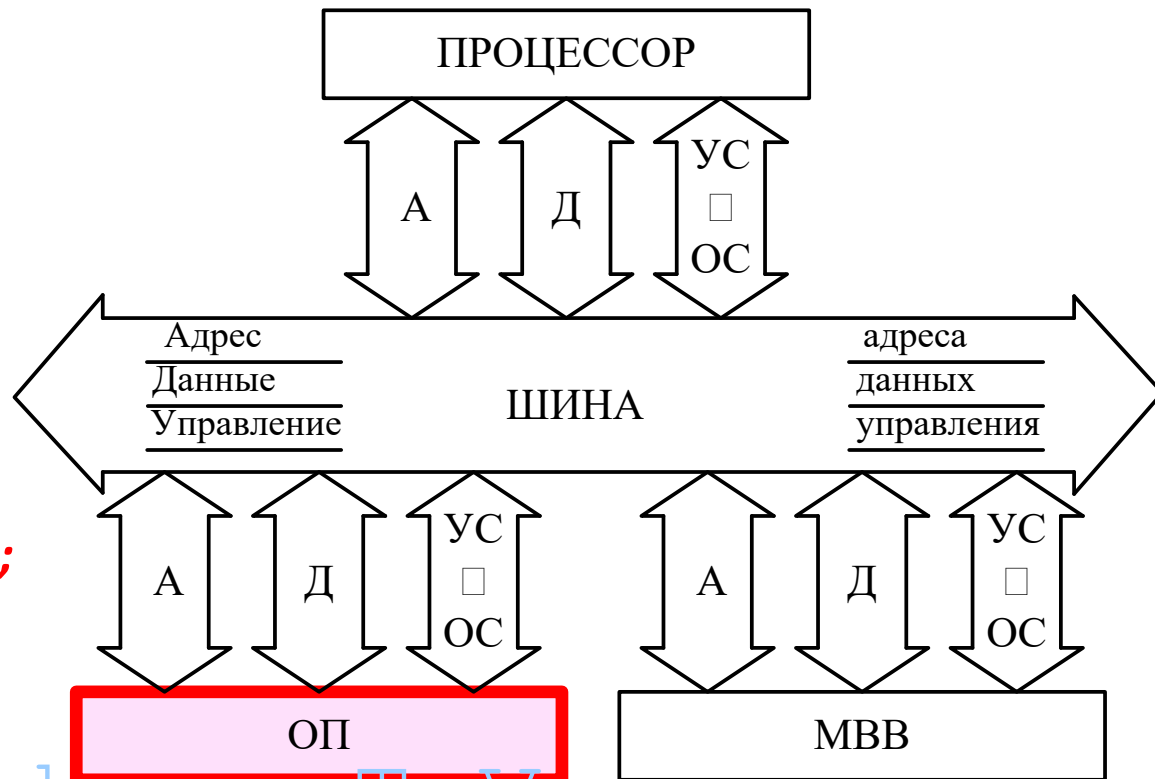
Оперативная память (ОП) является внутренней памятью компьютера и предназначена для хранения команд и данных, обрабатываемых компьютером в конкретный промежуток времени.

В структурном плане ОП реализуется в виде запоминающего устройства (ЗУ) с произвольным доступом: обращение к любой ячейке (слову) ЗУ осуществляется по уникальному адресу и может быть произведено в любой последовательности.

Размер ячейки (слова) ЗУ характеризуется **шириной выборки** (количество бит информации, одновременно записываемых или считываемых с этого устройства).

Ограничения задачи КП:

- инициатором обращения к ОП всегда является процессор (предполагается, что в рассматриваемой системе отсутствует механизм прямого доступа к памяти со стороны внешних устройств);
- кэширование ОП отсутствует.



Модуль ввода-вывода (МВВ) предназначен для реализации конкретной команды ввода/вывода на конкретном внешнем (по отношению к процессору) устройстве, подключенном к соответствующему порту (регистру) МВВ.

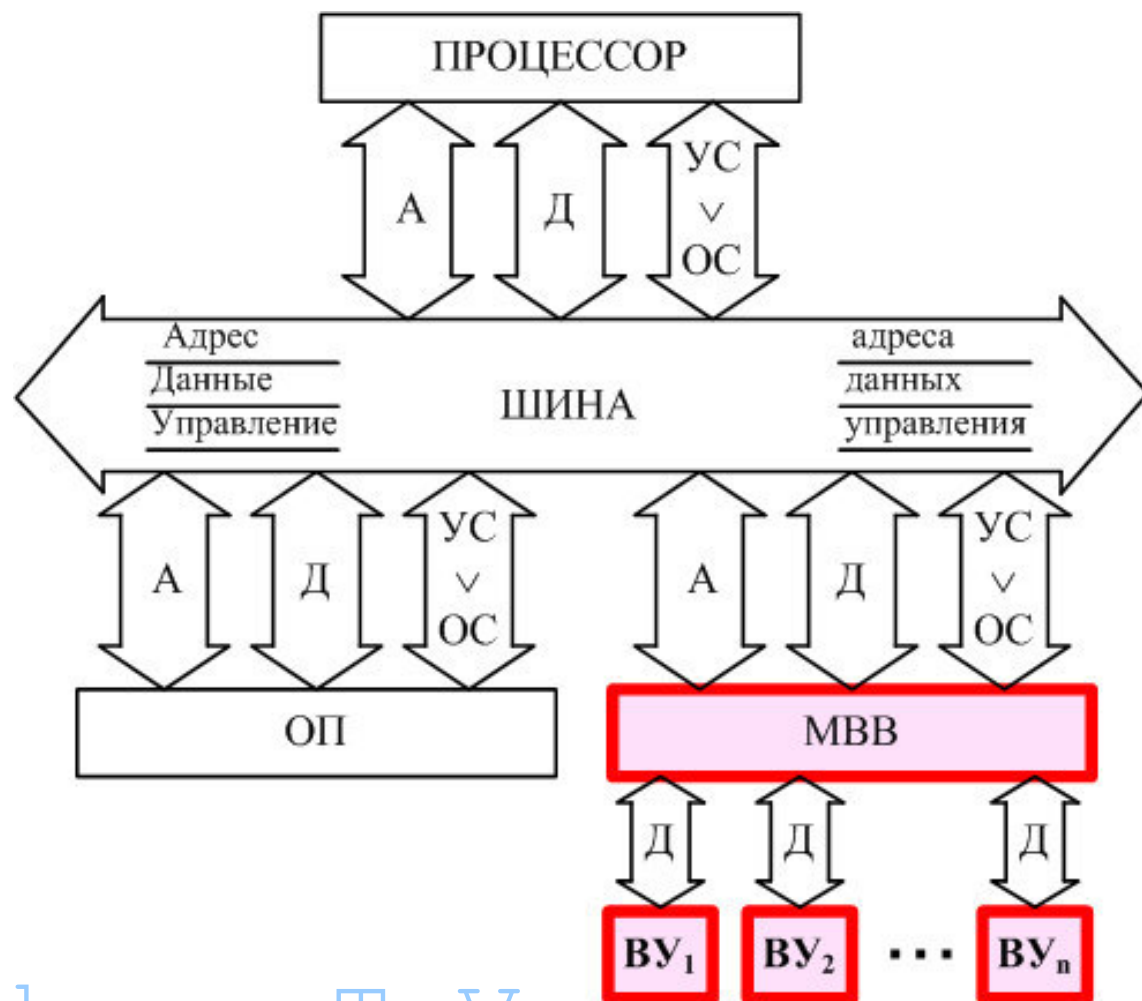
Ограничения задачи КП:

- организация ввода/вывода осуществляется по принципу программного управления, т.е. по инициативе процессора и под его полным контролем

(предполагается, что в рассматриваемой системе отсутствует механизм прямого доступа к памяти со стороны внешних устройств);

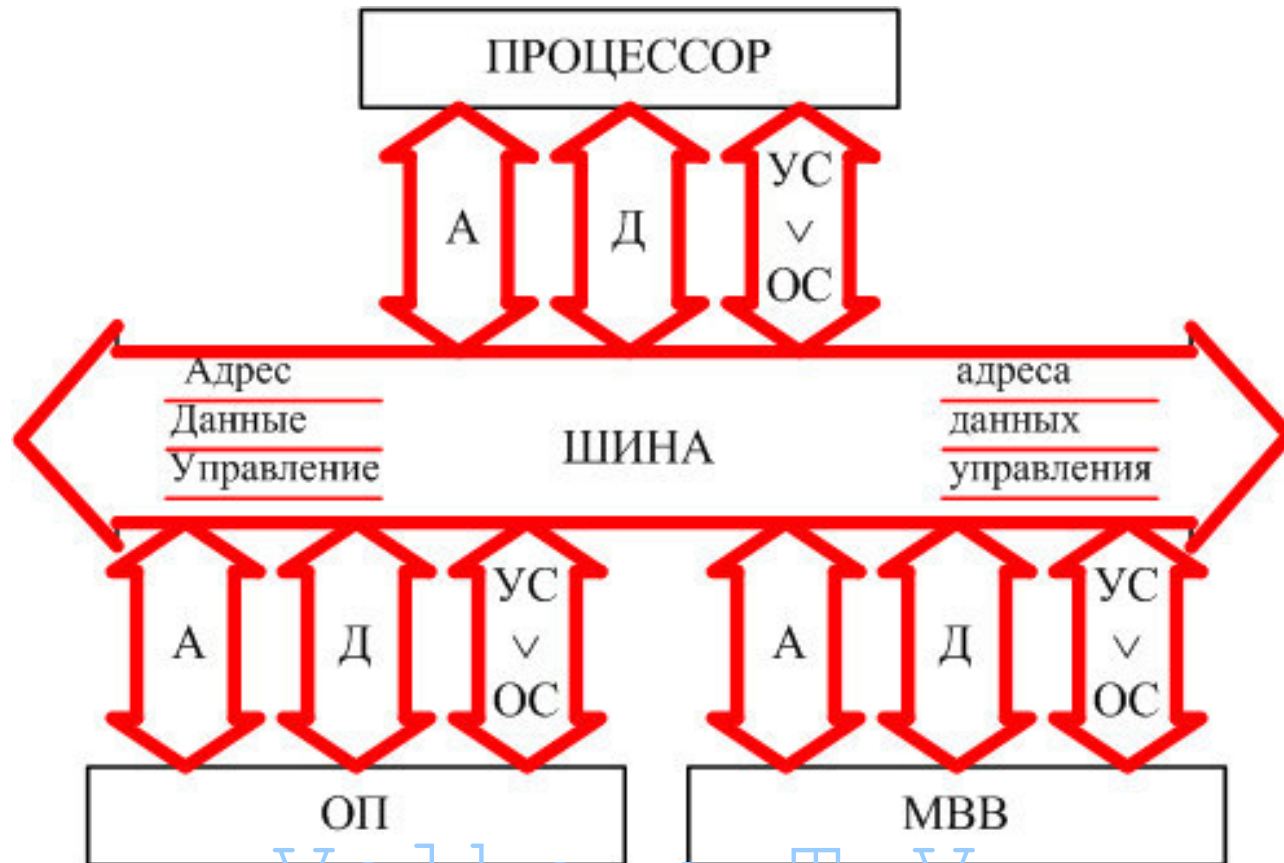
- предполагаем, что единица обмена данными между процессором и внешним устройством равна **1 байту**.

- предполагаем, что к МВВ можно подключить не более **256** внешних устройств (ВУ).

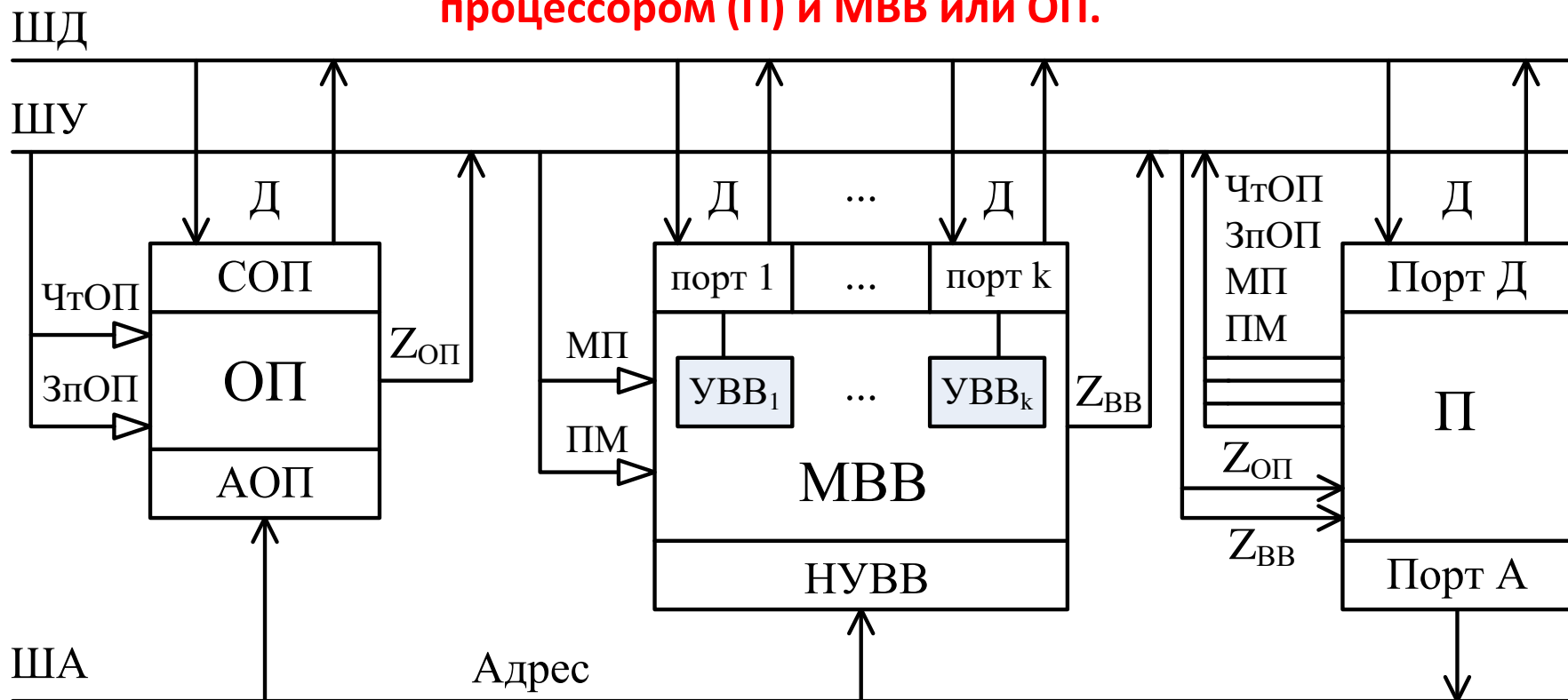


Подсистема компьютера, обеспечивающая информационный обмен между процессором, ОП и МВВ называется **системной шиной**. Совокупность цепей, по которым осуществляется обмен информационными сигналами, разделяется по функциональному назначению на три группы: **шину адреса**, **шину данных** и **шину управления**.

Разрядность каждой шины определяется максимальным количеством битов адреса, данных, управляющих или осведомительных слов, одновременно передающихся по цепям соответствующего назначения.



На рисунке 7.8 МУ_КП представлена схема обмена данными между процессором (П) и МВВ или ОП.



Используются следующие обозначения:

СОП – входы/выходы данных (слово ОП),

АОП – адресные входы ОП (адрес слова ОП),

порт 1 ... порт k – порты (регистры), через которые внешние устройства УВВ₁...УВВ_k обмениваются данными с процессором (П),

НУВВ – адресные входы МВВ (на них поступает номер (адрес) устройства ввода/вывода),

порт Д – порт процессора, через который он обменивается данными с внешними устройствами (ОП и МВВ),

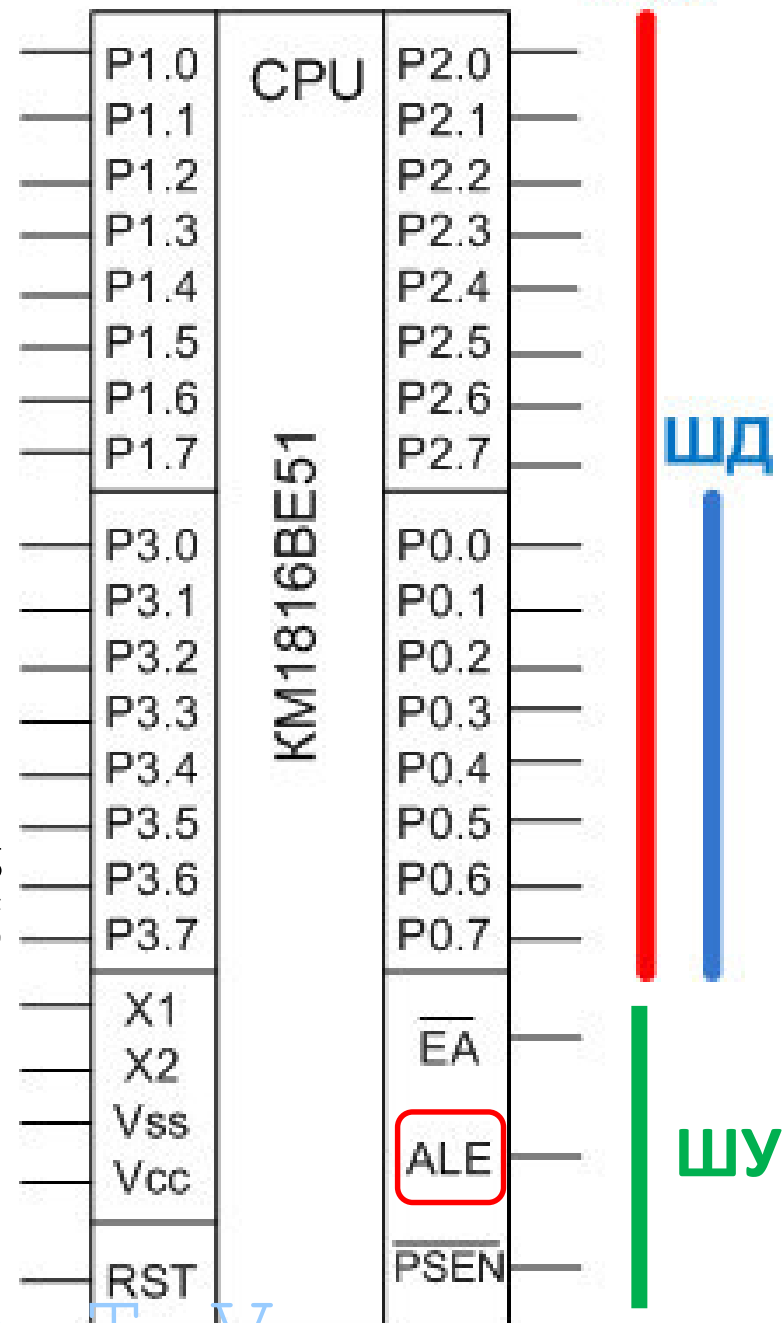
порт А – порт, через который процессор выдает адрес слова ОП или номер УВВ.

Примеры микросхем (из реальной жизни) для общего развития ;-)

Микропроцессор (микроконтроллер) KM1816BE51

16-разрядный адрес
выдается через два порта
P2 и P0. Для обмена
данными по шине данных
используется тот же порт
P0. В фазе выдачи адреса
на выходе ALE – уровень
логической единицы. В
фазе передачи (записи или
считывания) данных на
выходе ALE – уровень
логического нуля.

шу | \overline{WR}
 \overline{RD}



В КП для упрощения задачи не будем
совмещать линии адреса и линии данных
(выход ALE в этом случае не нужен).

Микросхемы памяти (ОЗУ)

RAM – Random Access Memory

КР 537 РУЗ:

12 адресных входов A0-A11;

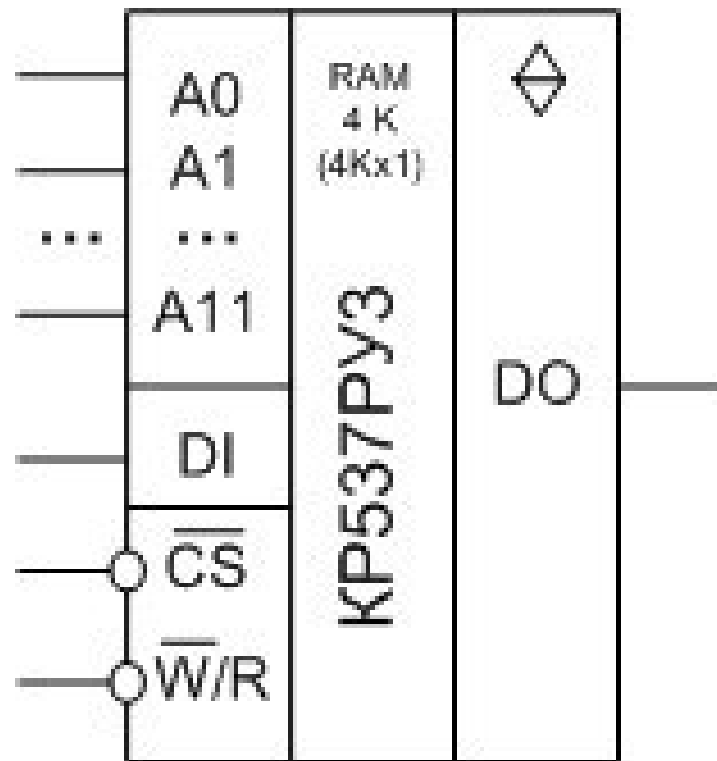
1 вход данных DI;

1 выход данных DO;

2 управляющих входа: \overline{CS} , $\overline{WR/RD}$

\overline{CS} – Chip Select – выбор кристалла (для создания многокристальных блоков памяти). Если на этом входе активный уровень (ноль) – микросхема выбрана и может выполнять чтение или запись.

$\overline{WR/RD}$ – режим записи, если ноль, режим чтения, если единица.



Блоки памяти из нескольких микросхем будем строить на лабораторной работе №2.

При помощи 12 битов адреса можно обращаться к

$$2^{12} = 2^2 \cdot 2^{10} = 4K = 4 \cdot 1024 = 4096 \text{ ячеек ОЗУ.}$$

Ячейки одноразрядные, поэтому объем ОЗУ записывают как 4К x 1

Микросхема ОЗУ КР537РУ10

интересна тем, что состоит из 8-битных ячеек, соответственно имеет 8 входов данных (при записи), которые вместе с тем являются и выходами данных (при чтении) – **двунаправленные выводы данных.**



11 адресных входов позволяют обращаться к

$$2^{11} = 2 \cdot 2^{10} = 2K = 2 \cdot 1024 = 2048$$

восьмиразрядных ячеек памяти.

Поэтому объем микросхемы записывают как 2K x 8 или 16K (в битах).

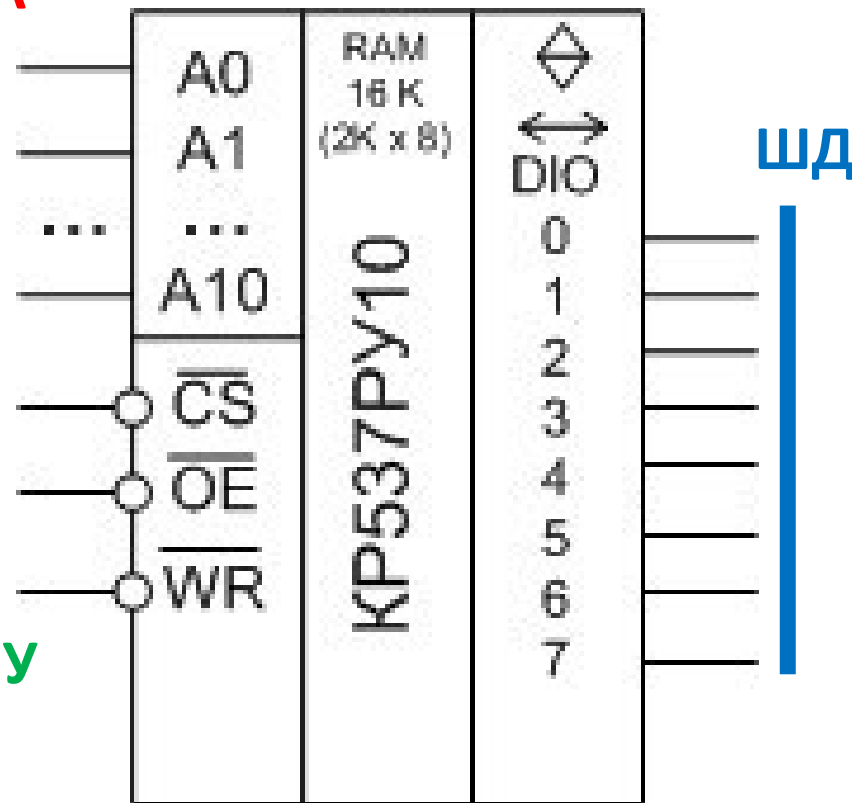
Добавлен один управляющий вход:

Если $\overline{WR} = 0$ – запись,

если $\overline{OE} = 0$ – чтение.

ША

ШУ

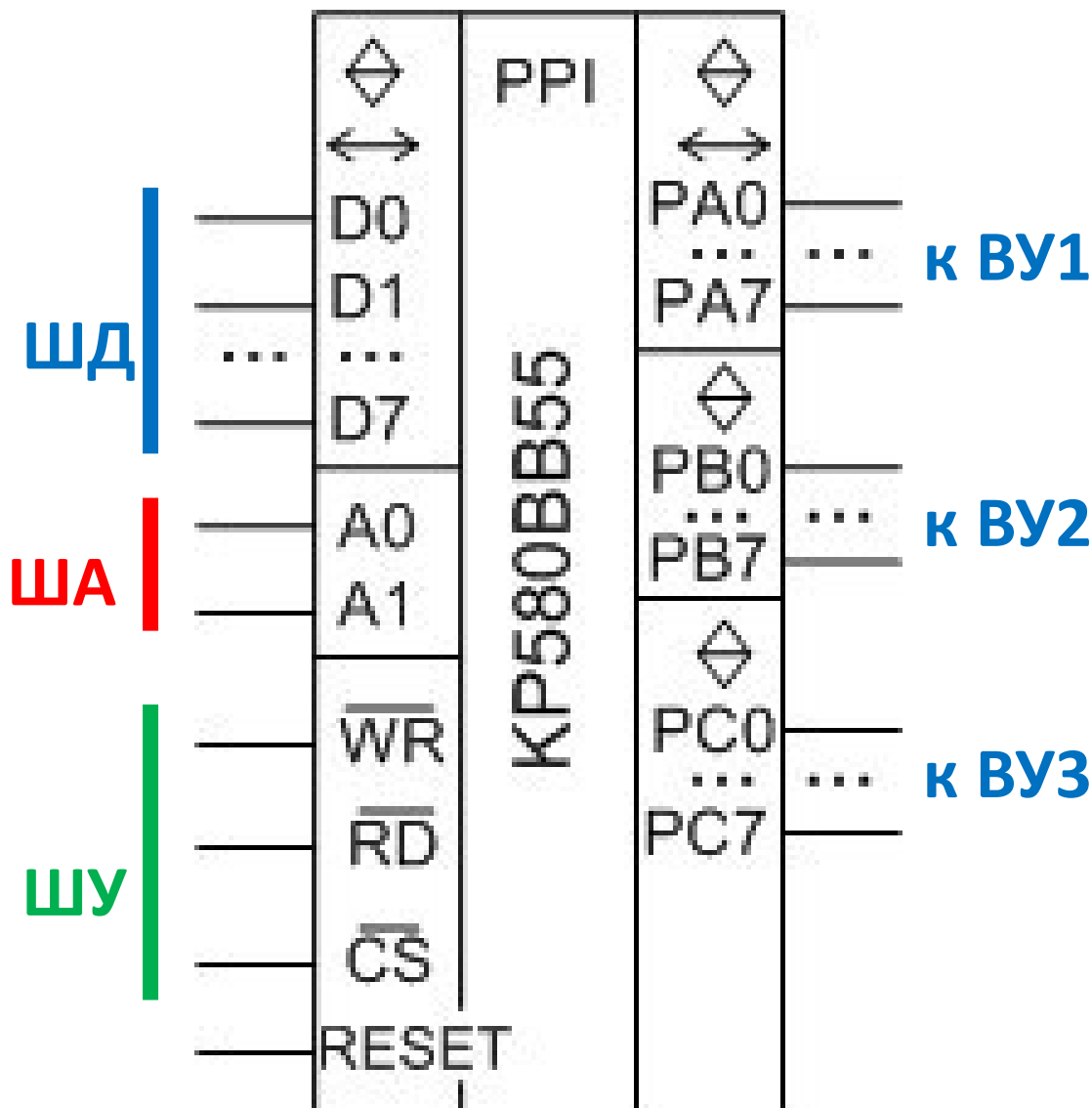


Так обозначаются выходы с тремя стабильными состояниями ("0", "1", высокое сопротивление).

Если $\overline{CS} = 1$, выходы в третьем состоянии и фактически оторваны от шины данных.

Модуль ввода - вывода.

Микросхема
КР580ВВ55
позволяет
подключить к
микропроцессору
через
восьмиразрядные
порты РА, РВ и РС
три внешних
устройства, которые
могут обмениваться
с
микроконтроллером
байтами.



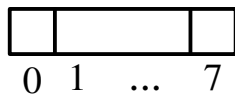
Итак, рассмотрели общую структуру проектируемой ЭВМ (ЦОУ) и привели примеры, подтверждающие, что постановка задачи имеет прямое отношение к тому, как в действительности проектируется вычислительная техника.

Переходим к рассмотрению **типов данных, обрабатываемых проектируемой ЭВМ.**

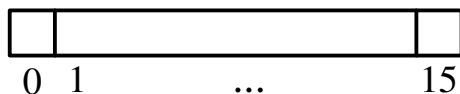
**Типы данных, которые
должны поддерживаться
проектируемой ЭВМ (ЦОУ).**

Типы данных, обрабатываемых ЦОУ: а) – основные структурные единицы данных, обрабатываемых проектируемым устройством;
б) – форматы представления данных в процессе обработки

Байт



Полуслово



Слово

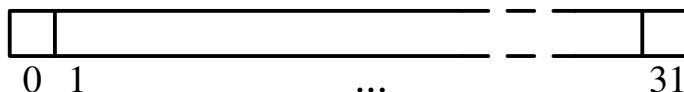
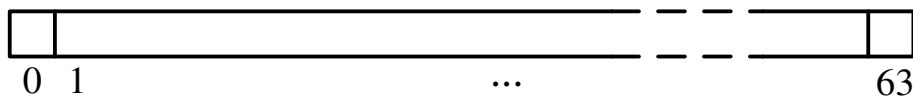


Рисунок 7.1 – МУ к КП

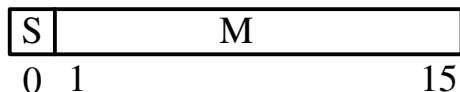
Базовая структурная единица

Двойное
слово

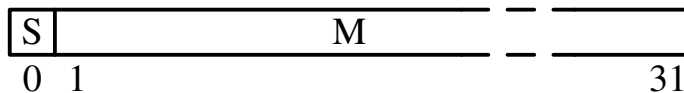


а)

Целые
числа

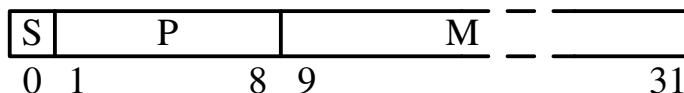


I2 Integer (длиной 2 байта)

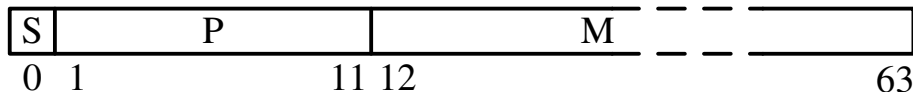


I4 Integer (длиной 4 байта)

Числа
С ПТ



F4 Float (длиной 4 байта)



F8 Float (длиной 8 байтов)

Целые числа могут быть представлены как со знаком (S) – форматы I2 и I4, так и без знака (для представления адресов).

Числа со знаком представляются в дополнительном коде.

Диапазон представления целых чисел – $[2^n, 2^{n+1}]$, где n – количество разрядов числа без учета разряда знака.

Двоичные числа с плавающей точкой представляются в виде двух чисел с фиксированной точкой: порядка (P) и мантиисы (M), при этом P – целое со знаком, $|M| < 1$ (правильная дробь).

Если используются нормализованная мантииса, то $\frac{1}{2} \leq |M| < 1$ для двоичной системы счисления. Диапазон представления чисел с плавающей точкой – $[M_{\min} \cdot 2^{P_{\min}}, M_{\max} \cdot 2^{P_{\max}}]$.

Двоичные вектора – наборы битов – форматы L1 (однобайтовый), L2 (двухбайтовый), L4 (четырёхбайтовый), L8 – (восьмибайтовый).

Для обеспечения достоверности выбираемых из ОП данных предлагается использовать принцип целочисленности границы для обрабатываемого типа данных.

Граница данного в ОП считается целочисленной, если ее адрес кратен числу байтов в данном.

Например, полуслово (данные длиной 2 байта) должно иметь адрес, кратный двум, слово (4 байта) – четырем, двойное слово (8 байтов) – восьми. Таким образом, если операнд имеет длину 2 байта (полуслово), то последний разряд у правильного адреса должен быть равен нулю. Это свидетельствует о том, что в памяти до выбираемого операнда уместится целое количество полуслов. Если операнд занимает в ОП 4 байта, то нулевыми должны быть два последних разряда его адреса (в памяти до выбираемого операнда уместится целое количество слов). Если операнд определяется как двойное слово, то его адрес должен оканчиваться тремя нулями (в памяти до выбираемого операнда уместится целое количество двойных слов).

Применение метода целочисленных границ обусловлено общим требованием высокой. Идея метода проиллюстрирована на рисунке 8.5 (МУ к КП).

**Рисунок 8.5 –
Расположение
информации в ОП
в соответствии с методом
целочисленных границ
(ПС – полуслово, С – слово,
ДС – двойное слово)**

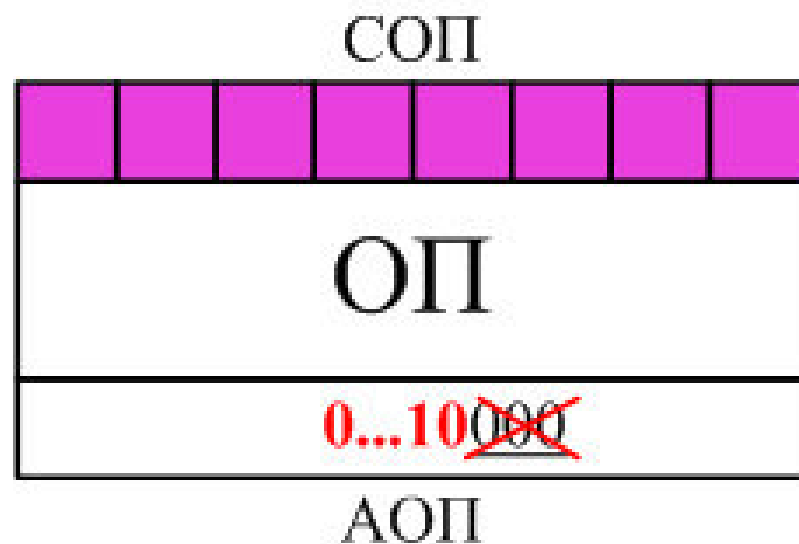
**Еще одно преимущество:
при использовании метода
целочисленных границ
можно гарантировать,
что для выборки данного,
число байтов в котором
равно ширине выборки,
потребуется одно
обращение к ОП.
Например, операнд
формата F8 при ширине
выборки ШВ=8 будет
целиком находиться в
двойном слове, выданном
оперативной памятью на
шину данных.**

Адрес байта	ОП	
	...	
0...10011		
0...1001 <u>0</u>		Граница ПС
0...10001		
0...10 <u>000</u>		Граница ДС, С, ПС
0...01111		
0...0111 <u>0</u>		Граница ПС
0...01101		
0...011 <u>00</u>		Граница С, ПС
0...01011		
0...0101 <u>0</u>		Граница ПС
0...01001		
0...01 <u>000</u>		Граница ДС, С, ПС
0...00111		
0...0011 <u>0</u>		Граница ПС
0...00101		
0...001 <u>00</u>		Граница С, ПС
0...00011		
0...0001 <u>0</u>		Граница ПС
0...00001		
0...00 <u>000</u>	0	7 Граница ДС, С, ПС

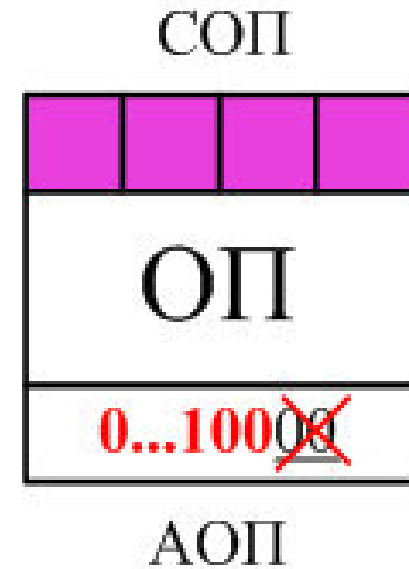
0...11 <u>000</u>	...	Граница ДС, С, ПС
0...10111		<i>Граница выборки из ОП</i>
0...1011 <u>0</u>		Граница ПС
0...10101		
0...101 <u>00</u>		Граница С, ПС
0...10011		
0...1001 <u>0</u>		Граница ПС
0...10001		
0...10<u>000</u>		Граница ДС, С, ПС
0...01111		<i>Граница выборки из ОП</i>
0...0111 <u>0</u>		Граница ПС
0...01101		
0...011 <u>00</u>		Граница С, ПС
0...01011		
0...0101 <u>0</u>		Граница ПС
0...01001		
0...010 <u>00</u>		Граница ДС, С, ПС
0...00111		<i>Граница выборки из ОП</i>
0...0011 <u>0</u>		Граница ПС
0...00101		
0...001 <u>00</u>		Граница С, ПС
0...00011		
0...0001 <u>0</u>		Граница ПС
0...00001		
0...000 <u>00</u>		Граница ДС, С, ПС

При соблюдении метода целочисленных границ, операнд выбирается за минимальное число обращений к ОП:

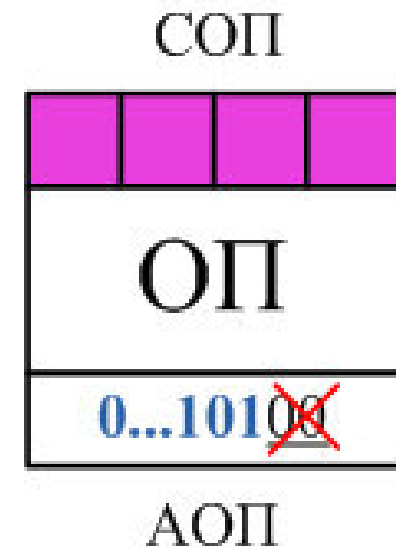
например, данное формата F8 или L8 при ШВ=8 (байт) будет выбрано за одно обращение к ОП, при ШВ=4 (байта) – за 2 обращения к ОП.



0...11000	...	Граница ДС, С, ПС
0...10111		
0...10110		Граница ПС
0...10101		
<u>0...10100</u>		Граница С, ПС
<hr/>		
0...10011		<i>Граница выборки из ОП</i>
0...10010		Граница ПС
0...10001		
<u>0...10000</u>		Граница ДС, С, ПС
<hr/>		
0...01111		<i>Граница выборки из ОП</i>
0...01110		Граница ПС
0...01101		
<u>0...01100</u>		Граница С, ПС
<hr/>		
0...01011		<i>Граница выборки из ОП</i>
0...01010		Граница ПС
0...01001		
<u>0...01000</u>		Граница ДС, С, ПС
<hr/>		
0...00111		<i>Граница выборки из ОП</i>
0...00110		Граница ПС
0...00101		
<u>0...00100</u>		Граница С, ПС
<hr/>		
0...00011		<i>Граница выборки из ОП</i>
0...00010		Граница ПС
0...00001		
<u>0...00000</u>		Граница ДС, С, ПС



Второе обращение к ОП:



0...11 <u>000</u>	...	Граница ДС, С, ПС
0...10111		<i>Граница выборки из ОП</i>
0...10110		Граница ПС
0...10101		
0...101 <u>00</u>		Граница С, ПС
0...10011		
0...10010		Граница ПС
0...10001		
0...10<u>000</u>		Граница ДС, С, ПС
0...01111		<i>Граница выборки из ОП</i>
0...01110		Граница ПС
0...01101		
0...011 <u>00</u>		Граница С, ПС
0...01011		
0...01010		Граница ПС
0...01001		
0...010 <u>00</u>		Граница ДС, С, ПС
0...00111		<i>Граница выборки из ОП</i>
0...00110		Граница ПС
0...00101		
0...001 <u>00</u>		Граница С, ПС
0...00011		
0...00010		Граница ПС
0...00001		
0...000 <u>00</u>		Граница ДС, С, ПС

Если метод целочисленных границ не используется, возникает две проблемы:

- 1) Данное нельзя выбрать за минимальное количество обращений к памяти (здесь потребуется два обращения вместо одного);
- 2) Невозможно осуществить проверку корректности выбираемого данного, т.е. определить, что выбрано: одно число формата F8 или 2 куска из разных чисел (если программист ошибся и неправильно задал адрес данного).

Фактически, проектировщик передает ответственность за корректность выбираемых данных программисту, но себе тоже усложняет жизнь вместе с усложнением проектирования микропрограммы выборки данных ;-)

Исходя из вышесказанного, мы будем использовать при проектировании ЦОУ метод целочисленных границ , а, значит,

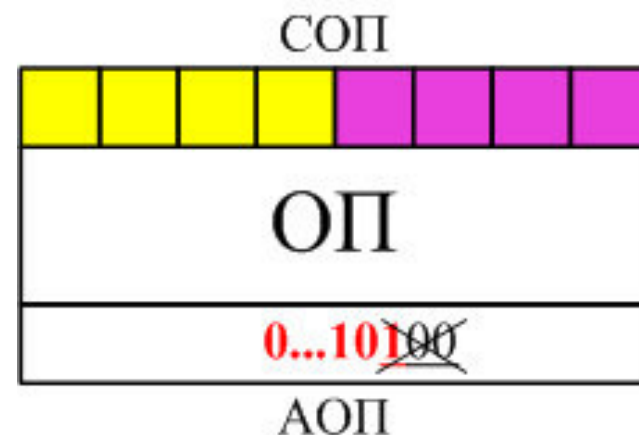
в микропрограмме выборки данного из ОП будем проверять, чтобы адрес данных был кратен числу байтов в данном, т.е. заканчивается на соответствующее число нулей.

Ситуация обнаружения некорректного адреса операнда будет отмечаться установкой в единицу соответствующего бита в регистре флагов прерываний.

Назовем эту исключительную ситуацию – нарушение спецификации.

Аналогично покажем применение метода целочисленных границ для данных, длиной в слово (I4, F4, L4) при ШВ=8:

- 1) Ниже нужного операнда (адрес – 0...10100) находится целое число слов.
- 2) Операнд выбирается за одно обращение к ОП, т.к. является частью двойного слова выборки.



Вопрос: как процессор определит, операнд является первым словом в двойном слове выборки или вторым?

0...11000	...	Граница ДС, С, ПС
0...10111		<i>Граница выборки из ОП</i>
0...10110		Граница ПС
0...10101		
0...10100		Граница С, ПС
0...10011		
0...10010		Граница ПС
0...10001		
0...10000		Граница ДС, С, ПС
0...01111		<i>Граница выборки из ОП</i>
0...01110		Граница ПС
0...01101		
0...01100		Граница С, ПС
0...01011		
0...01010		Граница ПС
0...01001		
0...01000		Граница ДС, С, ПС
0...00111		<i>Граница выборки из ОП</i>
0...00110		Граница ПС
0...00101		
0...00100		Граница С, ПС
0...00011		
0...00010		Граница ПС
0...00001		
0...00000		Граница ДС, С, ПС

Эти биты у
двух слов,
входящих в
двойное слово
выборки –
разные!

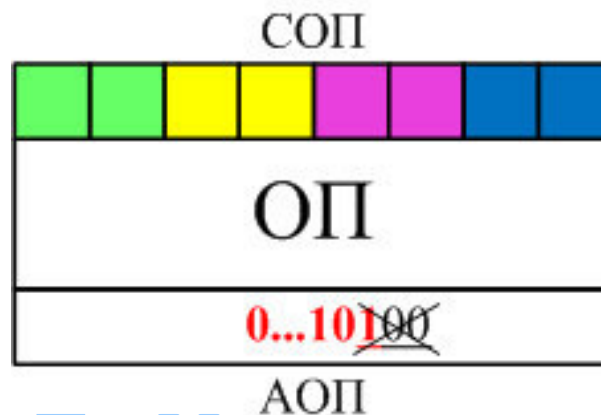
0...11 <u>000</u>	...	Граница ДС, С, ПС
0...10111		<i>Граница выборки из ОП</i>
0...1011 <u>0</u>		Граница ПС
0...10101		
0...101 <u>00</u>		Граница С, ПС
0...10011		
0...1001 <u>0</u>		Граница ПС
0...10001		
0...100 <u>00</u>		Граница ДС, С, ПС
0...01111		<i>Граница выборки из ОП</i>
0...0111 <u>0</u>		Граница ПС
0...01101		
0...011 <u>00</u>		Граница С, ПС
0...01011		
0...0101 <u>0</u>		Граница ПС
0...01001		
0...010 <u>00</u>		Граница ДС, С, ПС
0...00111		<i>Граница выборки из ОП</i>
0...0011 <u>0</u>		Граница ПС
0...00101		
0...001 <u>00</u>		Граница С, ПС
0...00011		
0...0001 <u>0</u>		Граница ПС
0...00001		
0...000 <u>00</u>		Граница ДС, С, ПС

0...11 <u>000</u>	...	Граница ДС, С, ПС
0...10111		<i>Граница выборки из ОП</i>
0...1011 <u>0</u>		Граница ПС
0...10101		
0...101<u>00</u>		Граница С, ПС
0...10011		
0...1001 <u>0</u>		Граница ПС
0...10001		
0...100 <u>000</u>		Граница ДС, С, ПС
0...01111		<i>Граница выборки из ОП</i>
0...0111 <u>0</u>		Граница ПС
0...01101		
0...011 <u>00</u>		Граница С, ПС
0...01011		
0...0101 <u>0</u>		Граница ПС
0...01001		
0...010 <u>000</u>		Граница ДС, С, ПС
0...00111		<i>Граница выборки из ОП</i>
0...0011 <u>0</u>		Граница ПС
0...00101		
0...001 <u>00</u>		Граница С, ПС
0...00011		
0...0001 <u>0</u>		Граница ПС
0...00001		
0...000 <u>000</u>		Граница ДС, С, ПС

Аналогично покажем применение метода целочисленных границ для данных, длиной в полуслово (при ШВ=8):

- 1) Ниже нужного операнда (адрес – 0...10100) находится целое число полуслов.
- 2) Операнд выбирается за одно обращение к ОП, т.к. является частью двойного слова выборки.

Вопрос: как процессор определит, какое из четырех полуслов, находящихся в слове выборки, является нужным операндом?



Эти биты у
четырех
полуслов,
входящих в
двойное
слово
выборки –
разные!

0...11 <u>000</u>	...	Граница ДС, С, ПС
0...10111		<i>Граница выборки из ОП</i>
0...101 <u>10</u>		Граница ПС
0...10101		
0...101 <u>00</u>		Граница С, ПС
0...10011		
0...100 <u>10</u>		Граница ПС
0...10001		
0...100 <u>00</u>		Граница ДС, С, ПС
0...01111		<i>Граница выборки из ОП</i>
0...0111 <u>0</u>		Граница ПС
0...01101		
0...011 <u>00</u>		Граница С, ПС
0...01011		
0...0101 <u>0</u>		Граница ПС
0...01001		
0...010 <u>00</u>		Граница ДС, С, ПС
0...00111		<i>Граница выборки из ОП</i>
0...0011 <u>0</u>		Граница ПС
0...00101		
0...001 <u>00</u>		Граница С, ПС
0...00011		
0...0001 <u>0</u>		Граница ПС
0...00001		
0...000 <u>00</u>		Граница ДС, С, ПС

Система команд процессора проектируемой ЭВМ (ЦОУ)

Система команд ЦОУ

Чтобы система команд ЦОУ была функционально полной, она должна включать следующие **классы команд**:

- 1) команды обращения к памяти по чтению и записи;
- 2) арифметические команды над целыми числами со знаком и без знака (сложение, вычитание, умножение, деление, сравнение);
- 3) арифметические команды над числами с плавающей точкой (сложение, вычитание, умножение, деление, сравнение);
- 4) логические команды (поразрядное «И», «ИЛИ», «Исключающее ИЛИ», инверсия),
- 5) команды арифметических и логических сдвигов (операндов одинарной (L4) и двойной (L8) длины) на заданное число разрядов (константа сдвига);
- 6) команды передачи управления (условных и безусловных переходов);
- 7) команды ввода/вывода;
- 0) специальные команды (HALT - останов).

Считаем, что **полная система команд создается всей группой**, при этом каждый **студент группы проектирует свое подмножество команд**, в которое входят:

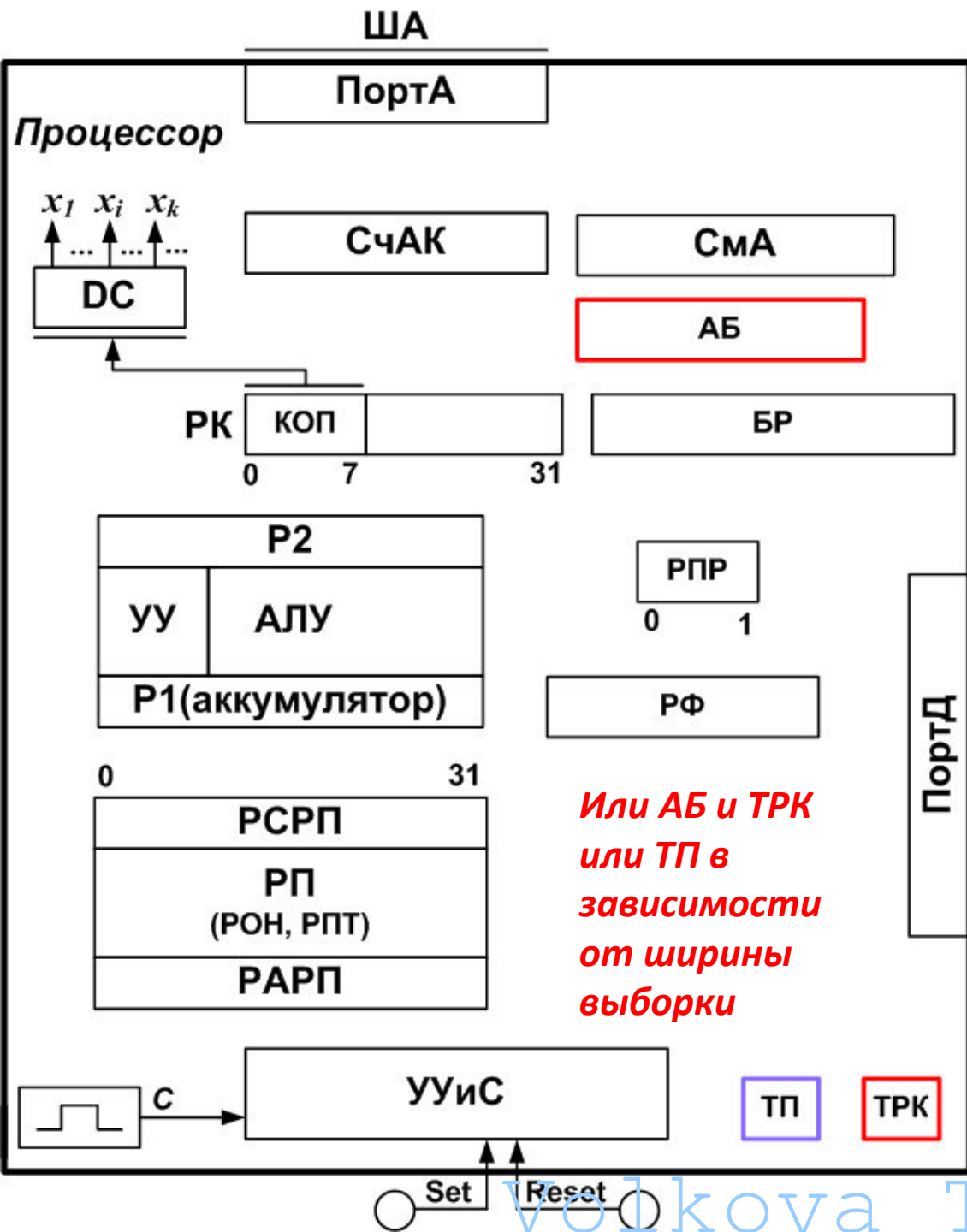
- 1) одна арифметическая команда;
- 2) одна логическая команда;
- 3) одна команда передачи управления (перехода);
- 4) одна команда пересылки информации;
- 5) одна команда ввода-вывода;
- 6) команда останова.

Вариант задания на КП определяет:

- 1) операцию, которую иницирует каждая команда;
- 2) типы данных, с которыми работает команда;
- 3) способы адресации операндов, которые может использовать студент при разработке форматов команд.

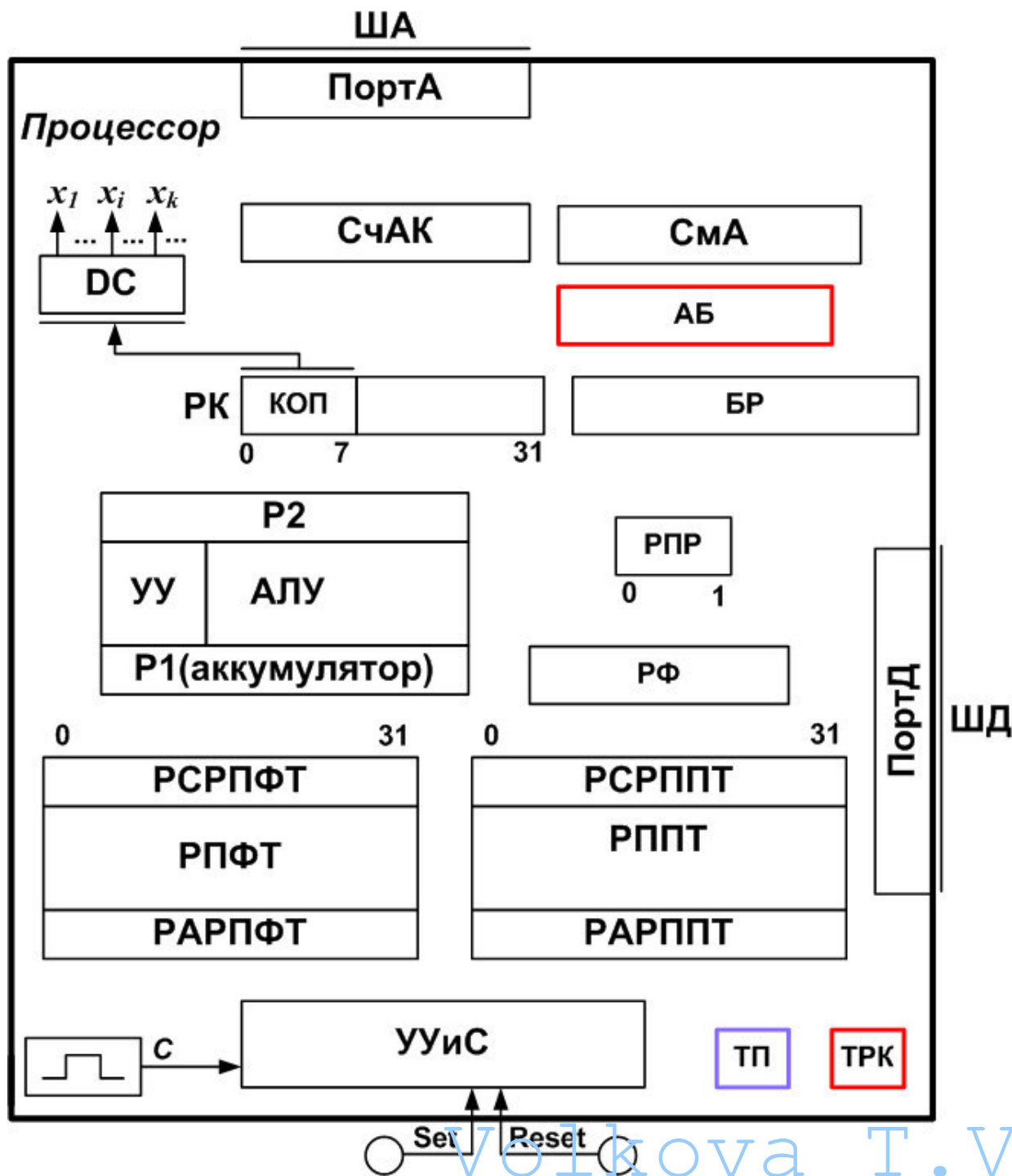
№ вар	Команды АЛУ				Дополнит. команды			ОП		РП		Способы адресации					
	КА	ДА	КЛ	ДЛ	ПУ	ОбрП	ВВ	Е _{ОП}	ШВ	Е _{РП}	Тип	Н	П	К	Р	КР	О
1	+	I2	✓	L1	УП _М	ЗР F4	ПМ	128	4	16	У	+	+	+	+	+	
2	−	F4	∧	L4	ПСч	ЗП F8	МП	256	4	8	Р	+	+	+	+		+

Структура процессора проектируемой ЭВМ (ЦОУ)



Упрощенная структура процессора, если задан тип регистровой памяти (РП) – У (универсальная): данные с фиксированной точкой и данные с плавающей точкой хранятся в одном и том же устройстве РП.

Для всех вариантов будет одинаковой разрядность РК (32 бита – максимальная длина команды), в том числе, поля КОП (8 битов, чтобы закодировать 256 команд – по условию задачи); РСРП (32 бита – размер слова, принятый в проектируемой системе); РПР (2 бита кодируют 4 возможных значения признака результата). **Разрядность остальных элементов должна быть вычислена!**



Упрощенная структура процессора, если задан тип регистровой памяти (РП) – Р (раздельная): данные с фиксированной точкой и данные с плавающей точкой хранятся в двух разных устройствах РП.

Если разрядность элемента структурной схемы не указана, значит она должна быть вычислена в соответствии с данными варианта задания!

№ вар	Команды АЛУ				Дополнит. команды			ОП		РП		Способы адресации					
	КА	ДА	КЛ	ДЛ	ПУ	ОбрП	ВВ	Е _{ОП}	ШВ	Е _{РП}	Тип	Н	П	К	Р	КР	О
1	+	I2	✓	L1	УП _М	ЗР F4	ПМ	128	4	16	У	+	+	+	+	+	
2	–	F4	∧	L4	ПСч	ЗП F8	МП	256	4	8	Р	+	+	+	+		+

Эти параметры
влияют на
разрядность
шины данных и
элементов, в
которые
загружаются
данные

Эти параметры
влияют на
разрядность
адресных элементов

Эти параметры
влияют на
разрядность
регистра команд (за
счет использования
способов адресации
нам удастся
сократить
максимальную
длину команды до 4
байтов).

- 1) УУиС (вырабатывает управляющие сигналы, т.е. управляет работой всех других устройств);
- 2) СчАК (длина определяется количеством битов адреса в системе, зависящим от объема оперативной памяти в байтах);
- 3) СмА – сумматор адреса (используется для вычисления адреса операнда при относительной адресации; на 1 бит длиннее, чем СчАК, чтобы выявлять ситуацию переполнения при сложении);
- 4) РК (длина определяется максимальной длиной команды);
- 5) АЛУ, имеющее два регистра для операндов, один из которых является аккумулятором (длина регистров определяется максимальной длиной операнда) и выполняющее требуемый (определяется разработчиком) набор арифметических и логических операций;

Регистры специальных функций (РСФ)

- 6) РПР (двухразрядный);
- 7) РФ (разрядность определяется количеством возможных прерываний в системе);
- 8) ПортА (разрядность определяется количеством битов (линий) шины адреса, которое зависит от объема оперативной памяти $E_{оп}$ и ширины выборки из оперативной памяти ШВ).
- 9) ПортД (длина равна количеству битов (линий) шины данных, которое, в свою очередь, определяется максимальной длиной операнда).

Регистры для хранения данных (РОН и РПТ)

- 10) Регистровая память, состоящая из ограниченного количества РОН (регистры общего назначения для хранения данных с фиксированной точкой) и РПТ (регистры для хранения данных с плавающей точкой), универсальная или раздельная; разрядность регистра адреса регистровой памяти (РАРП) определяется исходя из числа регистров, разрядность РСРП определяется размером слова, принятым в системе (у нас – 32 бита).

команды

Ширина выборки (4 или 8 байт) больше минимальной длины команды (2 байта), поэтому за один раз из памяти может быть считано не одна, а несколько команд.

При ШВ=4 байта для сокращения числа обращений к ОП используется:

- ✓ 16-разрядный буферный регистр **БР**, в котором сохраняется неиспользованная часть считанной из ОП (на этапе выборки команды) информация (следующая команда или часть следующей команды);
- ✓ **ТП** – триггер перехода, с помощью которого осуществляется управление процессом извлечения следующей команды. Если ТП=0 (предыдущая команда не являлась командой перехода), следующую команду надо брать из БР, иначе – обращаться за ней в ОП.

При ШВ=8 байт для сокращения числа обращений к ОП используется:

- ✓ 64-разрядный буферный регистр **БР** (длина равна ширине выборки), в котором сохраняется вся считанная на этапе выборки команды информация – 8 байт (там может содержаться максимум 4 команды);
- ✓ адресный регистр **АБ** (хранит адрес восьмибайтного слова, которое считано на БР),
- ✓ **ТРК** – триггер команды, с помощью которого осуществляется управление процессом записи очередной команды на РК.

Подробнее использование этих элементов структурной схемы процессора будет рассмотрено позже при обсуждении микропрограммы выборки команды.

ОПРЕДЕЛЕНИЕ РАЗРЯДНОСТИ АДРЕСНЫХ ЭЛЕМЕНТОВ

Число разрядов, необходимое для задания адреса байта ОП (N_A), можно вычислить по формуле:

$$N_A = \log_2 E_{оп} ,$$

где $E_{оп}$ – объем ОП в байтах.

Например, при $E_{оп} = 256 \text{ М байт}$,

$$N_A = \log_2 256 \text{ М} = \log_2 (2^8 \cdot 2^{20}) = \log_2 2^{28} = 28$$

Для объемов 127М и 512М вычислить самостоятельно.

Адрес

0		n-3	n-2	n-1	n
Адрес байта					
Адрес полуслова					
Адрес слова					
Адрес двойного слова					

$$n + 1 = N_A$$

Приведем пример расчета для $E_{оп} = 256 \text{ М байт}$.

$$N_A = \log_2 256 \text{ М} = \log_2 (2^8 \cdot 2^{20}) = \log_2 2^{28} = 28$$

Адрес

0		24	25	26	27
Адрес байта					
Адрес полуслова					
Адрес слова					
Адрес двойного слова					

Полуслов в ОП в 2 раза меньше, чем байтов, слов в 2 раза меньше, чем полуслов, двойных слов в 2 раза меньше, чем слов.

Покажем, что адрес двойного слова в данном примере действительно будет 25-разрядным, т.е. в нем должно быть на 3 разряда меньше, чем в полном адресе (адресе байта).

Двойных слов в 8 раз меньше, чем байтов, значит,

$$N_{A_{двсл}} = \log_2 (2^{28} : 2^3) = \log_2 2^{25} = 25$$

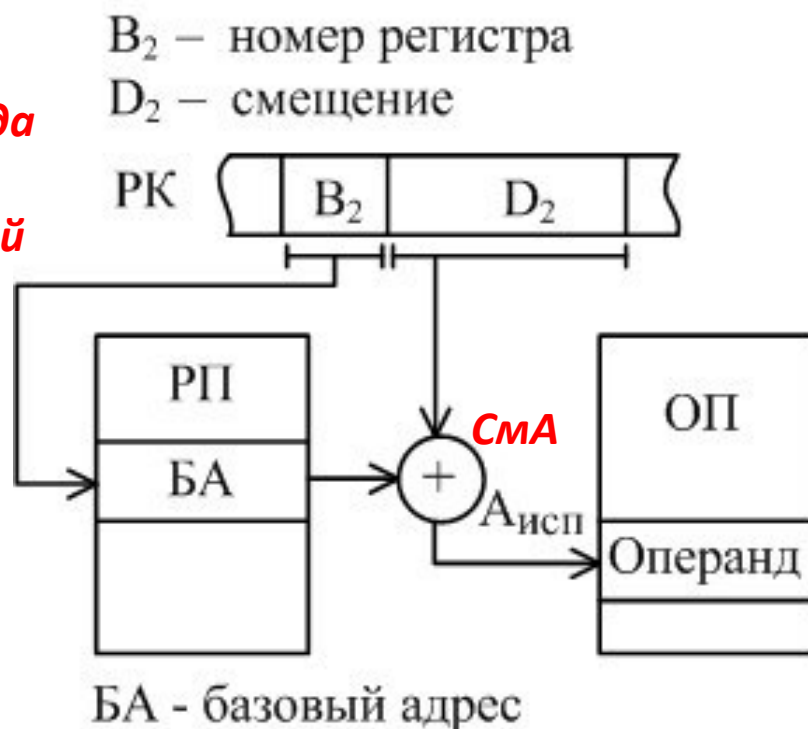
1) Длина (разрядность) СчАК равна числу разрядов в адресе байта.



2) Длина СмА на 1 больше.



*Схема
вычисления
адреса операнда
при
относительной
адресации*



Если программист неправильно задал базовый адрес в регистре с номером B_2 (предыдущей командой пересылки информации) и/или смещение (в данной команде), то при выполнении сложения может возникнуть переполнение в СмА. Для обнаружения этой ситуации (попытка обращения в несуществующую память!) мы и добавляем лишний (старший) бит в СмА. Если он равен 1, значит возникла исключительная ситуация **«нарушение адресации»**: адрес превысил максимально допустимое значение).

3) Число разрядов шины адреса (ША).

Зависит от ширины выборки.

Расчет при ШВ= 4.

Поскольку выборка из ОП осуществляется 4-байтными словами, число разрядов ША равно числу разрядов в адресе слова (для нашего примера – 26).

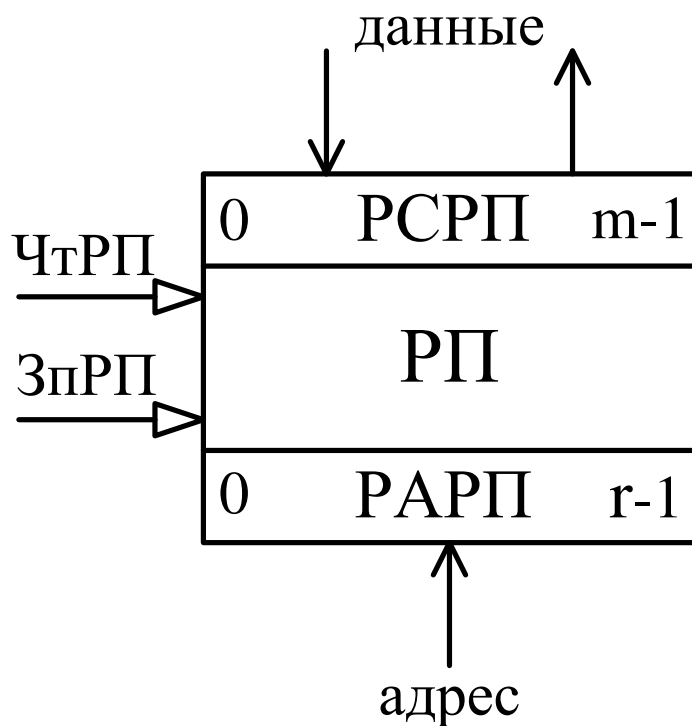
Расчет при ШВ=8.

Поскольку выборка из ОП осуществляется двойными (8-байтными) словами, число разрядов ША равно числу разрядов в адресе двойного слова (для нашего примера – 25).

4) Длина порта адреса (ПортА) равна числу разрядов шины данных.

5) Длина адресного регистра АБ (используется только, если ширина выборки равна 8 байтам, на этапе выборки команды запоминает адрес двойного слова, считанного из ОП на буферный регистр БР) равна длине ПортА.

Регистровая память (РП) организуется как сверхоперативная память с прямой адресацией. РП допускает выборку слова в течение одного такта. По типу РП определяется как общая (для данных в формате с фиксированной точкой и с плавающей точкой) или раздельная, состоящая из двух самостоятельных блоков (регистры общего назначения (РОН) и регистры данных формата с плавающей точкой (РПТ)). Условное обозначение блока РП показано на рисунке 7.6.



Разрядность **m** регистра слова регистровой памяти (РСРП) определяется длиной информационного слова. Для проектируемого устройства **m=32**.

Разрядность **r** регистра адреса регистровой памяти (РАРП) определяется как

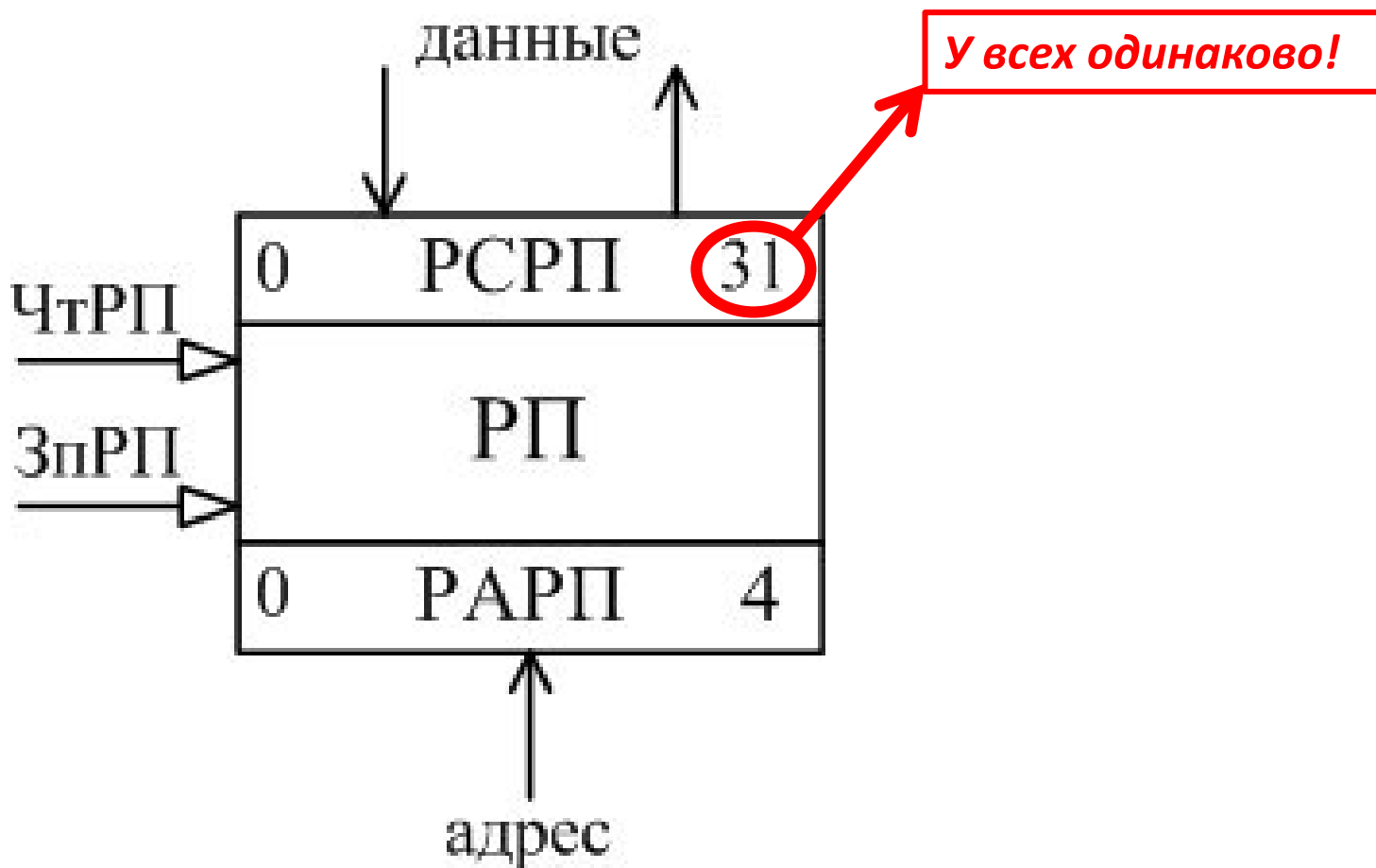
$$r = \lceil \log_2 E_{РП} \rceil$$

где $E_{РП}$ – количество регистров в блоке РП.

Примечание. Если по варианту регистровая память раздельная, то каждое из устройств регистровой памяти (РПФТ и РППТ) содержит $E_{РП}$ регистров, заданное вариантом на курсовое проектирование.

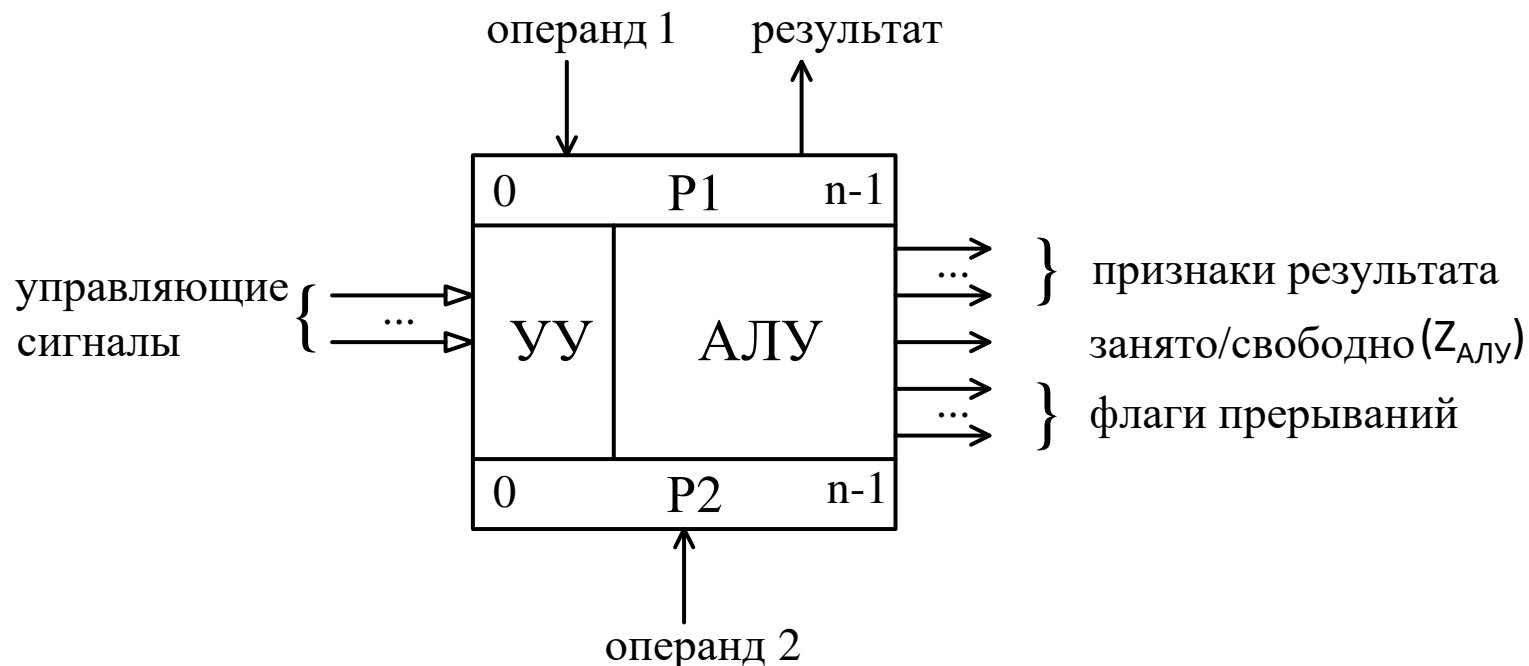
Если вариантом задания предусмотрено $E_{\text{РП}}=32$, то РАРП должен содержать 5 разрядов.

На структурной схеме регистровая память будет выглядеть так:

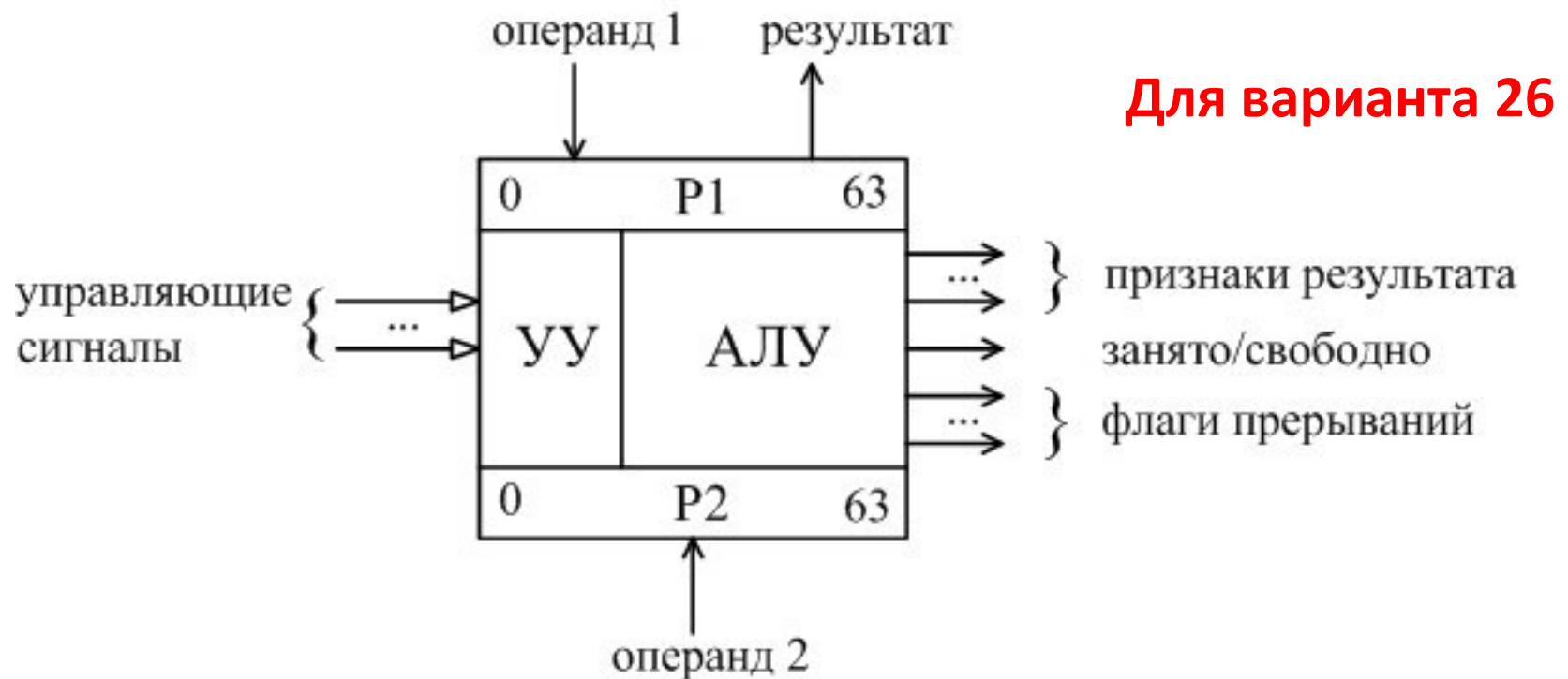


Главным элементом процессорного блока является АЛУ. Условное обозначение АЛУ показано на рисунке 7.5.

Двуместные операции в АЛУ выполняются над операндами, хранящимися в регистрах $P1(0:n-1)$ (первый операнд) и $P2(0:n-1)$ (второй операнд). Результат выполненной операции помещается в $P1$. Разрядность регистров n выбирается равной максимальной длине операндов. Каждая операция АЛУ иницируется своим управляющим сигналом, поступающим на вход устройства управления (УУ). АЛУ вырабатывает три группы осведомительных сигналов: признаки результата, флаги прерываний, признак занятости.

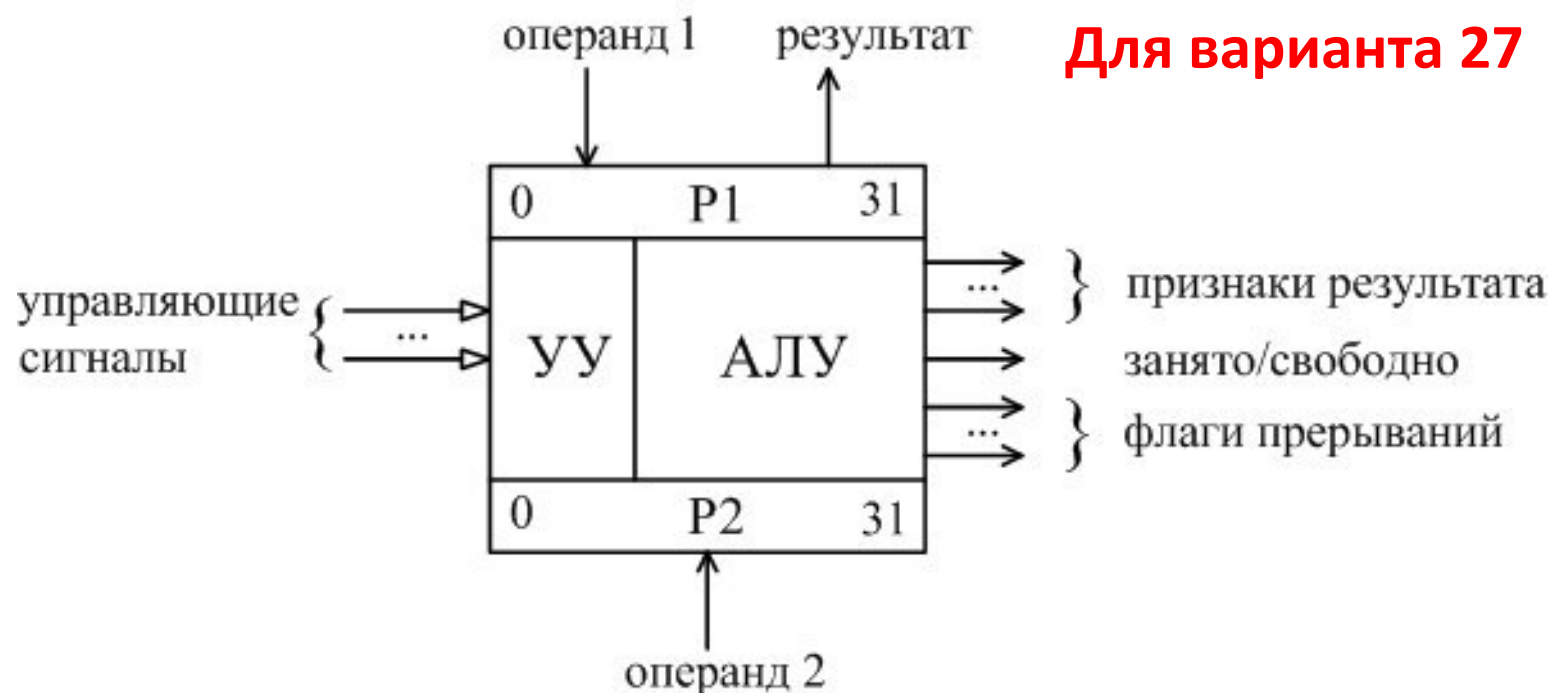


№ вар	Команды АЛУ				Дополнит. команды			ОП		РП		Способы адресации					
	КА	ДА	КЛ	ДЛ	ПУ	ОбрП	ВВ	Е _{ОП}	ШВ	Е _{РП}	Тип	Н	П	К	Р	КР	О
26	+	F8	∧	L4	ПИ<	ЗР I2	МП	512	8	8	Р	+	+	+	+		+
27	×	I2	⊕	L4	УП _М	ЗП F4	ПМ	256	8	16	Р	+	+	+		+	+
28	*	I4	√	L1	ПСч	ЗР I2	МП	128	8	32	У	+	+		+	+	+

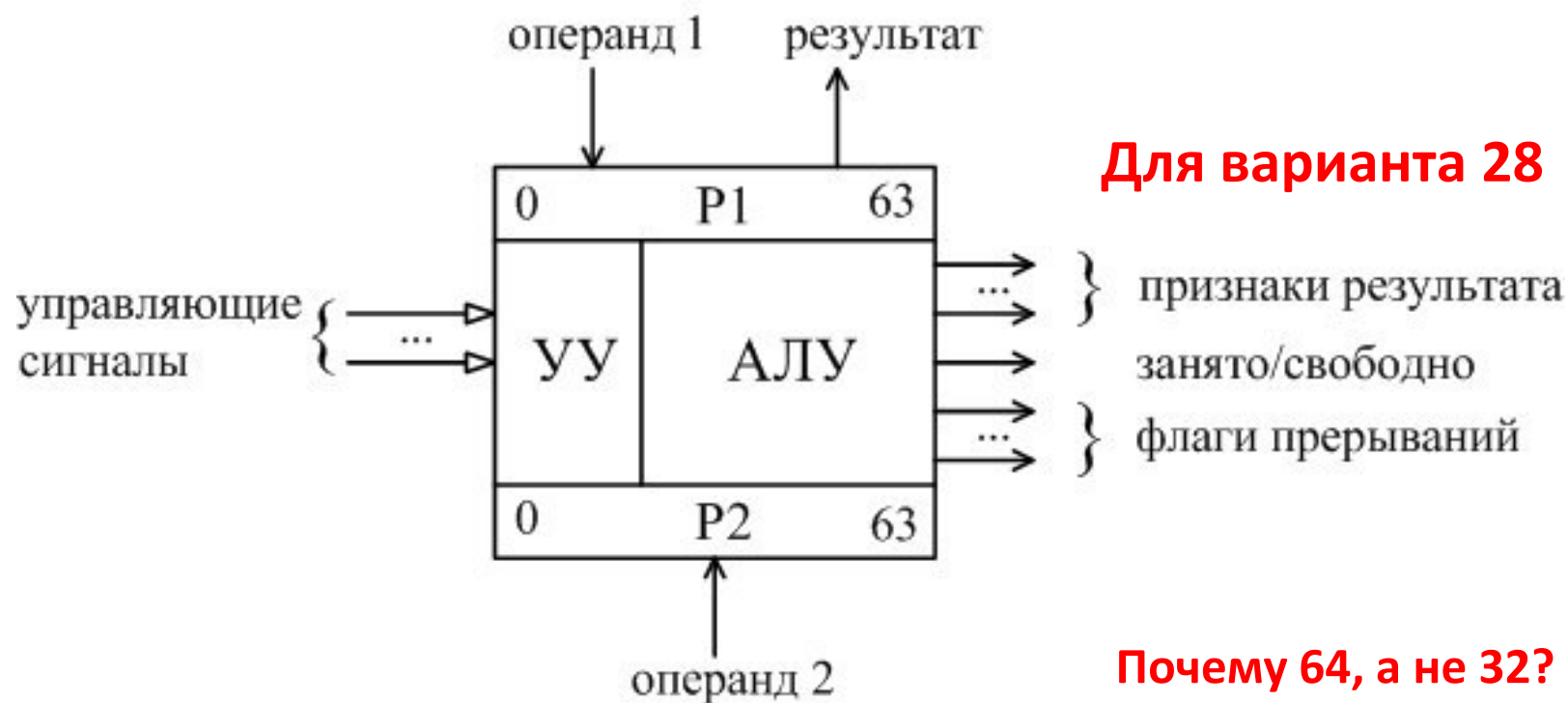


№ вар	Команды АЛУ				Дополнит. команды			ОП		РП		Способы адресации					
	КА	ДА	КЛ	ДЛ	ПУ	ОбрП	ВВ	Е _{ОП}	ШВ	Е _{РП}	Тип	Н	П	К	Р	КР	О
26	+	F8	∧	L4	ПИ<	ЗР I2	МП	512	8	8	Р	+	+	+	+		+
27	×	I2	⊕	L4	УП _М	ЗП F4	ПМ	256	8	16	Р	+	+	+		+	+
28	*	I4	√	L1	ПСч	ЗР I2	МП	128	8	32	У	+	+		+	+	+

Для варианта 27



№ вар	Команды АЛУ				Дополнит. команды			ОП		РП		Способы адресации					
	КА	ДА	КЛ	ДЛ	ПУ	ОбрП	ВВ	Е _{ОП}	ШВ	Е _{РП}	Тип	Н	П	К	Р	КР	О
26	+	F8	^	L4	ПИ<	ЗР I2	МП	512	8	8	Р	+	+	+	+		+
27	×	I2	⊕	L4	УП _М	ЗП F4	ПМ	256	8	16	Р	+	+	+		+	+
28	*	I4	√	L1	ПСч	ЗР I2	МП	128	8	32	У	+	+		+	+	+



Для варианта 28

Почему 64, а не 32?

№ вар	Команды АЛУ				Дополнит. команды			ОП		РП		Способы адресации					
	КА	ДА	КЛ	ДЛ	ПУ	ОбрП	ВВ	Е _{ОП}	ШВ	Е _{РП}	Тип	Н	П	К	Р	КР	О
26	+	F8	∧	L4	ПИ<	ЗР I2	МП	512	8	8	Р	+	+	+	+		+
27	×	I2	⊕	L4	УП _М	ЗП F4	ПМ	256	8	16	Р	+	+	+		+	+
28	*	I4	√	L1	ПСч	ЗР I2	МП	128	8	32	У	+	+		+	+	+



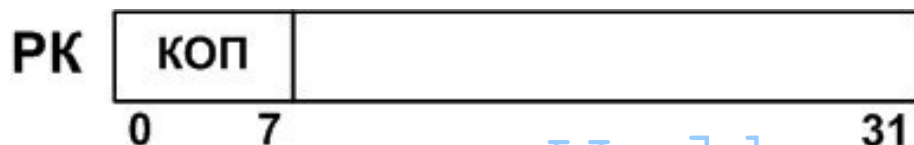
ОПРЕДЕЛЕНИЕ РАЗРЯДНОСТИ РЕГИСТРА КОМАНД

Рекомендации (обязательные для исполнения ;-)

- 1) За счет грамотного использования предусмотренных вариантом задания способов адресации операндов добиться, чтобы максимальная длина команды не превышала 4 байт.
- 2) Для обеспечения удобства выборки команды из ОП сделать длину каждой команды кратной двум (не страшно, если в команде будут лишние разряды).

Таким образом, если при разработке форматов команд, мы получим команду длиной, например, 23 бита, мы продлим ее до 32 битов. Тогда в системе будут команды только длиной 2 байта или 4 байта, что может немного увеличить длину программы, составленной из таких команд, но существенно упростит алгоритм выборки команды из ОП (будет рассмотрен позже).

Поле КОП каждой команды будет занимать 8 битов в начале команды, т.к. подразумевается, что полная система команд учебной ЭВМ должна содержать не более 256 команд. $\log_2 256 = 8$



Разрядность РК равна максимально возможной длине команды.

Всем обеспечить 32 бита!

БР будет использоваться для сокращения числа обращений к ОП в микропрограмме выборки команды. Обращение к ОП – длительная операция, занимающая несколько тактов работы процессора.

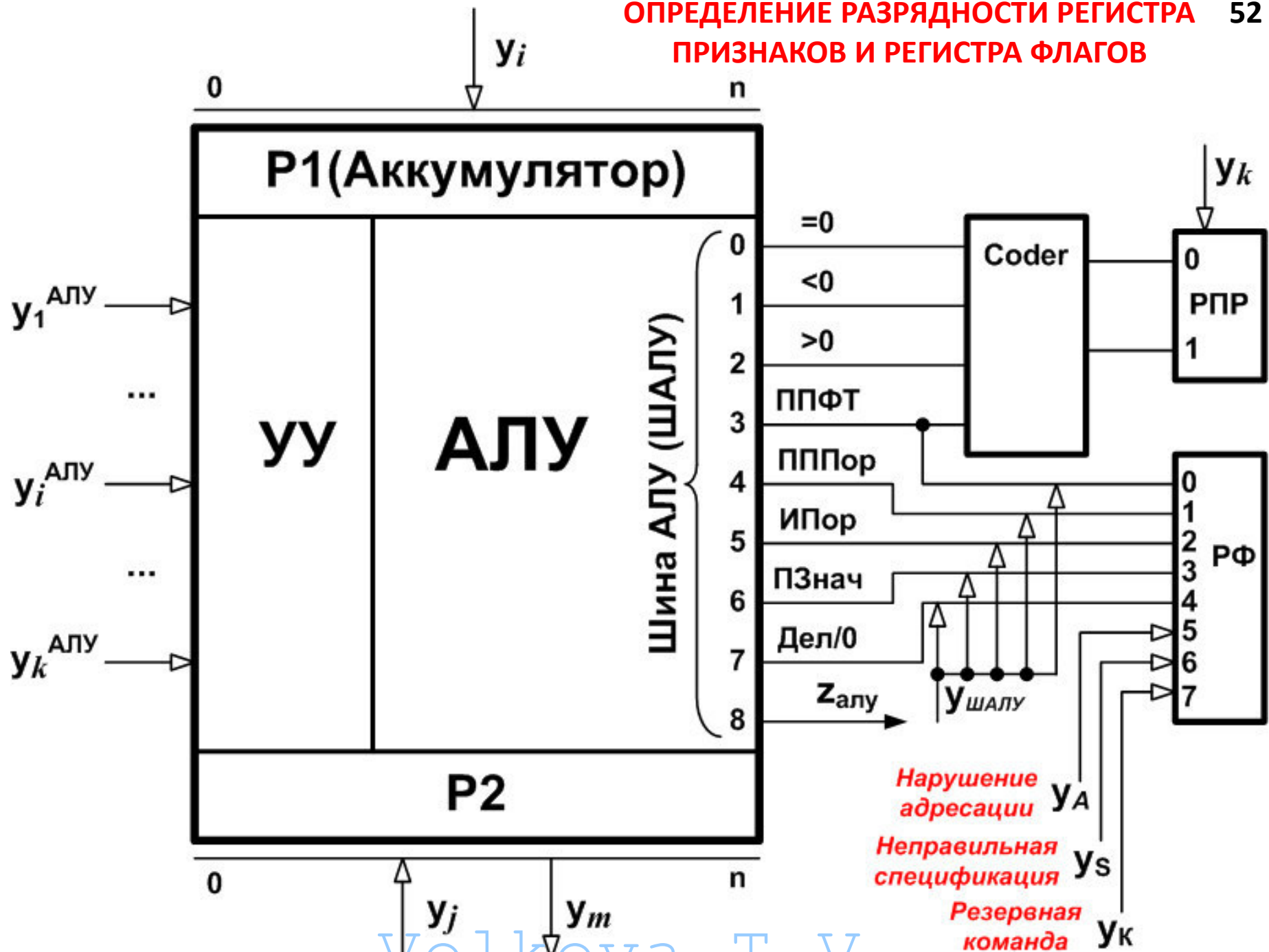
Длина БР определяется в зависимости от ширины выборки из ОП с учетом принятой (см. выше) минимально возможной длины команды (2 байта).

Если ШВ=4, то за один раз процессор выбирает из ОП слово (4 байта), в котором может находиться максимум две команды (длиной по 2 байта), последовательно расположенные в исполняемой программе. Если процессор должен выполнять первую из них, то вторую желательно сохранить внутри процессора на БР, чтобы второй раз не считывать то же самое слово из ОП.

Таким образом, при ШВ=4 байтам длина БР равна 2 байтам.

Если ШВ=8, то за один раз процессор выбирает из ОП двойное слово (8 байтов), в котором может находиться максимум четыре команды (длиной по 2 байта), последовательно расположенные в исполняемой программе. Если процессор должен выполнять первую из них, то следующие желательно сохранить внутри процессора на БР, чтобы несколько раз не считывать то же самое двойное слово из ОП. Также нет гарантии, что первая из указанных четырех команд снова не понадобится процессору: любая из последующих команд может быть командой перехода на первую команду. Поэтому желательно сохранять на БР все двойное слово, считанное из ОП на этапе выборки команды.

Таким образом, при ШВ=8 байтам длина БР равна 8 байтам.

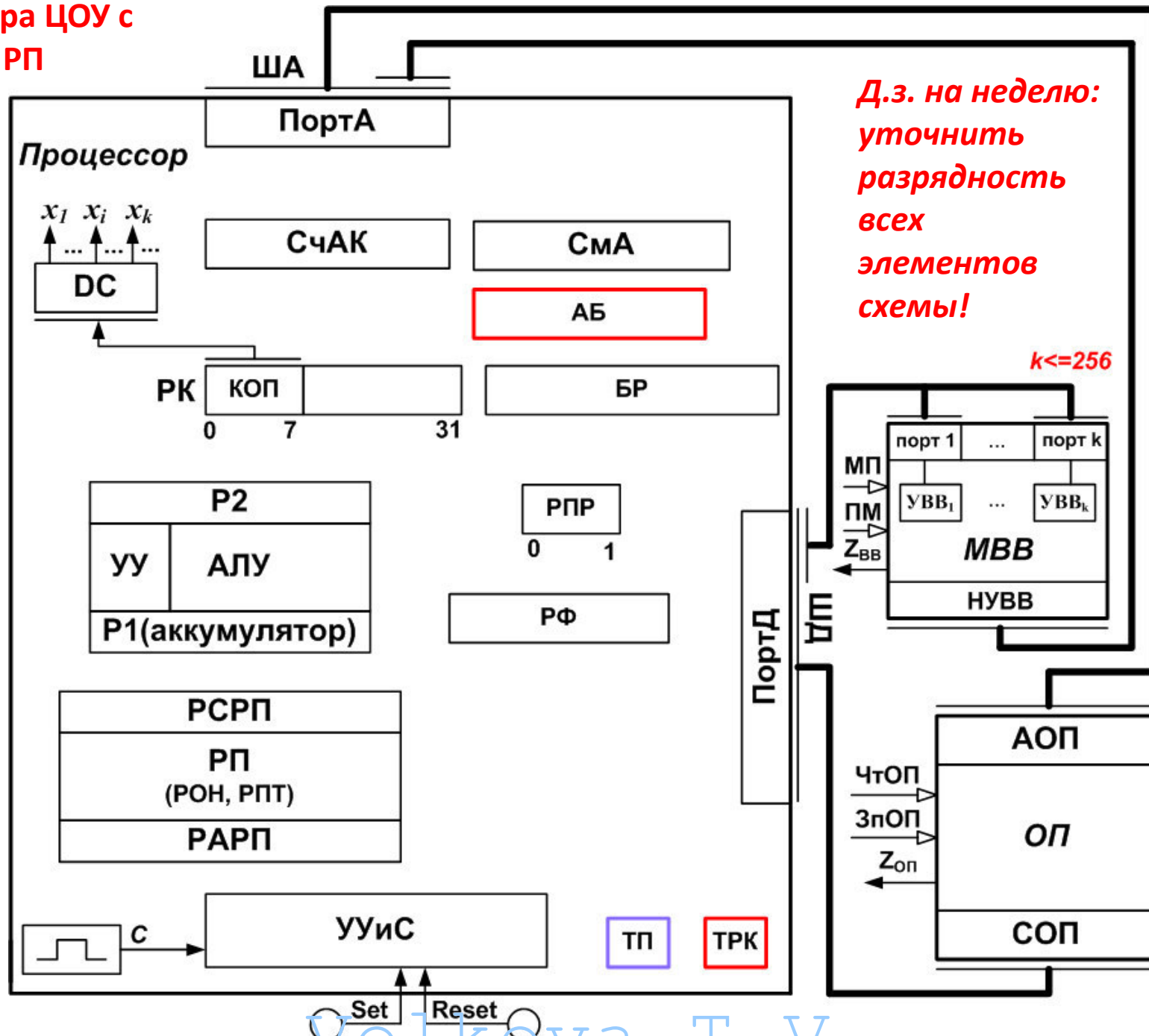


Длина регистра признака результата, как уже было сказано, равна 2 битам (унитарный код ШАЛУ(0:3) с помощью шифратора переводится в двоичный позиционный код признака результата (00 – результат равен нулю, 01 – результат меньше нуля, 10 – результат больше нуля, 11 – переполнение с фиксированной точкой).

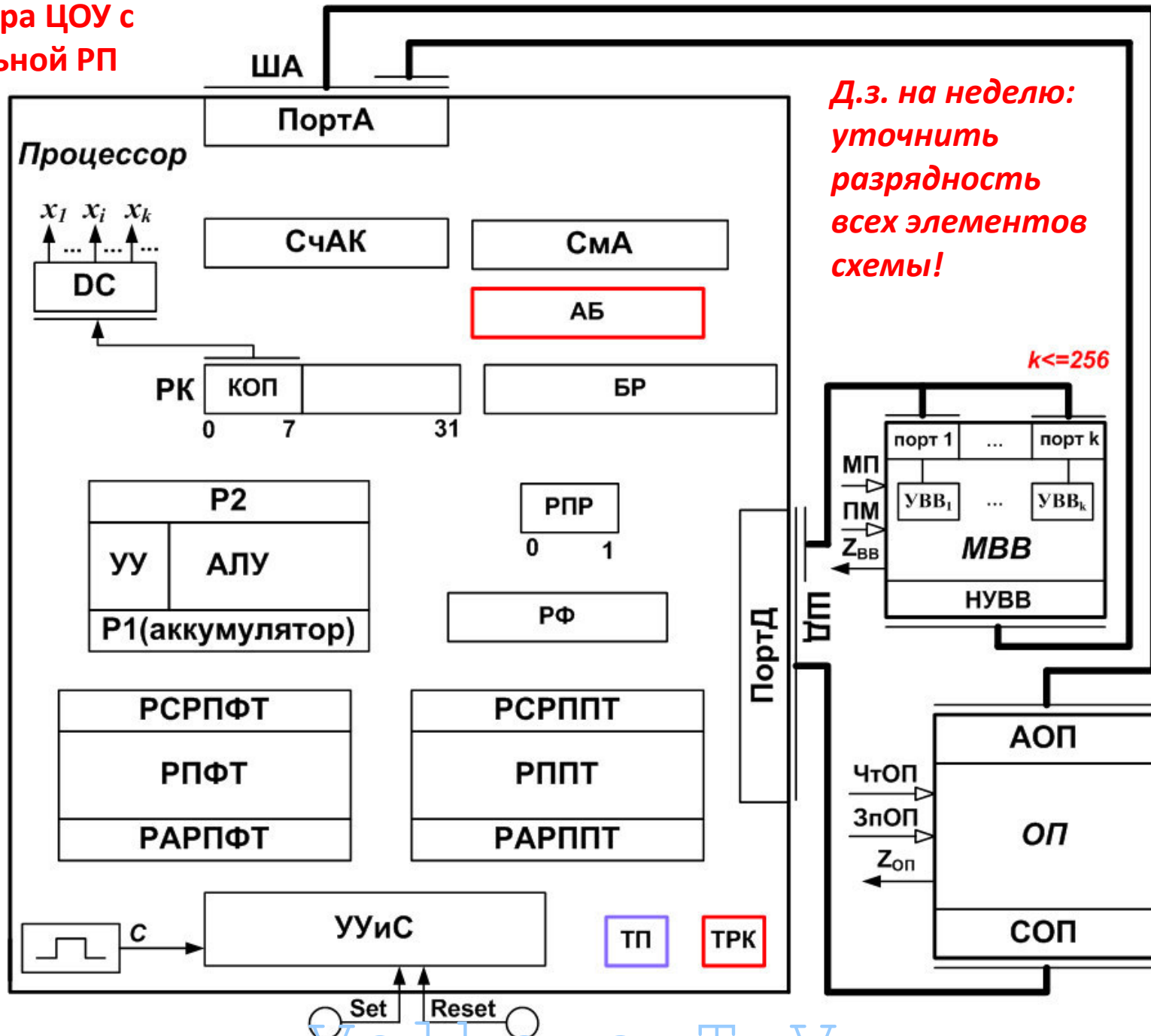
Число битов в регистре флагов РФ определяется исходя из количества прерываний от АЛУ (для конкретного варианта задания), наличия/отсутствия относительной адресации (при наличии добавляется 1 бит, который будет сигнализировать об исключительной ситуации «Нарушение адресации» – выход адреса за допустимую границу) + 1 бит для ситуации «Нарушение спецификации» (адрес операнда не заканчивается на требуемое число нулей) +1 бит для ситуации «Резервная команда» (код считанной команды не совпадает ни с одним из кодов команд, реализованных согласно варианту задания). Максимальное количество битов в РФ – 8.

Хорошая новость: систему прерываний в рамках КП проектировать не будем. При обнаружении исключительной ситуации ограничимся установкой соответствующего флага в РФ и остановом процессора. В связи с этим в структуре процессора отсутствует указатель стека (УС).

Структура ЦОУ с общей РП



Структура ЦОУ с раздельной РП



- 1. Детализация постановки задачи: уточнение структуры ЦОУ и разрядности составляющих его устройств (результаты представить на следующем практическом занятии вместе с заполненным, согласно полученному варианту, листом задания) .**
- 2. Разработка форматов команд, исполняемых процессором (согласно варианту задания на КП).**
- 3. Осмысление предложенного преподавателем алгоритма выборки команды и уточнение разрядности упоминаемых в нем устройств (см. п.1).**
- 4. Разработка отдельных ГСА выполнения команд (согласно разработанным форматам).**
- 5. Построение общей ГСА функционирования процессора.**
- 6. Разметка ГСА (нанесение управляющих и осведомительных сигналов – y_i и x_j).**
- 7. Прорисовка схем выполнения микроопераций и вычисления осведомительных сигналов на структурной схеме.**
- 8. Разработка схемы управляющего автомата (устройства управления и синхронизации).**

Хорошая новость: детализируем только схему УУиС. Все остальные устройства оставляем на уровне структурной схемы, но формулируем требования к ним в разделе «Описание структурной схемы ЦОУ».

В процессе выполнения КП эту схему нужно будет детализировать.

