

Архитектура процессора: уровни реализации алгоритмов и способы интерпретации команд

- 1 Уровни реализации алгоритмов.
 - 1.1 Микропрограммный уровень.
 - 1.2 Программный уровень.
- 2 Способы интерпретации команд.
 - 2.1 Иерархия уровней внутренних языков.
 - 2.2 Основные уровни внутренних языков и способы их интерпретации.
- 3 Особенности CISC- и RISC-архитектуры.
 - 3.1 Процессоры с CISC-архитектурой.
 - 3.2 Процессоры с RISC-архитектурой.
- 4 Виды программистских структур.

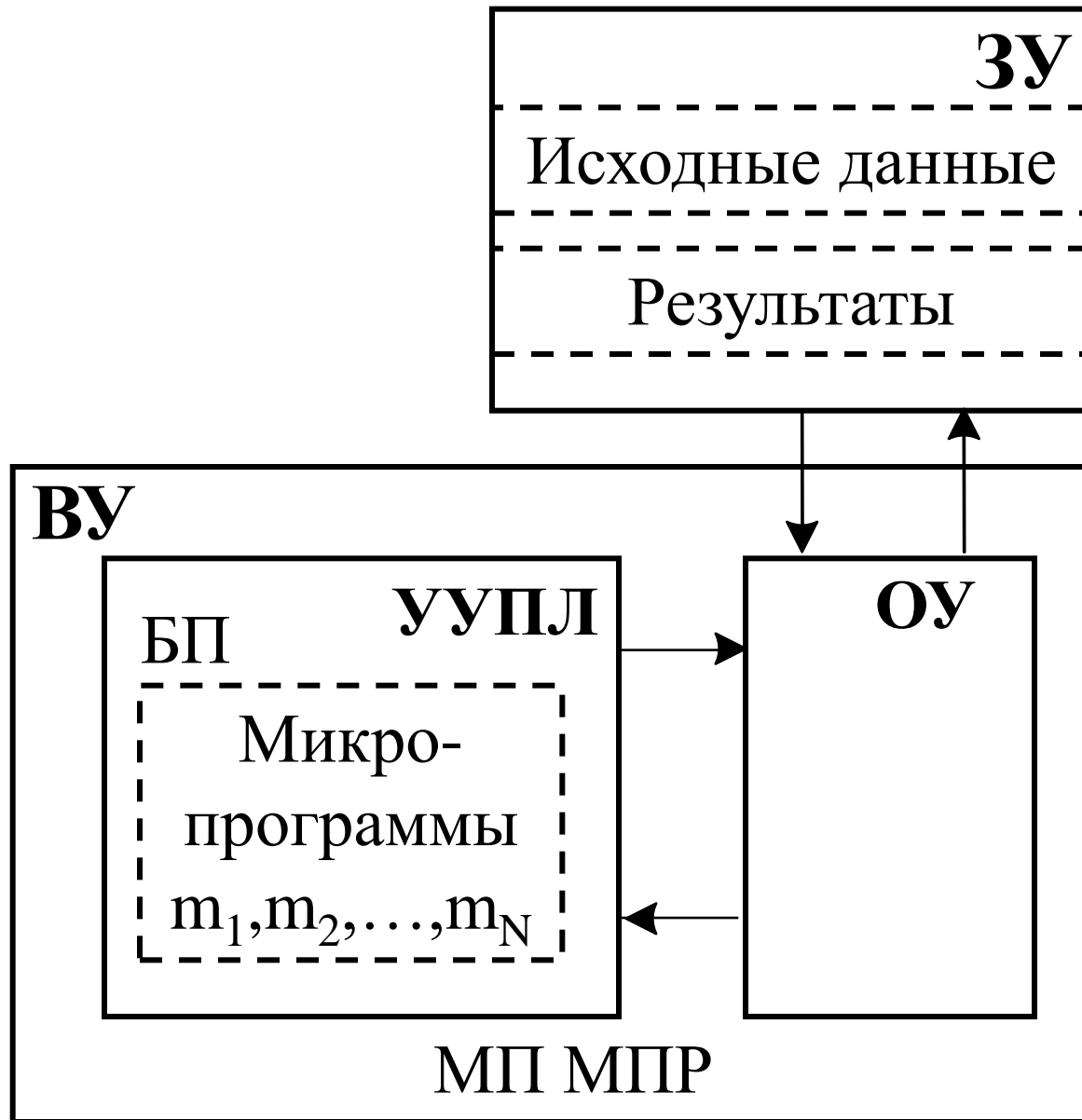
- **Знать:**
 - микропрограммную и программную реализации алгоритмов их достоинства и недостатки;
 - основные уровни внутренних языков и способы их интерпретации, понятие виртуальной машины;
 - особенности, достоинства и недостатки процессоров с CISC- и RISC-архитектурой;
 - структуру и применение регистрового файла с перекрывающимися окнами;
 - виды программистских структур.
- **Уметь:**
- **Помнить:** о влиянии архитектуры на структуру и производительность процессора.
- **Литература:** [1,14].

1 Уровни интерпретации алгоритмов

1.1 Микропрограммный уровень

- Заданное множество алгоритмов $\Lambda = \{A_1, A_2, \dots, A_N\}$ может быть реализовано микропрограммно с помощью вычислительного устройства (ВУ).
- Предположим, что в ВУ используется устройство управления с программируемой логикой (УУПЛ), тогда в памяти микропрограмм будет храниться множество микропрограмм заданных алгоритмов: $M = \{m_1, m_2, \dots, m_N\}$.

Микропрограммная реализация алгоритмов



БП – блок памяти
микропрограмм
устройства
управления с
программируе-
мой логикой;

МП МПР –
микропрогра-
мируемый
микропроцессор
.

Время выполнения алгоритма и объем памяти

- Время выполнения алгоритма при микропрограммной реализации равно времени выполнения микропрограммы: $T_{МП} = S_{МК} * t_{МК}$,
где $S_{МК}$ — число (средневзвешенное) выполняемых микрокоманд, а $t_{МК}$ — время выполнения одной микрокоманды.
- Объем памяти микропрограмм, занимаемый алгоритмом, рассчитывается по формуле
$$V_{МП} = M * n_{МК}$$

где M — число микрокоманд в микропрограмме (длина микропрограммы), $n_{МК}$ разрядность микрокоманды.

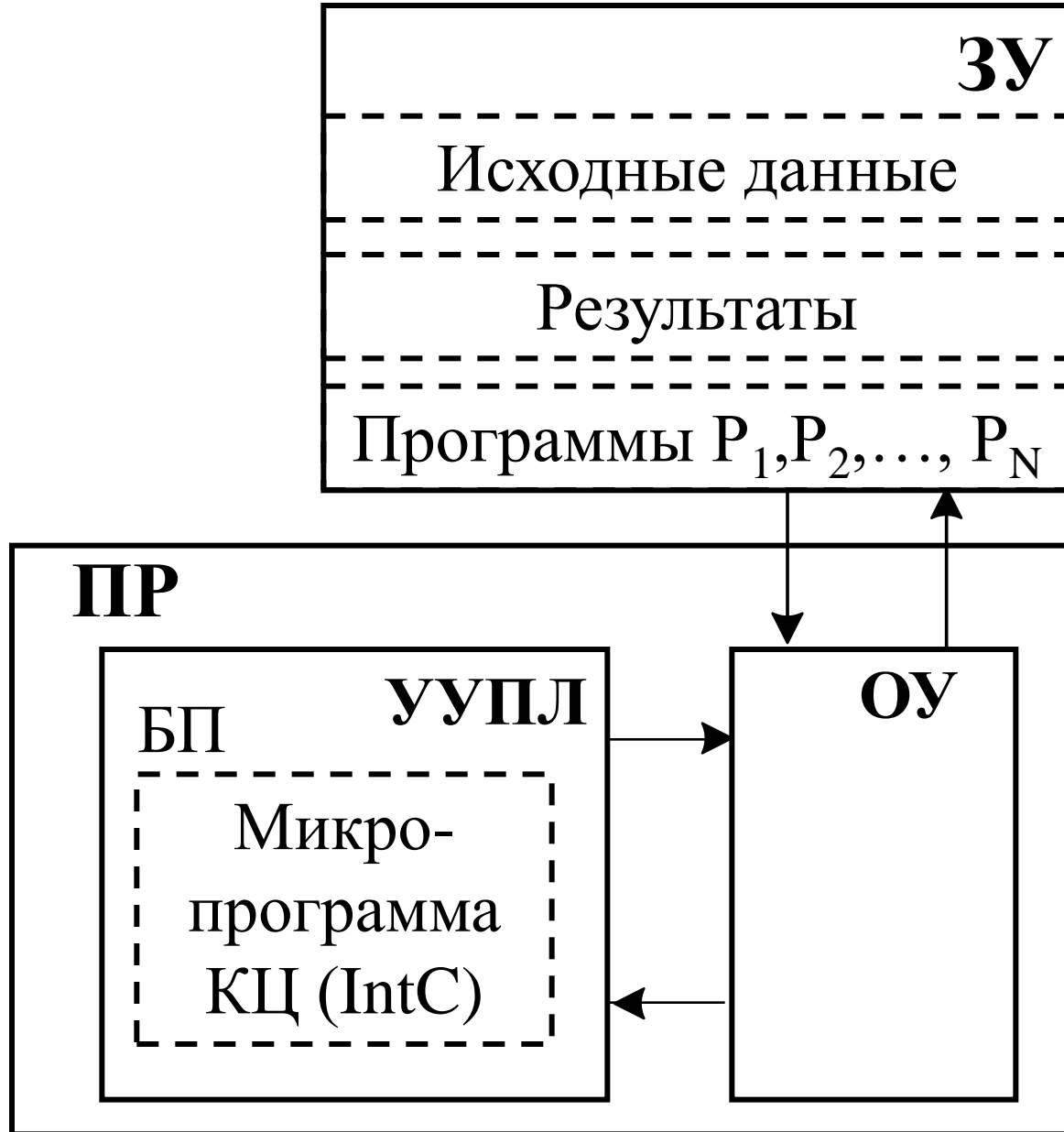
Пример расчета времени выполнения алгоритма и объема памяти

- Предположим, что алгоритм в среднем требует выполнения $S_{МК}=1000$ микрокоманд, а время выполнения одной микрокоманды составляет $t_{МК}=0,05$ мкс. Тогда время выполнения алгоритма будет равно $T_{МП}=1000*0,05=50$ мкс.
- Пусть длина микропрограммы, соответствующей алгоритму, составляет $M=100$ микрокоманд, а разрядность микрокоманды – $n_{МК}=60$ бит. Тогда объем памяти микропрограмм, занимаемый алгоритмом будет равен $V_{МП}=100*60=6000$ бит.

1.2 Программный уровень

- Заданное множество алгоритмов $\Lambda = \{A_1, A_2, \dots, A_N\}$ может быть реализовано программно с помощью процессора (ПР).
- В этом случае в запоминающем устройстве будут храниться программы алгоритмов $\Pi = \{P_1, P_2, \dots, P_N\}$.
- Предположим, что в процессоре используется устройство управления с программируемой логикой (УУПЛ), тогда в памяти микропрограмм будет храниться микропрограмма командного цикла (интерпретатор команд), обеспечивающая последовательное считывание команд из запоминающего устройства и их выполнение.

Программная реализация алгоритмов



КЦ – командный цикл;

IntC – интерпретатор команд;

ПР – процессор (с устройством управления с программируемой логикой).

Время выполнения алгоритма

- Время выполнения алгоритма при программной реализации равно времени выполнения программы:
$$T_{\Pi} = S_K * t_K$$

где S_K — число (средневзвешенное) выполняемых команд, а t_K — время выполнения одной команды.

- Время выполнения команды равно

$$t_K = S_{MKKЦ} * t_{MK}$$

где $S_{MKKЦ}$ — число (средневзвешенное) выполняемых микрокоманд в микропрограмме командного цикла, а t_{MK} — время выполнения одной микрокоманды.

Объем памяти

- Объем памяти, занимаемый программами, рассчитывается по формуле $V_{\Pi} = M_K * n_K$
где M_K – число команд в программе (длина программы),
 n_K – разрядность команды.
- Объем памяти микропрограмм, занимаемый алгоритмом, рассчитывается по формуле
 $V_{МП} = M_{МК} * n_{МК}$
где $M_{МК}$ – число микрокоманд в микропрограмме (длина микропрограммы);
 $n_{МК}$ – разрядность микрокоманды.

Пример расчета времени выполнения алгоритма и объема памяти

- Предположим, что алгоритм в среднем требует выполнения $S_K=200$ команд, а команда выполняется за время $t_K=S_{MKKЦ}*t_{MK}$. Пусть $S_{MKKЦ}=10$ (микрокоманд), а $t_{MK}=0,05$ (мкс). Тогда $t_K=0,5$ мкс, а $T_{П}=100$ мкс.
- Пусть длина программы составляет $M_K=50$ (команд), а разрядность команды — $n_K=16$ бит. Тогда $V_{П}=50*16=800$ бит.
- Предположим, что длина микропрограммы командного цикла равна $M_{МК}=40$, а разрядность микрокоманды $n_{МК}=60$ бит. Тогда $V_{МП}=40*60=2400$ бит.

Сравнение микропрограммной (МП) и программной (П) реализации алгоритма

Показатели	Реализация	
	МП	П
Время выполнения МК ($t_{МК}$), мкс	0,05	0,05
Время выполнения команды (t_K), мкс	-	0,5
Время выполнения алгоритма (T), мкс	50	100
Разрядность микрокоманды ($n_{МК}$), бит	60	60
Разрядность команды (n_K), бит	-	16
Объем памяти микропрограмм ($V_{МП}$), бит	6000	2400
Объем памяти программ (V_P), бит	-	800

2 Способы интерпретации команд

2.1 Иерархия уровней внутренних языков

Основные виды интерпретаторов

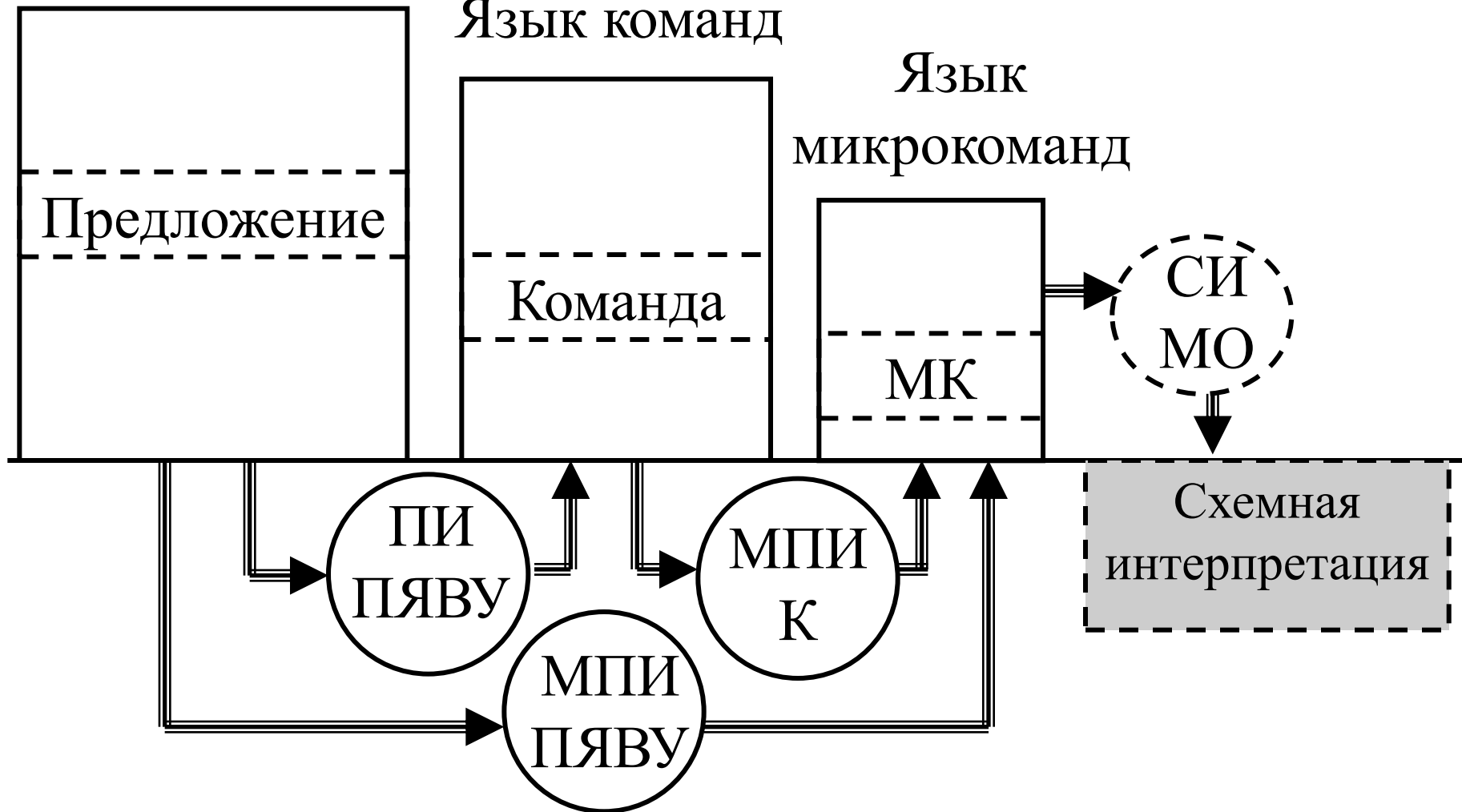
- Схемный интерпретатор микроопераций (СИ МО).
- Микропрограммный интерпретатор команд (МПИ К).
- Программный интерпретатор команд.
- Программный интерпретатор предложений языка высокого уровня (ПИ ПЯВУ).
- Микропрограммный интерпретатор предложений языка высокого уровня (МПИ ПЯВУ).

Иерархия уровней внутренних языков

Язык высокого
уровня

Язык команд

Язык
микрокоманд



2.2 Основные уровни внутренних языков и способы их интерпретации

Уровень внутреннего языка	Способ интерпретации		
	Схемный	Микро- программный	Програм- мный
Команды	RISC	CISC	Виртуальные машины
Предложения языка высокого уровня (ЯВУ)	-	С ЯВУ в качестве внутреннего	Виртуальные с внутренним ЯВУ

3 Особенности процессоров с CISC- и RISC-архитектурой

3.1 Процессоры с CISC-архитектурой

CISC-процессоры – это процессоры со сложным набором команд (Complex Instruction Set Computer).

Особенности CISC-процессоров.

Для внутреннего языка типичны следующие черты:

- большое число различных машинных команд,
- реализация некоторыми командами сложных преобразований;
- разнообразие способов адресации операндов;
- множество форматов команд различной разрядности;
- наличие команд, в которых обработка совмещается с обращением к памяти.

Программистская структура характеризуется следующей особенностью.

- В состав процессора входит сравнительно небольшое число регистров общего назначения.

Достоинства CISC-процессоров:

- возможность создания более коротких программ за счет большой «вычислительной мощности» отдельных команд;
- меньший (по сравнению с RISC-процессорами) семантический разрыв между внутренним языком машины и языком программирования высокого уровня.

Недостатки CISC-процессоров:

- усложнение структуры процессора, обусловленное достаточно сложным внутренним языком;
- необходимость использования устройства управления с программируемой логикой, характеризуемого значительными аппаратными затратами;
- сложные компиляторы, преобразующие программы с языка ассемблера в машинные коды из-за сложности внутреннего языка машины;
- значительное время машинного такта, из-за сложности выполняемых микроопераций;
- частое обращение к ЗУ, обусловленное небольшим числом регистров общего назначения;
- сложность реализации эффективного конвейера команд.

3.2 Процессоры с RISC-архитектурой

О появлении ЭВМ RISC-I

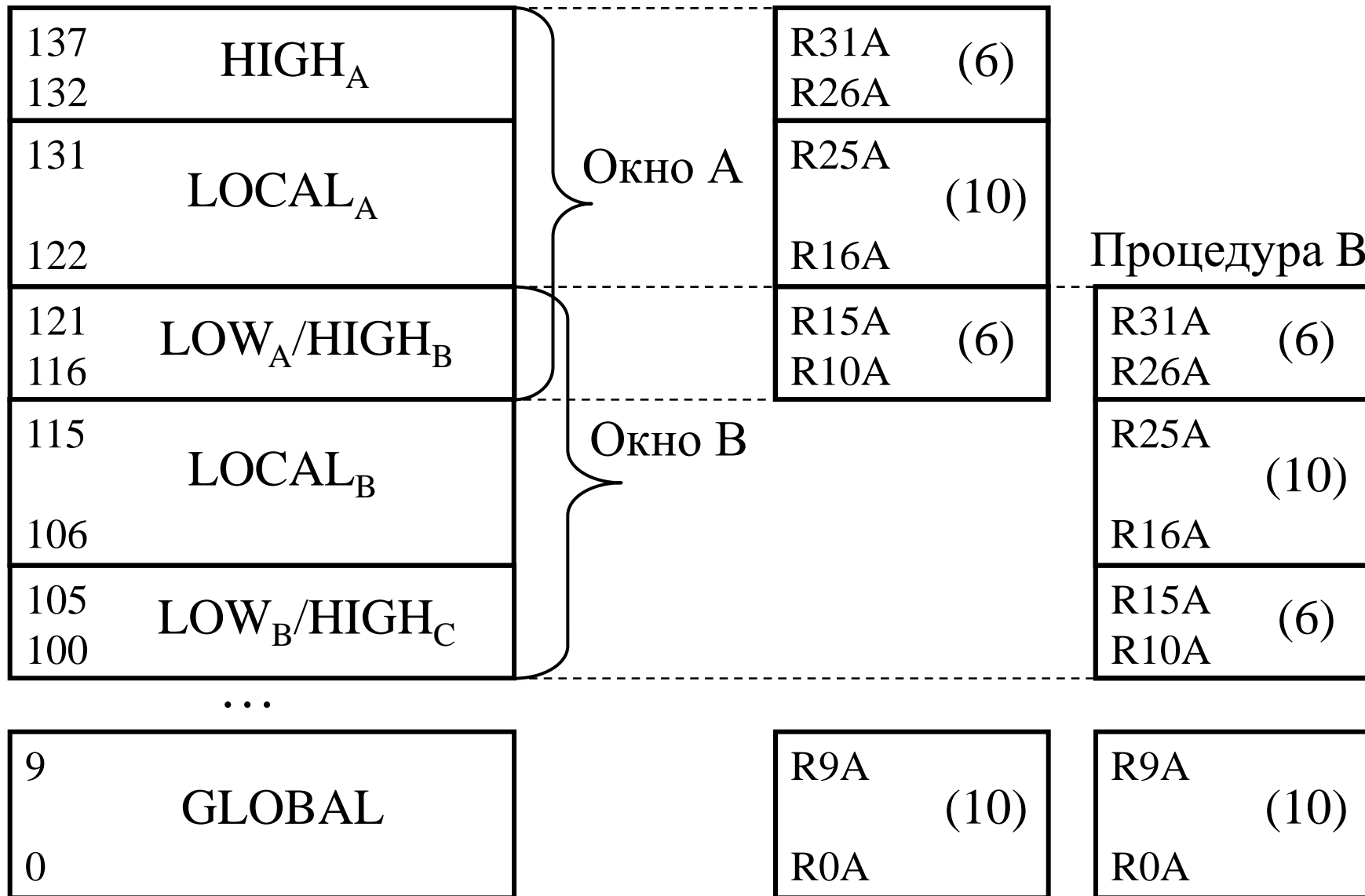
- Д. Паттерсон (D. Patterson) и его коллеги из отделения Калифорнийского университета в Беркли (1980 год) осуществили тщательный анализ программ на языках высокого уровня, в частности на СИ и Паскале и установили, что данные различных типов имеют разную частоту появления:
 - Константы – $20 \pm 7\%$;
 - Скалярные данные – $55 \pm 11\%$ (среди них 80% – локальные переменные);
 - Данные, организованные в массивы и структуры – $25 \pm 14\%$ (среди них 90% – глобальные переменные).
- *Кроме того, они обнаружили, что на вызов процедур приходится не менее 30% всего времени обработки.*
- Эти обстоятельства привели их проектированию ЭВМ RISC-I, в которой был использован регистровый файл с перекрывающимися окнами, а также набор команд, позволяющих ускорить данный вид обработки.

Система перекрывающихся окон (ЭВМ RISC-I)

Физические регистры

Логические регистры

Процедура А



RISC-процессоры – это процессоры с сокращенным набором команд (Reduced Instruction Set Computer).

Особенности RISC-процессоров.

Для внутреннего языка типичны следующие черты:

- малое число различных команд (≤ 128);
- реализация командами только простых преобразований, выполнение всех (75%) команд за один цикл (такт);
- стандартная однословная длина всех команд, равная естественной длине слова и ширине шины данных;
- малое количество форматов команд (≤ 4);
- малое число способов адресации (≤ 4);
- доступ к памяти осуществляется только командами «Чтение» и «Запись»; все команды, кроме «Чтение» и «Запись», имеют тип «регистр-регистр».

Программистская структура характеризуется следующими особенностями.

- В состав процессора входит достаточно большой регистровый файл, содержащий не менее 32 РОН.
- Регистровый файл организован в виде частично перекрывающихся друг друга «окон» регистров.

Достоинства RISC-процессоров:

- организация большого регистрового файла в виде перекрывающихся друг друга «окон» регистров позволяет при вызовах подпрограмм осуществлять быстрое переключение процессора на подпрограмму (и возвращаться к вызвавшей программе) без сохранения внутренних регистров процессора в стеке, а для передачи параметров использовать общие области смежных перекрывающихся окон.

Достоинства RISC-процессоров (продолжение):

- сокращение числа различных команд, форматов команд и данных, видов адресации позволяет разместить на кристалле процессора простое устройство управления с «жесткой» логикой, занимающее малую площадь кристалла (примерно 6%, обычно около 50%);
- Благодаря простому внутреннему языку процессора возможно создание эффективных оптимизирующих компиляторов, преобразующих программу с языка высокого уровня в машинные команды процессора, при этом часть вычислений, предусмотренных в программе на языке высокого уровня выполняется компилятором при преобразовании программы до запуска ее на выполнение.

Недостатки RISC-процессоров:

- неразвитость внутреннего языка процессора, приводящая к более длинным программам, по сравнению с программами для CISC-процессоров, прежде всего, из-за малой «вычислительной мощности» команд;
- увеличение семантического разрыва между простым внутренним языком процессора и языками высокого уровня. Это приводит к усложнению разработки трансляторов, однако простота внутреннего языка «смягчает» недостаток;
- необходимость программной реализации сложных операций (как в первых микропроцессорах).

4 Виды программистских структур

В зависимости от числа и способов использования программно доступных регистров (ПДР) процессора выделяются следующие виды программистских структур (ПС).

- *Аккумуляторные ПС* – среди небольшого числа (как правило специализированных) ПДР выделяется один (реже два) регистр – аккумулятор, который неявно адресуется командами многих операций в качестве источника операнда и/или приемника результата.
- *ПС с регистрами общего назначения* – все или большая часть ПДР равноправны и могут единообразно использоваться в командах, при большом числе ПДР они образуют регистровый файл.
- *Стековые ПС* – для хранения операндов используется специальный регистровый стек, с которым работают безадресные команды.