

ПРОЦЕССОРЫ ЭВМ

Архитектура процессора

1 Основные определения.

1.1 Архитектура и программистская структура.

1.2 Система данных и система команд.

2 Система команд: набор операций.

3 Модификации команд.

4 Число адресных полей в команде.

5 Способы адресации.

- **Знать:** понятие архитектуры процессора, основные виды и особенности операций, модификации команд арифметико-логических операций, форматы команд с различным числом адресов, основные способы адресации: вычисление исполнительного адреса, достоинства и недостатки.
- **Уметь:** для заданного числа операций, объема адресуемой памяти, числа адресов и способов формирования исполнительных адресов определить формат команды; представить схему формирования исполнительного адреса для заданного сочетания способов формирования адреса.
- **Помнить:** о сочетании различных способов адресации при формировании исполнительного адреса операнда.
- **Литература:** [1,14].

1 Основные определения

1.1 Архитектура и программистская структура



- Архитектура – описание процессора с точки зрения системного программиста.
- Программистская структура – множество программно доступных регистров иногда соединенных в соответствии направлениями пересылок данных, выполняемых командами.

1.2 Система данных и система команд

- Система данных определена, если заданы виды данных (числовые, графические, символьные) и для каждого вида данных определены необходимые характеристики. Например: для числовых данных – система счисления; формы представления чисел (ФЗ, ПЗ); используемые коды (ПК, ДК,...).
- Система команд определена, если заданы следующие составляющие: набор операций, способы адресации, форматы и модификации команд, а также описаны функции, выполняемые каждой командой.

2 Система команд: набор операций

Арифметические операции

- *Арифметические операции* над данными в форме с фиксированной запятой: сложение, сложение модулей, вычитание, вычитание модулей, инкремент, декремент, умножение, деление и другие.
- *Арифметические операции* над данными в форме с плавающей запятой: сложение, вычитание, умножение, деление или вычисление обратной величины, суммирование произведений, вычисление элементарных функций и другие.

Сдвиги и вспомогательные операции

- *Сдвиги логические* в сторону младших (старших) разрядов на один разряд или на заданное число разрядов.
- *Сдвиги арифметические* в сторону младших (старших) разрядов на один разряд или на заданное число разрядов.
- *Сдвиги циклические* в сторону младших (старших) разрядов на один разряд или на заданное число разрядов.
- *Подсчет числа единиц* (нулей) в двоичном коде.
- *Определение номера первого по порядку разряда*, начиная с младших (старших) разрядов двоичного кода, содержащего цифру «0» («1»).

Логические операции и операции установки значений

- Инверсия.
- Конъюнкция.
- Сумма по модулю два и поразрядные операции из других логических функций.
- Установка регистра или отдельных его разрядов в единичное (нулевое) состояние.

Операции пересылки

- Пересылки данных из регистра в регистр ($R - R$).
- Пересылки данных из регистра в память ($R - M$), в том числе и запись в стек.
- Пересылки данных из памяти в регистр ($M - R$), в том числе и чтение из стека.
- Пересылки данных из памяти в память ($M - M$).

Операции управления

- Безусловного перехода.
- Условного перехода по прямому (инверсному) значению признака, при этом признак может выбираться с помощью маски.
- Условного (безусловного) обращения к подпрограмме.
- Условного (безусловного) возврата из подпрограммы.
- Инициализации программных прерываний.
- Возврата из прерывающих программ.
- Остановка.

Операции ввода и вывода

- *Ввод данных* из регистров внешнего устройства (ВУ) осуществляется с помощью специальных команд путем пересылки данных в регистры процессора (ПР) или память.
- *Ввод данных* из регистров ВУ производится с помощью команд чтения данных из памяти в регистр ПР (или память). В этом случае регистры ВУ отображаются в адресном пространстве памяти, с которой работает процессор, и рассматриваются как ячейки памяти.
- *Вывод данных* из регистров ПР в регистры ВУ осуществляется с помощью специальных команд путем пересылки данных из регистра ПР в регистр ВУ (из памяти в регистр ВУ).
- *Вывод данных* из регистров ПР в регистры ВУ с помощью команд записи данных в память. В этом случае регистры ВУ отображаются в адресном пространстве памяти.

Специальные операции

- Операции диагностики.
- Операции чтения и загрузки системных регистров процессора.
- Операции установки значений определенным полям системных регистров процессора.
- Другие операции привилегированных команд.

В простейшем случае в команде выделяются две части.



- КОП – код операции. Разрядность поля КОП определяется следующим образом: $n_{\text{КОП}} = E(\log_2 K)$, K – число различных операций.

3 Модификации команд

Одноместные и двухместные арифметико-логические операции

- Арифметико-логические операции можно поделить на двухместные $c := a * b$ и одноместные $d := \#a$,
где a — источник первого; b — второго операнда; c , d — приемники результатов; $*$, $\#$ — символы операций.
- Двухместные операции требуют введения в команду трех указателей: a , b , c .
- Число указателей может быть сокращено до двух, если результат операции помещается на место одного из операндов: $a := a * b$ или $b := a * b$.

Модификации команд двухместных арифметико-логических операций

- Например, если используются одноадресные команды и $a, b \in \{AC, A\}$ (AC – идентификатор аккумулятора, A – адрес ячейки памяти), то возможны четыре модификации команд двухместных арифметико-логических операций:
 $(AC) := (AC) \% M[A],$
 $M[A] := (AC) \% M[A],$
 $(AC) := M[A] \% (AC),$
 $M[A] := M[A] \% (AC);$
где (AC) – содержимое AC , а $M[A]$ – содержимое ячейки памяти с адресом A , $\%$ – операция.

Модификации команд одноместных арифметико-логических операций

- Одноместные операции ($d := \#a$) предполагают использование двух указателей: a и d .
- Число указателей можно сократить до одного, если результат операции помещать на место операнда:
 $a := \#a$.
- Например, если $a, d \in \{AC, M[A]\}$, то будет четыре модификации команд одноместных арифметико-логических операций:
 $(AC) := \#M[A],$
 $M[A] := \#(AC),$
 $(AC) := \#(AC),$
 $M[A] := \#M[A].$

Число адресных полей

- *Безадресные команды*

КОП

Пример. $(AC):=(AC)+1$.

- *Одноадресные команды*

КОП	a
-----	---

Примеры.

1) $a=A$ – адрес ячейки памяти. $(AC):=(AC)+ M[A]$.

2) $a=i$ – номер регистра. $R[i]:=R[i]+ (AC)$,
где $R[i]$ – содержимое регистра с номером i .

- *Двухадресные команды*

КОП	a	b
-----	---	---

Примеры. 1) $a=A1, b=A2$. $M[A1]:=M[A1]+ M[A2]$.

2) $a=i1, b=i2$. $R[i1]:=R[i2]+ (AC)$.

Число адресных полей

(трех- и четырехадресные команды)

- Трехадресные команды*

КОП	a	b	c
-----	---	---	---

Примеры.

1) $a=i1, b=i2, c=A1. \quad R[i1]:=R[i2]+ M[A1].$

2) $a=A1, b=A2, c=A3. \quad M[A1]:=M[A2]+M[A3].$

- Четырехадресные команды*

КОП	a	b	c	K
-----	---	---	---	---

КОП	a	b	c	АСК
-----	---	---	---	-----

K — константа,

АСК — адрес
следующей
команды

Пример.

$a=i1, b=i2, c=i3, K=1. \quad R[i1]:=R[i2]+ R[i3]+1.$

4 Способы адресации

- **Неявная адресация.** Источники операндов и приемник результата указываются неявно, обычно с помощью кода операции команды.

КОП

Пример. $(AC) := (AC) + 1$.

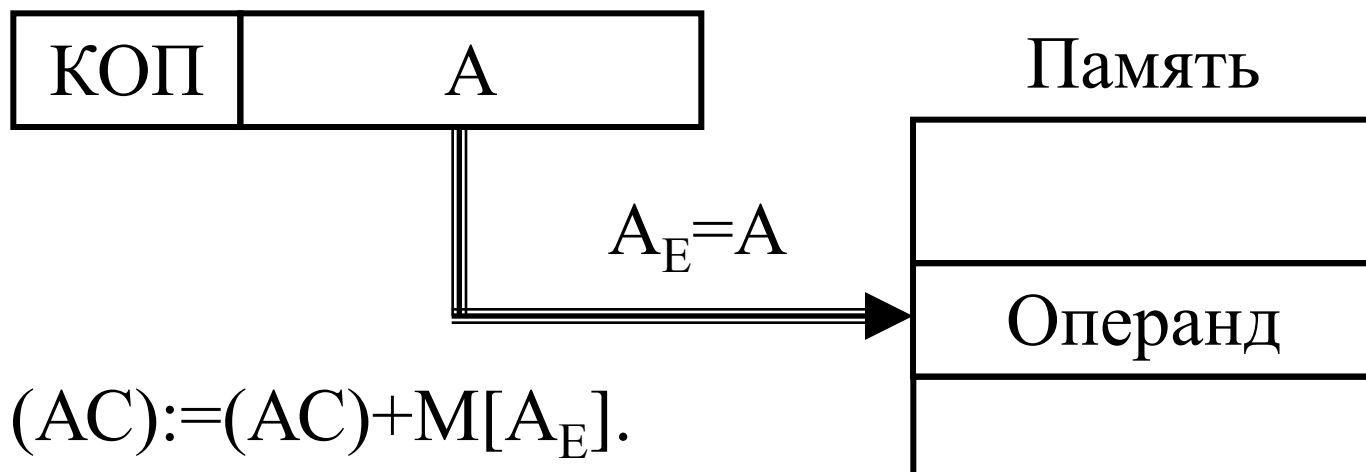
- **Непосредственная адресация.** В коде команды задается сам операнд (операнды).

КОП

Операнд (O)

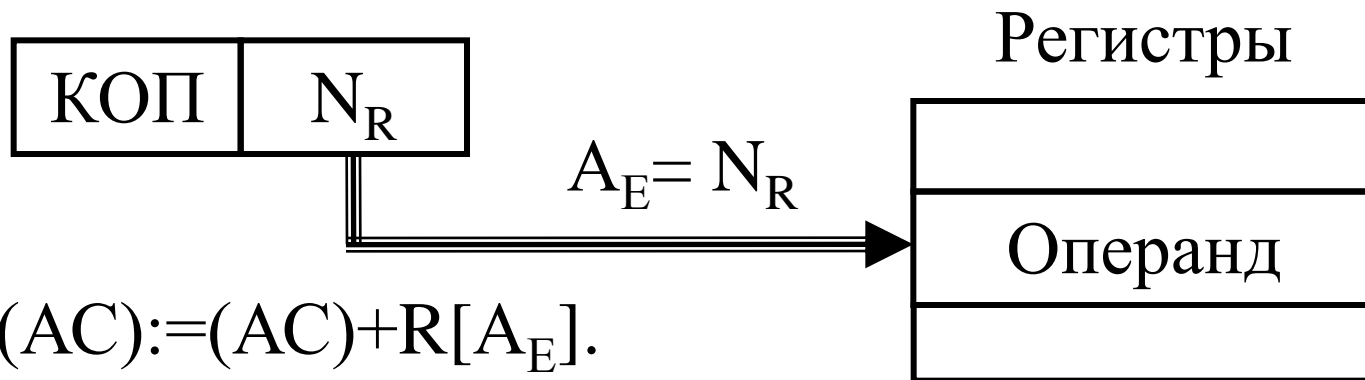
Пример. $(AC) := (AC) + O$.

- **Прямая (абсолютная) адресация.** В коде команды задается адрес (А) операнда.



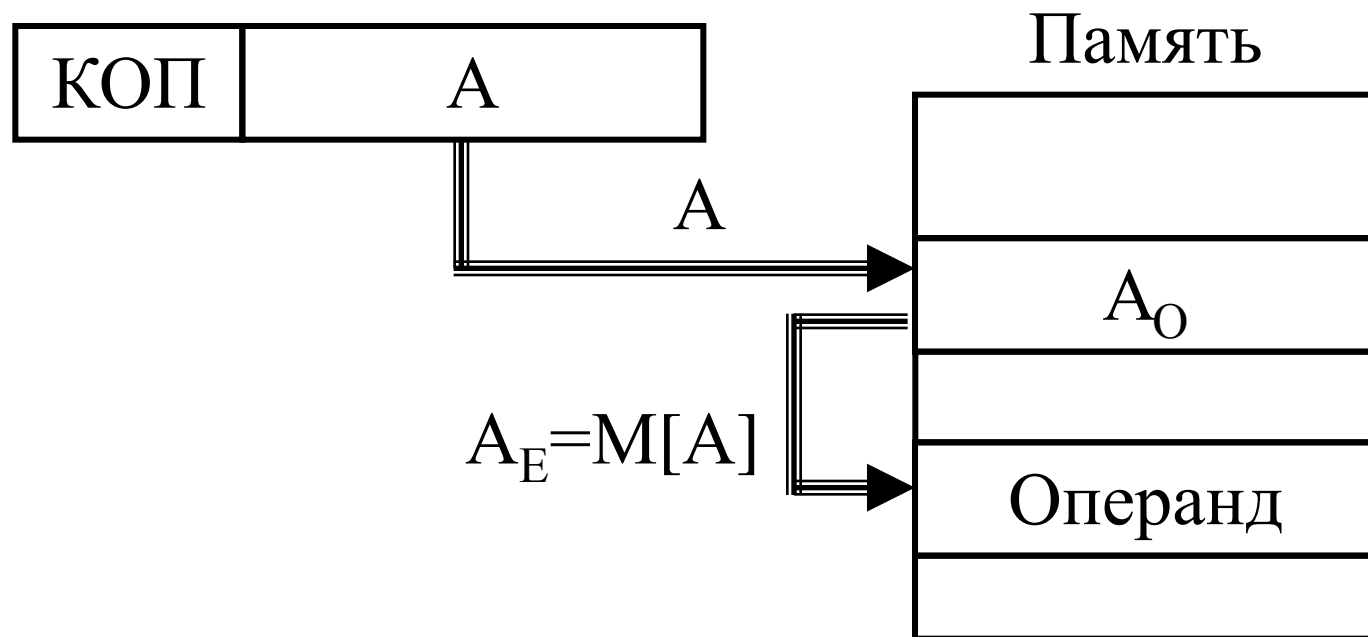
$$(AC) := (AC) + M[A_E].$$

- **Регистровая (прямая регистровая) адресация.** В коде команды задается номер N_R регистра, содержащего операнд.



$$(AC) := (AC) + R[A_E].$$

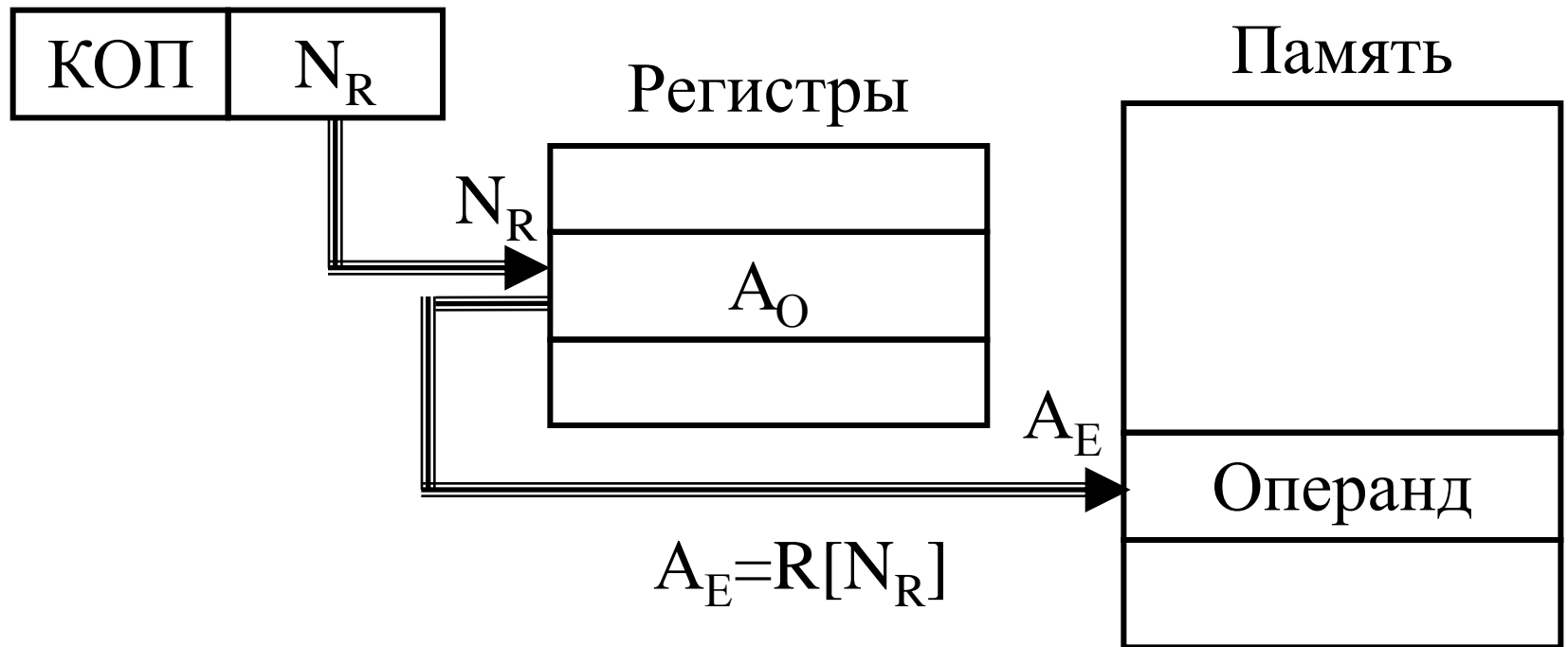
- **Косвенная адресация.** В коде команды при одноступенчатой косвенной адресации задается адрес (A) ячейки памяти, в которой хранится адрес операнда (A_O). Косвенная адресация может быть многоступенчатой, в том числе и с произвольным числом ступеней.



$(AC) := (AC) + M[M[A]].$

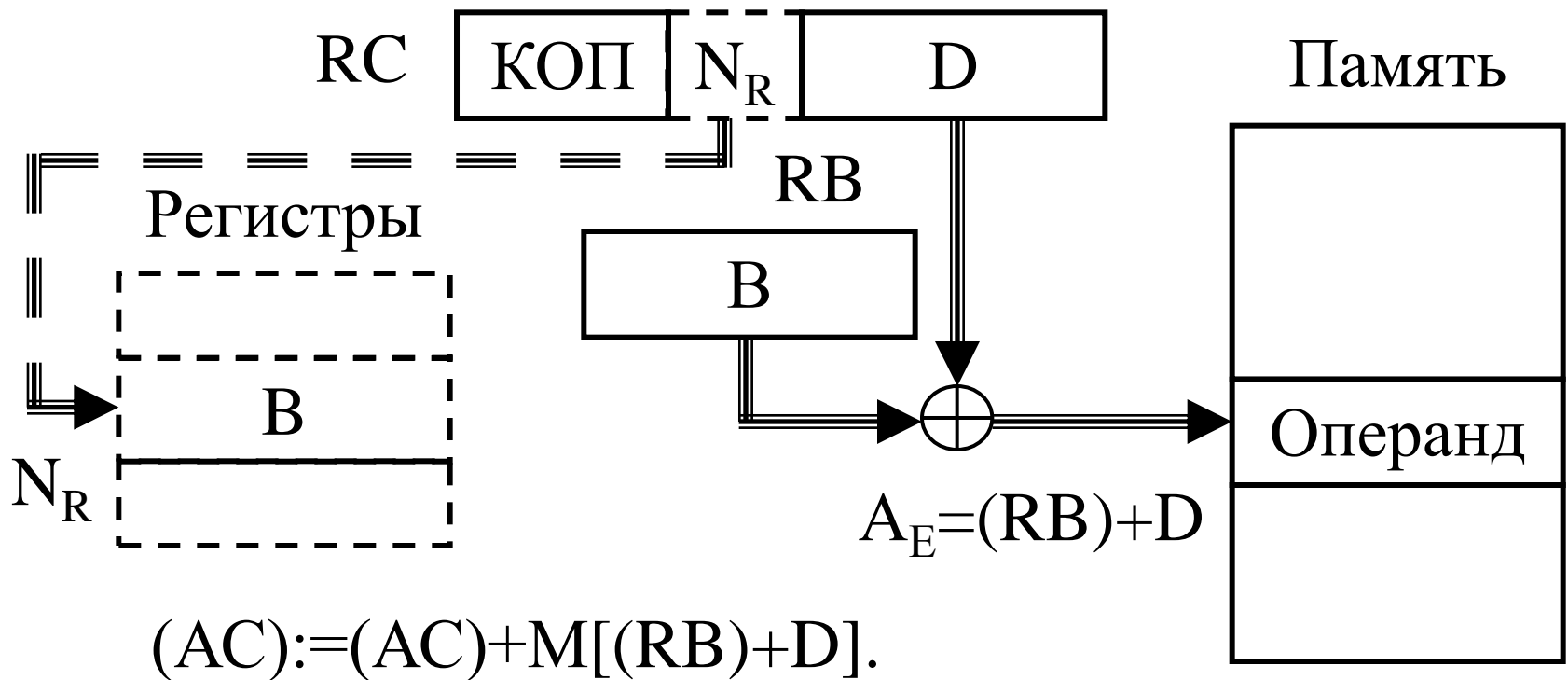
$(AC) := (AC) + M[\dots M[A]].$

- *Косвенная регистровая адресация.* В коде команды задается номер N_R регистра, содержащего адрес (A_O) ячейки памяти, в которой хранится операнд.

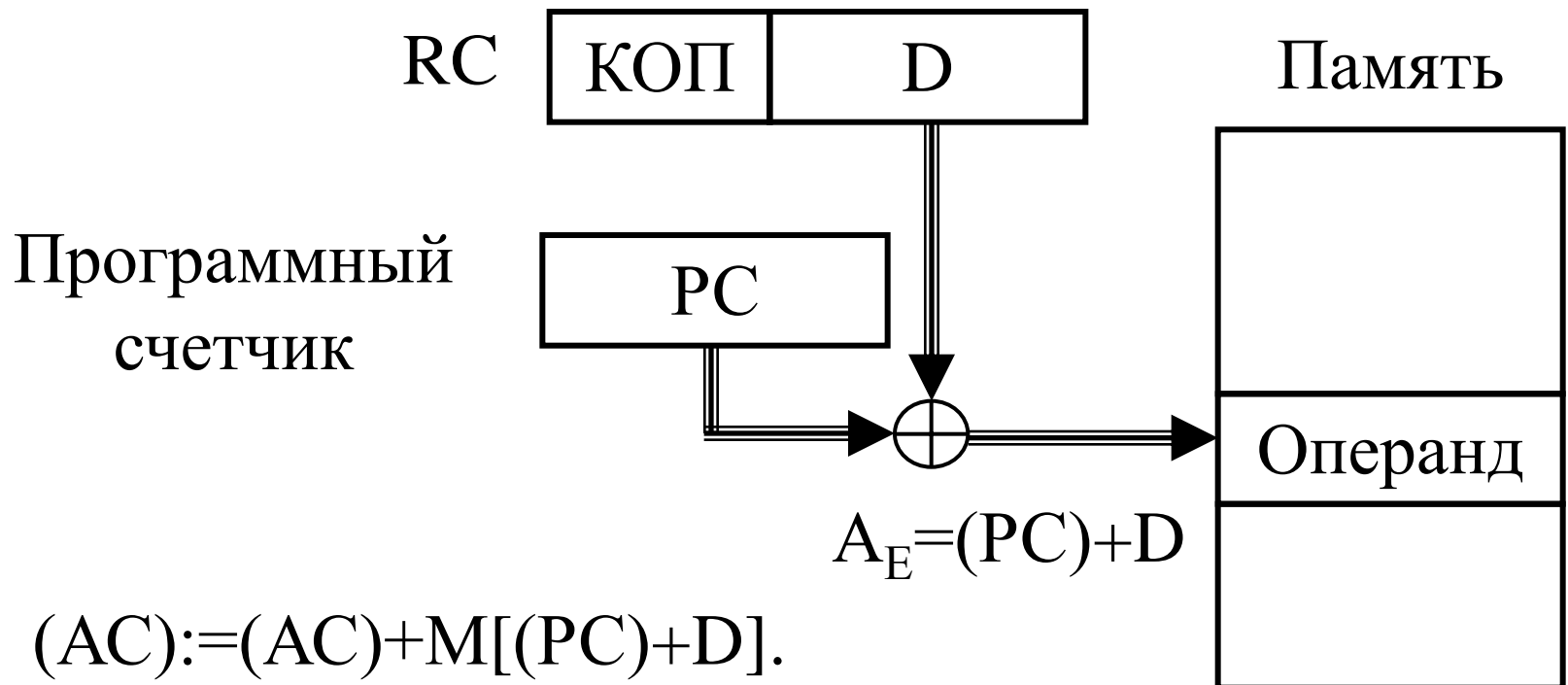


$$(AC) := (AC) + M[R[N_R]].$$

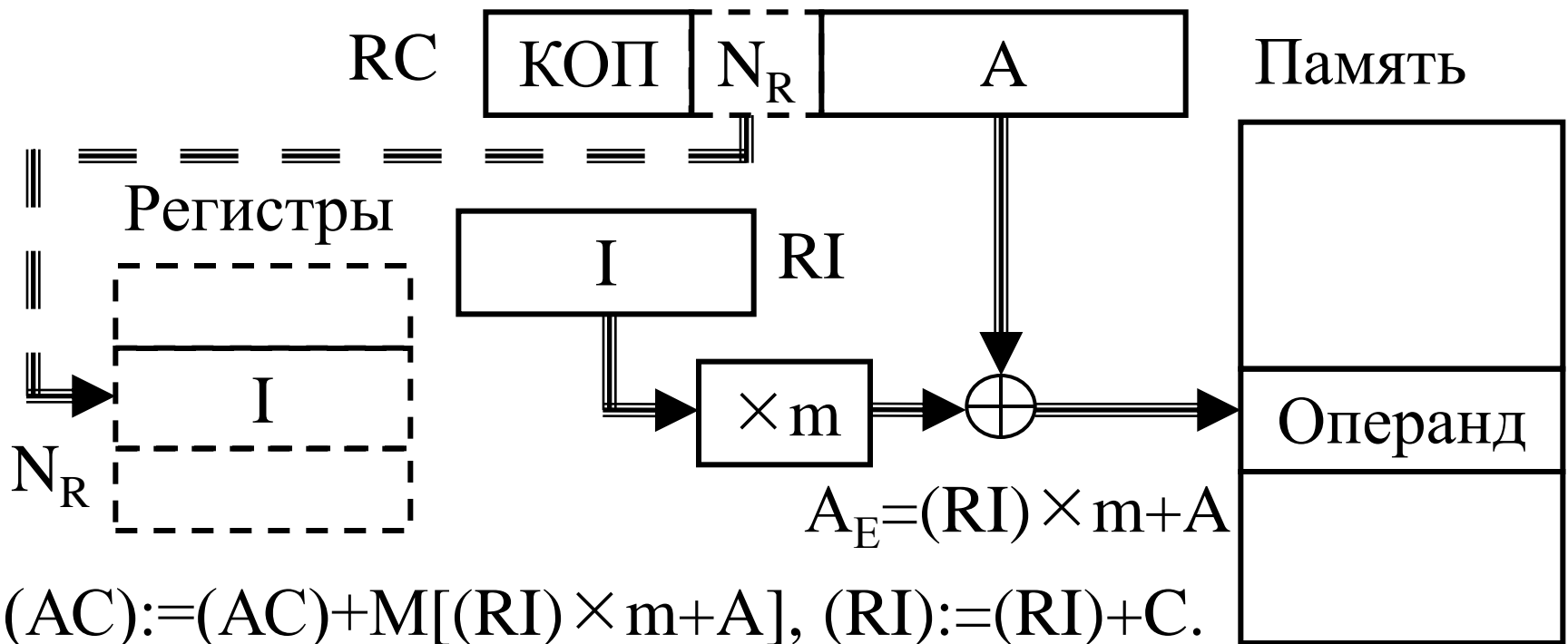
- Базовая адресация.** Исполнительный адрес вычисляется путем суммирования адреса (D) из команды (смещения) и базы (B), которая берется из специального регистра базы (RB) или регистра общего назначения (РОН), выбираемого по номеру, указанному в команде. $A_E = (RB) + D$ или $A_E = R[N_R] + D$.



- **Относительная адресация.** Исполнительный адрес вычисляется путем суммирования адреса (D) из команды (смещения) и содержимого программного счетчика (PC). Обычно поле D интерпретируется как двоичное число в дополнительном коде, что позволяет задавать как положительное, так и отрицательное смещение.



- Индексная адресация.** Исполнительный адрес вычисляется путем суммирования адреса (A) из команды и индекса (I), который берется из специального регистра индекса (RI) или РОН. Индекс может быть умножен на масштабный множитель $m=1,2,4,8$. $A_E = (RI) \times m + A$ или $A_E = R[N_R] \times m + A$.



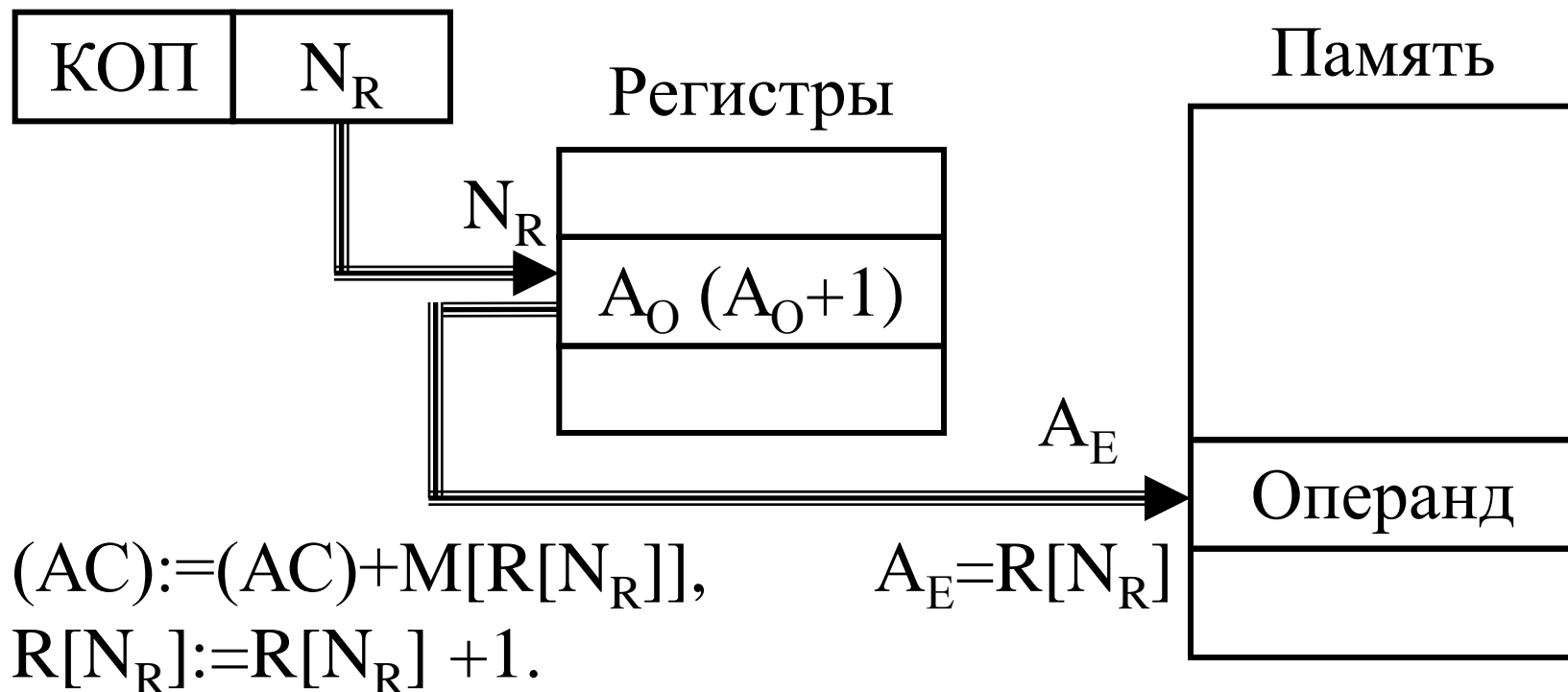
- *Автоиндексация.* При индексной адресации осуществляется уменьшение или увеличение содержимого индексного регистра до или после обращения к нему с помощью дополнительной команды. Обычно изменение содержимого регистра происходит на единицу, поэтому целесообразно сделать это «автоматически» микропрограммно в команде, использующей индексную адресацию.

Автоинкрементная		Автодекрементная	
Постинкрементная	Преинкрементная	Постдекрементная	Предекрементная
$A_E = (RI) + A,$ $(RI) := (RI) + 1$	$(RI) := (RI) + 1,$ $A_E = (RI) + A$	$A_E = (RI) + A,$ $(RI) := (RI) - 1$	$(RI) := (RI) - 1,$ $A_E = (RI) + A$

Способы адресации

1. Неявная
2. Непосредственная
3. Прямая
 1. К памяти
 2. К регистру
4. Косвенная
 1. Регистровая
 2. К памяти
 1. Одноступенчатая
 2. Многоступенчатая
5. Базовая (с регистром базы, с общими регистрами)
6. Относительная
7. Индексная (с индексным регистром, с общими регистрами)
 1. Общая
 2. Автоинкрементная
 1. Постинкрементная
 2. Преинкрементная
 3. Автодекрементная
 1. Постдекрементная
 2. Предекрементная

- **Косвенная регистровая автоинкрементная адресация.** В коде команды задается номер N_R регистра, содержащего адрес (A_O) ячейки памяти, в которой хранится операнд. После формирования исполнительного адреса содержимое регистра с номером N_R увеличивается на единицу.



- **Базово-индексная адресация.** Исполнительный адрес формируется следующим образом:

$$A_E = (RB) + D + (RI) \times m$$

или

$$A_E = R[N_{RB}] + R[N_{RI}] \times m + D,$$

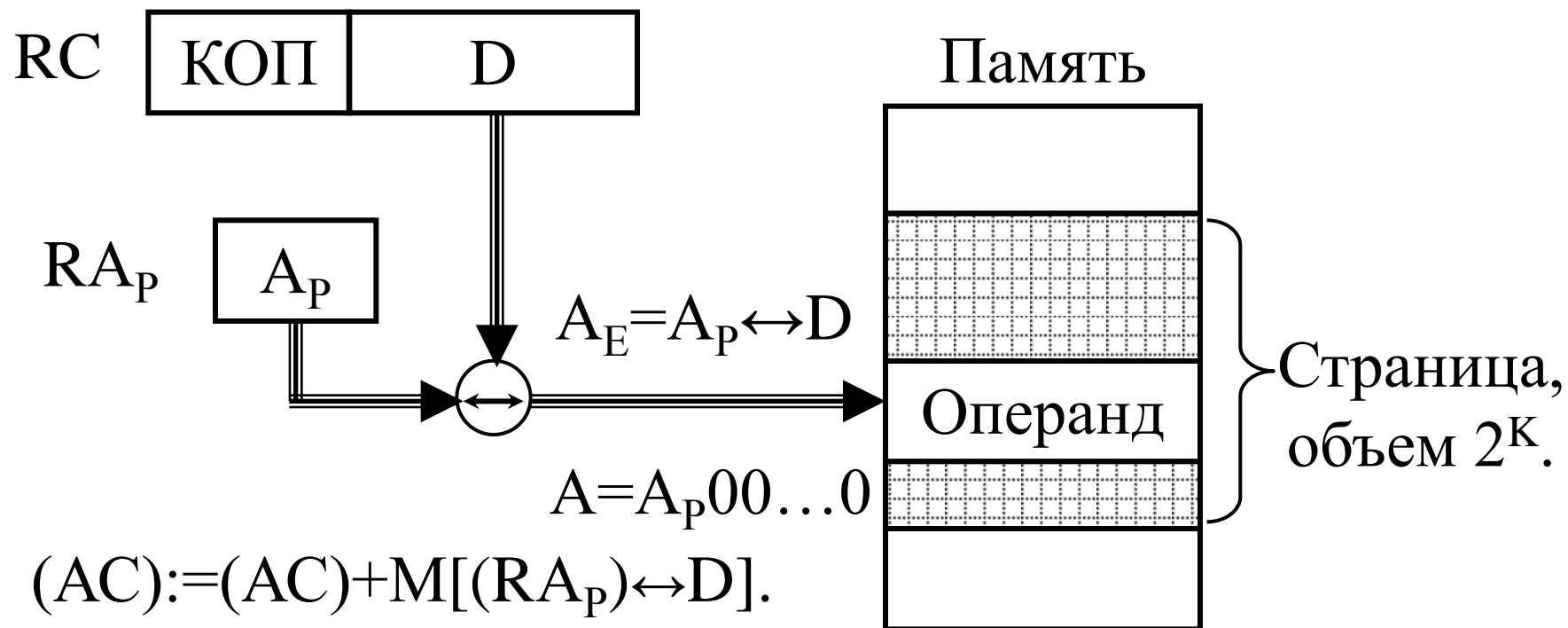
где RB – регистр, а N_{RB} – номер регистра базы; RI – регистр, а N_{RI} – номер регистра индекса; D – смещение адреса, указанное в команде; m – масштабный множитель для умножения индекса.

- **Косвенная автодекрементная адресация.** Исполнительный адрес формируется следующим образом:

$$M[A] := M[A] - 1,$$

$$A_E = M[A].$$

- Страничная адресация.** Исполнительный адрес вычисляется путем конкатенации (присоединения) K -разрядного адреса (D) операнда на странице, указанного в команде к старшим разрядам адреса (A_p), которые рассматриваются как адрес страницы и берутся из специального регистра адреса страницы (RA_p). $A_E = (RA_p) \leftrightarrow D$.



- ***Особенности задания способов адресации***
 - Для указания способа адресации в формате команды может быть специальное поле для каждого адреса. Разрядность этого поля определяется числом различных способов формирования адреса, применяемых в команде.
 - Исполнительные адреса первого и второго операндов команды могут формироваться различными способами. Например, для первого операнда применяется непосредственная, а для второго – косвенная регистровая адресация.
 - При формировании исполнительного адреса операнда может использоваться сочетание различных способов формирования адреса.

1. Неявная
2. Непосредственная
3. Прямая
 1. К памяти
 2. К регистру
4. Косвенная
 1. Регистровая
 2. К памяти
 1. Одноступенчатая
 2. Многоступенчатая

5. Базовая
 5. С регистром базы
 6. С общими регистрами
6. Относительная
7. Индексная
 1. Общая
 2. Автоинкрементная
 1. Постинкрементная
 2. Преинкрементная
 3. Автодекрементная
 1. Постдекрементная
 2. Предекрементная