Примеры архитектур процессоров

1 Микропроцессор КР580ИК80 (Intel 8080).

- 1.1 Программистская структура.
- 1.2 Особенности состава операций и представления данных
- 1.3 Форматы команд и способы адресации.

2 Микропроцессор К1810ВМ86 (Intel 8086).

- 2.1 Программистская структура.
- 2.2 Представление данных и состав операций.
- 2.3 Выбор сегмента и формирование физического адреса.
- 2.4 Способы адресации и форматы команд.

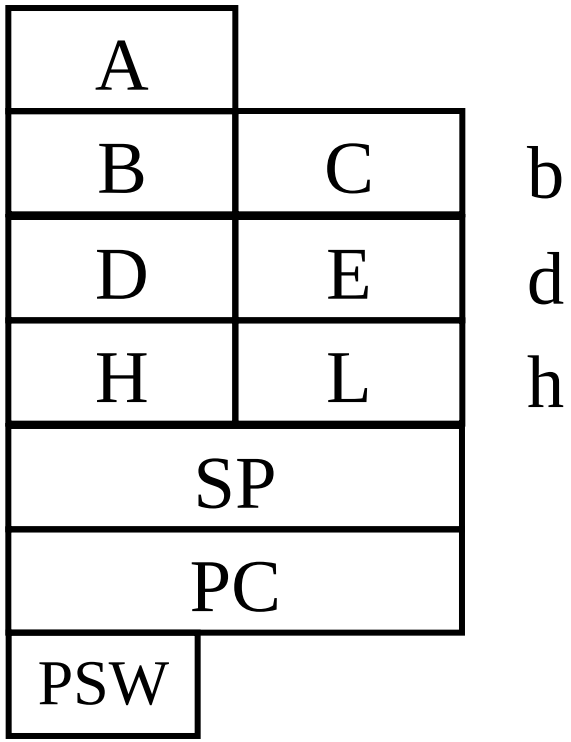
3 Процессоры Intel P6.

- 3.1 Программистская структура.
- 3.2 Форматы команд и особенности формирования эффективного адреса.

- **Знать:** Основные черты архитектур процессоров Intel 8080, Intel 8086, Intel P6:
  - программистские структуры;
  - особенности составов операций и форм представления данных;
  - форматы команд и способы формирования исполнительных адресов ;
  - вычисление физического адреса в процессорах Intel 8086 и Intel P6 (реальный режим).
- **Уметь:**
- **Помнить**: о «расширении» архитектуры процессора по мере увеличения числа функций операционной системы, поддерживаемых аппаратно и находящих отражение в архитектуре процессора (специальные регистры, команды и т.п.).
- **Литература:** [1,5-7,9,14].

1 Микропроцессор КР580ИК80

1.1 Программистская структура (ПС)



Номера 8- и 16- разрядных регистров:

B – 000, C – 001, b – 00;

D – 010, E – 011, b – 01;

H – 100, L – 101, b – 10;

A – 111. Память (M) – 110.

Признаки (PSW):

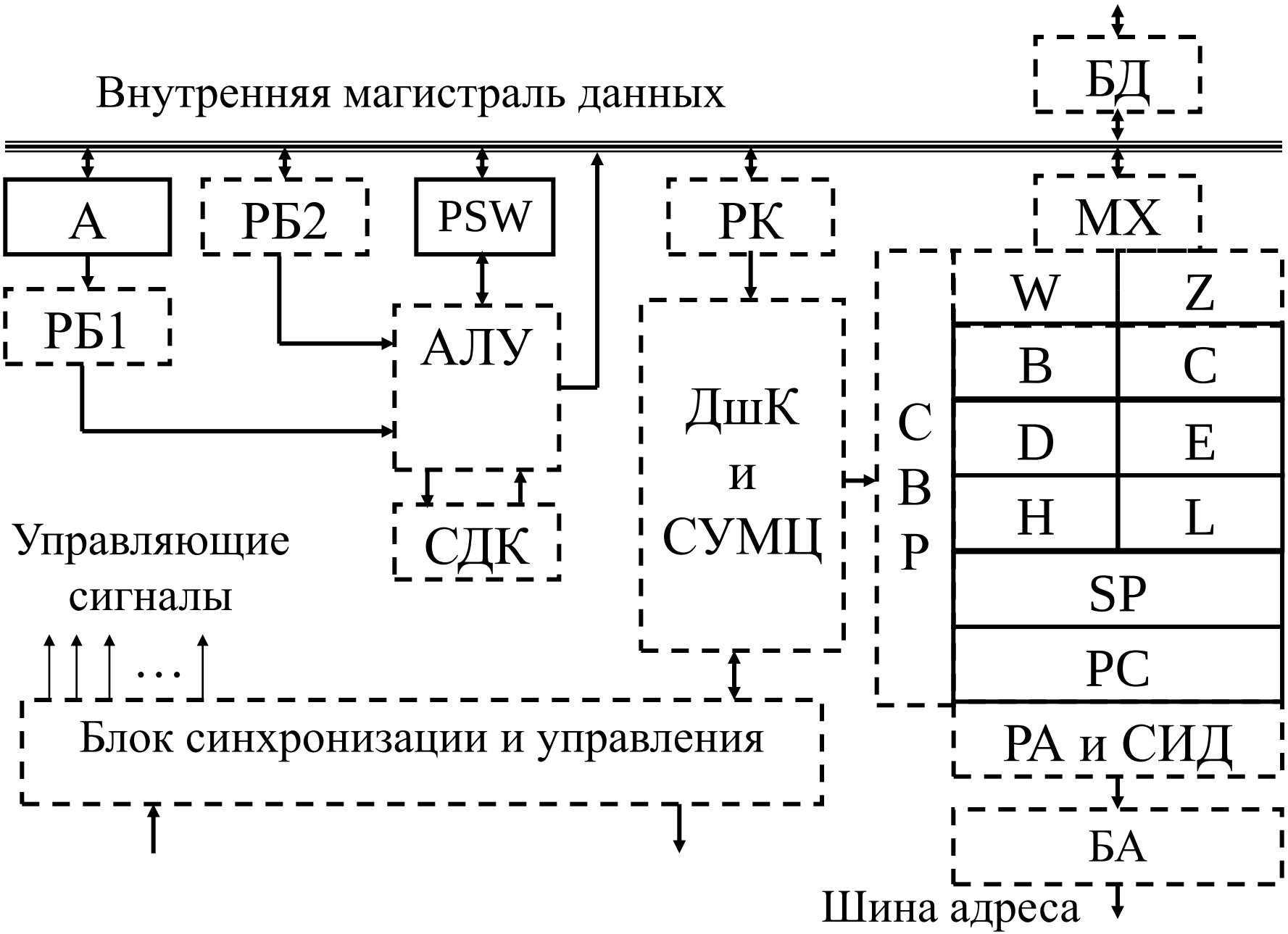
S – знак, Z – ноль,

C – перенос из старшего разряда,

AC – межтетрадный перенос (доступен только командам десятичной коррекции),

P – признак четности результата.

Отображение ПС на аппаратуру МП



Обозначения на схеме МП

- А – аккумулятор, РБ1, РБ2 – буферные регистры;
- PSW – регистр слова состояния процессора;
- АЛУ – арифметико-логическое устройство;
- СДК – схема десятичной коррекции;
- РК – регистр команд; ДшК и СУМЦ – дешифратор команд и схема управления машинным циклом;
- СВР – схема выбора регистра; БД – буфер данных, а БА – буфер адреса; МХ –мультиплексор;
- W, Z дополнительные регистры команды для загрузки второго и третьего байтов команды;
- РА и СИД – регистр адреса и схема инкремента-декремента.

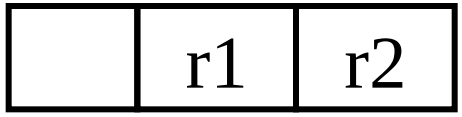
1.2 Особенности состава арифметических операций и представления данных

- Микропроцессорная БИС рассчитана на выполнение логических и простых арифметических операций с 8-разрядными числами в двоичной и десятичной (код 8421) системах счисления, а также операций с двойной разрядностью (с 16-разрядными числами).
- Сложные арифметические операции (умножение, деление) выполняются программно.
- Для программной реализации сложных операций и выполнения операций с двойной разрядностью предусмотрены команды, формирующие и учитывающие перенос из старшего разряда байта.

1.3 Форматы команд и способы адресации

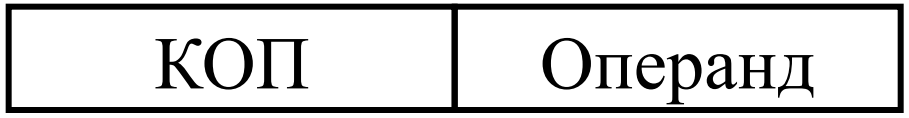
HLT (76), NOP (00), RET (C9)

КОП



MOV B,C (41), MOV B,M (46)

POP D (D1), INC C (0C)



MVI C, d8 (0E «Операнд»)

LXI SP, d16

CALL Adr

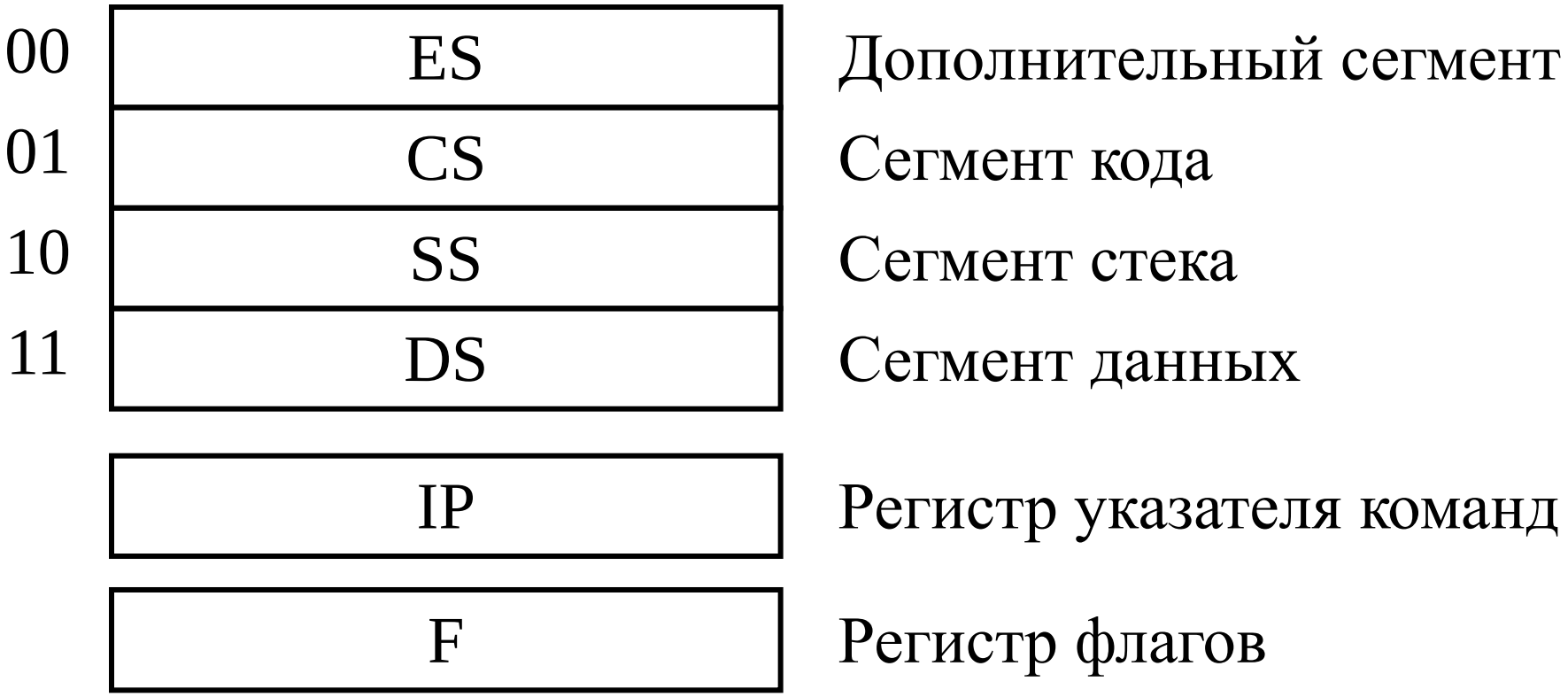
2 Микропроцессор К1810ВМ86

2.1 Программистская структура

000	AH (100)	AL (000)	AX	Аккумулятор
001	CH (101)	CL (001)	CX	Счет
010	DH (110)	DL (010)	DX	Данные
011	BH (111)	BL (011)	BX	База

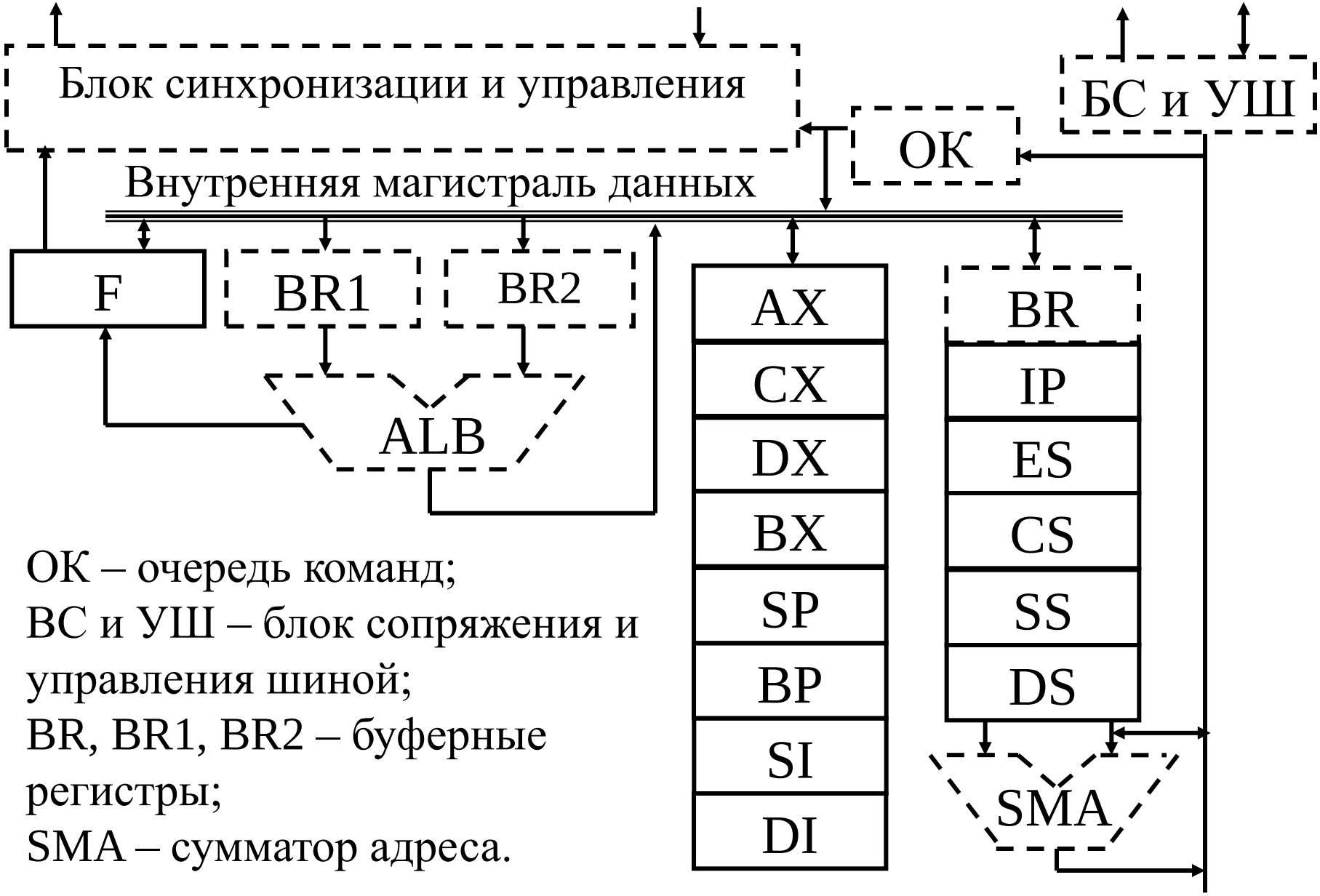
100	SP	Указатель стека
101	BP	Указатель базы
110	SI	Индекс источника
111	DI	Индекс приемника

Регистры ПС



Признаки: перенос (заём) (CF); четность (PF); перенос (заём) из младшей тетрады (AF); ноль (ZF); знак (SF); пошаговое прерывание (TF); разрешение прерывания (IF); направление обработки строк (DF) (от меньших адресов к большим или наоборот); переполнения (OF).

Отображение ПС на аппаратуру МП



ОК – очередь команд;
ВС и УШ – блок сопряжения и управления шиной;
BR, BR1, BR2 – буферные регистры;
SMA – сумматор адреса.

2.2 Представление данных и состав операций

Представление данных

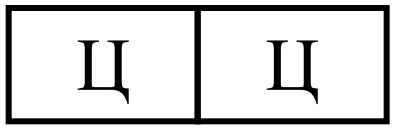
Арифметические операции выполняются над целыми числами в следующих форматах.

Беззнаковый байт

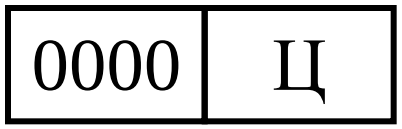
Беззнаковое слово

Знаковый байт

Знаковое слово



Упакованное десятичное



Неупакованное десятичное

Состав операций

- *Операции передачи данных* (MOV, XCHG, XLAT, LEA, LDS, LES, SAHF, LAHF, PUSH, POP, IN, OUT).
- *Арифметические операции* (ADD, ADC, INC, SUB, SBB, DEC, NEG, CMP, MUL, IMUL, DIV, IDIV).
- *Логические операции* (NOT, AND, OR, XOR, TEST).
- *Операции сдвигов* (RCL, RCR, SHL, SHR, SAL, SAR).
- *Операции передачи управления* (JAMP, JNZ, JZ,…, JCXZ, CALL, RET, LOOPZ, LOOPNZ, INT, IRET).
- *Цепочечные операции* (MOVS, CMPS, SCAS, LODS, STOS).
- *Операции управления микропроцессором* (CTC, CLC CMC, STD, CLD, STI, CLI, HLT, WAIT, LOCK, NOP, ESC).

Установка признаков

Команды	Признаки					
	OF	SF	ZF	AF	DF	CF
ADD, ADC, SUB, SBB, NEG, CMP	+	+	+	+	+	+
INC, DEC	+	+	+	+	+	-
MUL, IMUL	+	н	н	н	н	+
DIV, IDIV	н	н	н	н	н	н
AND, OR, XOR, TEST	0	+	+	н	+	0
SHL, SHR, SAL,SAR	+	+	+	н	+	+
RCL, RCR	+	-	-	-	-	+

«+» – установка по результатам операции, «-» – сохранение ранее установленного, «н» – неопределенное значение, «0» – установка в ноль.

Префиксы команд

- *Префикс повторения* (*REP)* вызывает повторное действие цепочечной команды над следующими элементами. Повторение рассчитано на максимальную длину цепочки (64К байт) и заканчивается по одному или двум условиям. Префикс имеет пять модификаций: REP (применяется с командами MOVS и STOS), REPE, REPZ, REPNE, REPNZ (применяется с командами CMPS, SCAS).
- *Префикс блокировки (LOCK)* вызывает в микропроцессоре формирование активного сигнала блокировки (^LOCK) на время выполнения следующей команды (сигнал используется арбитром шины для запрещения доступа к ней других устройств).
- *Префикс замены сегмента* определяет сегментный регистр, который будет использован в следующей за ним команде для формирования физического адреса памяти.

2.3 Выбор сегмента и формирование физического адреса

- Сегмент не указывается в команде, а выбирается одним из двух следующих способов.

Выбор сегмента по умолчанию

(в зависимости от типа адресуемого элемента)

Цель обращения к памяти	**База по умолчанию**	**Источник смещения**
Выборка команд	CS	IP
Операции со стеком	SS	SP
Выборка данных	DS (или CS,ES,SS)	EA
Выборка данных и база в BR	SS (или CS,DS,ES)	EA
Адресация строки источника	DS (или CS,ES,SS)	SI
Адресация строки приемника	ES	DI

Задание сегмента с помощью байта префикса замены сегмента

- Байт SR префикса располагается перед командой и задает используемый в ней сегмент.

001	sr	110

sr	Сегмент
00	ES
01	CS
10	SS
11	DS

Формирование физического адреса

Логический адрес

Сегмент (b) Смещение (a)

3	4	5	6

7	8	9	A

	3	4	5	6	0
+		7	8	9	A
	3	B	D	F	A

A_F=b*16+a

Физический адрес

2.4 Способы адресации и форматы команд

Использование кода операции

- Способ вычисления адреса может быть указан в коде операции (КОП) или задан специальным байтом – постбайтом адресации.
- КОП может также содержать следующие поля:
  - D – бит направления
    - (D=0 – «внутрь регистра », D=1 – «из регистра»);
  - W – бит ширины
    - (W=0 – команда работает с байтами, W=1 – со словами).

Постбайт адресации

- Постбайт адресации располагается после кода операции и имеет следующий формат.

md – поле режима указывает режим адресации и определяет, как используется содержимое поля r/m при нахождении операнда;

r/m – поле «регистр/память», если md=11, то поле содержит номер регистра (000 – AX, …, 111 – DI);

reg – поле регистра определяет 8- или 16-разрядный регистр, в котором содержится операнд: 000 – AL (при W=0) или AX (при W=1) , …, 111 – BH (при W=0) или DI (при W=1).

Режимы адресации

- md=00 – операнд находится в памяти и используется косвенная адресация через регистры BX, BP, SI, DI, а при r/m=110 – прямая адресация.
- md=01 – операнд находится в памяти и используется косвенная адресация через регистры BX, BP, SI, DI и 8-разрядное смещение d8 (со знаковым расширением до 16 разрядов).
- md=10 – операнд находится в памяти и используется косвенная адресация через регистры BX, BP, SI, DI и 16-разрядное смещение d16.
- md=11 – операнд находится в регистрах общего назначения.

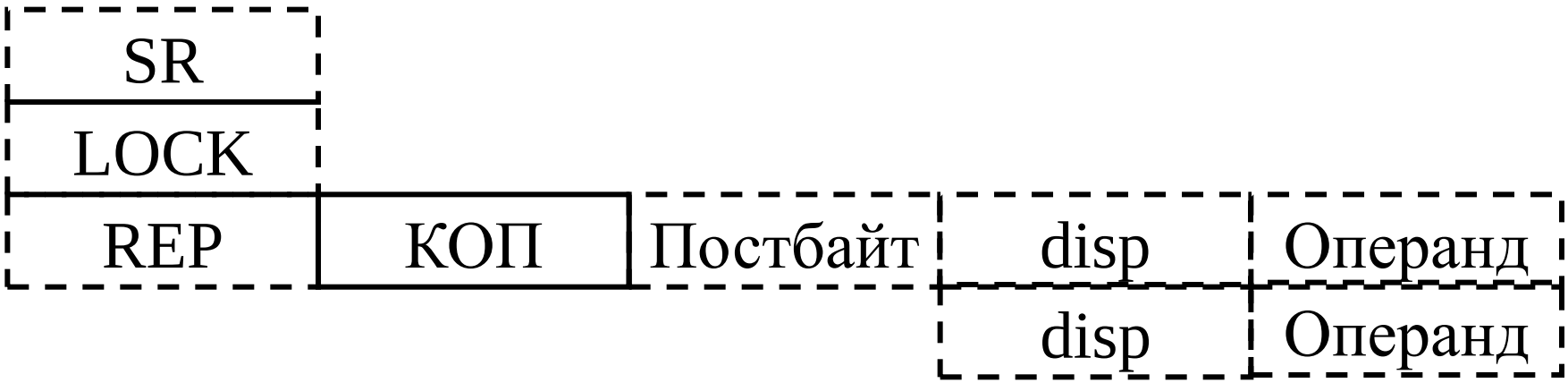
Вычисление адреса

r/m, (reg)	md=00	md=01	md=10	md=11	
				W=0	W=1
000	BX+SI	BX+SI+d8	BX+SI+d16	AL	AX
001	BX+DI	BX+DI+d8	BX+DI+d16	CL	CX
010	BP+SI	BP+SI+d8	BP+SI+d16	DL	DX
011	BP+DI	BP+DI+d8	BP+DI+d16	BL	BX
100	SI	SI+d8	SI+d16	AH	SP
101	DI	DI+d8	DI+d16	CH	BP
110	d16	BP+d8	BP+d16	DH	SI
111	BX	BX+d8	BX+d16	BH	DI

Способы адресации

- Неявная адресация
- Непосредственная адресация
- Регистровая адресация
- Прямая (абсолютная) адресация
- Косвенная регистровая адресация
- Базовая адресация
- Индексная адресация
- Базовая индексная адресация
- Автоинкрементная и автодекрементная адресация (запись в стек и чтение из стека, адресация цепочек).
- Относительная адресация
- Адресация портов ввода вывода (прямая с указанием адреса порта или косвенная регистровая)

Форматы команд



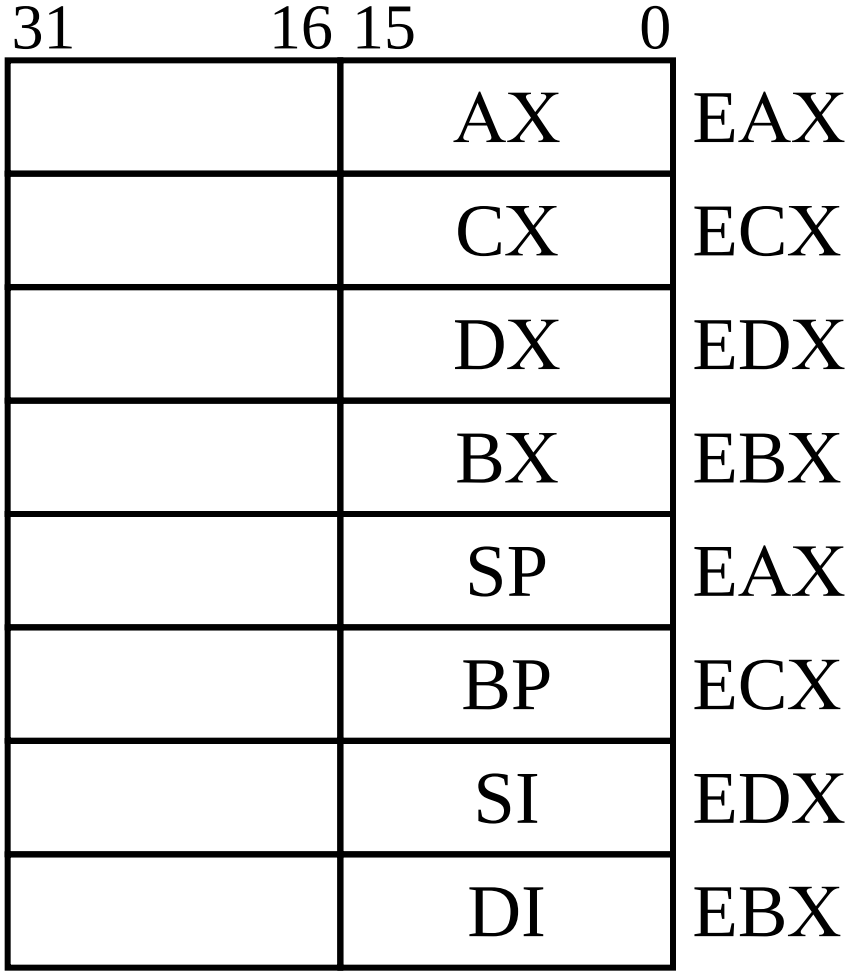
- Например, команда **lock add cs: [bp+di+100h], 77h** имеет шестнадцатеричное представление:
- **F0 2E 80 83 0100 77** (F0 – младший байт в ОП). Здесь F0 – префикс блокировки (lock), 2E –префикс замены сегмента (cs:), 80 – код команды (операции) (add), 83 – постбайт (задает способ формирования адреса: bp+di+disp), 0100 – смещение (100h) и 77 – непосредственный операнд.

3 Процессоры Intel P6

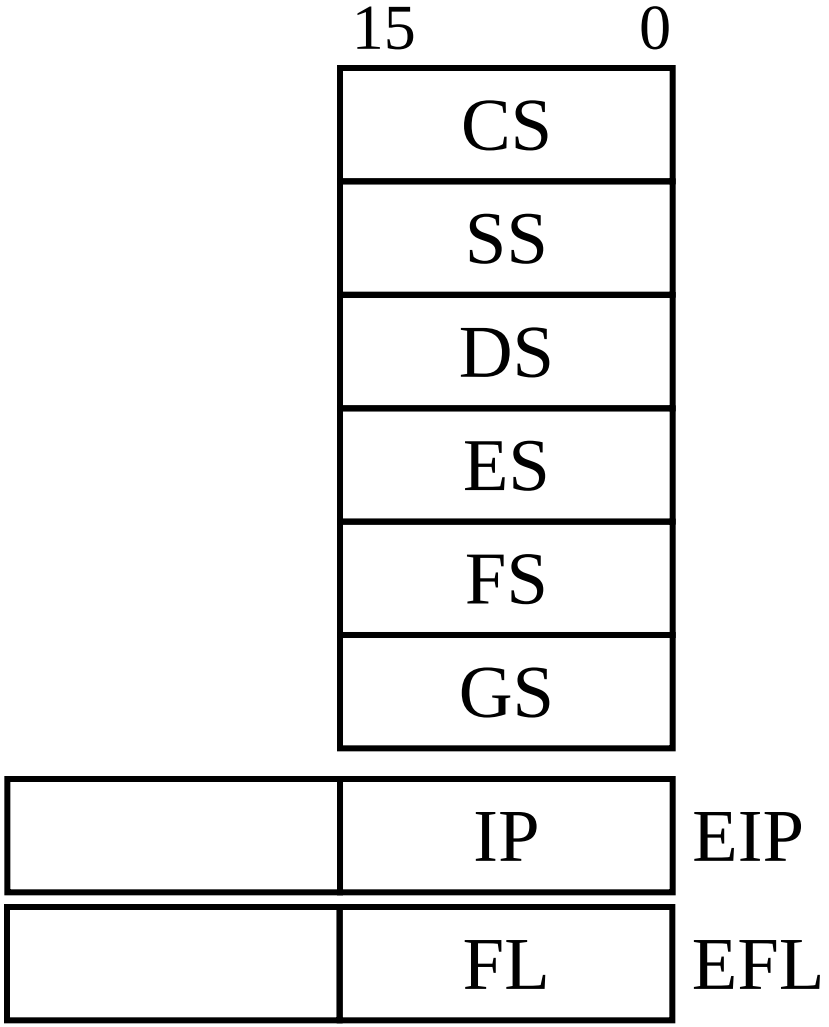
3.1 Программистская структура

Регистры базовой архитектуры

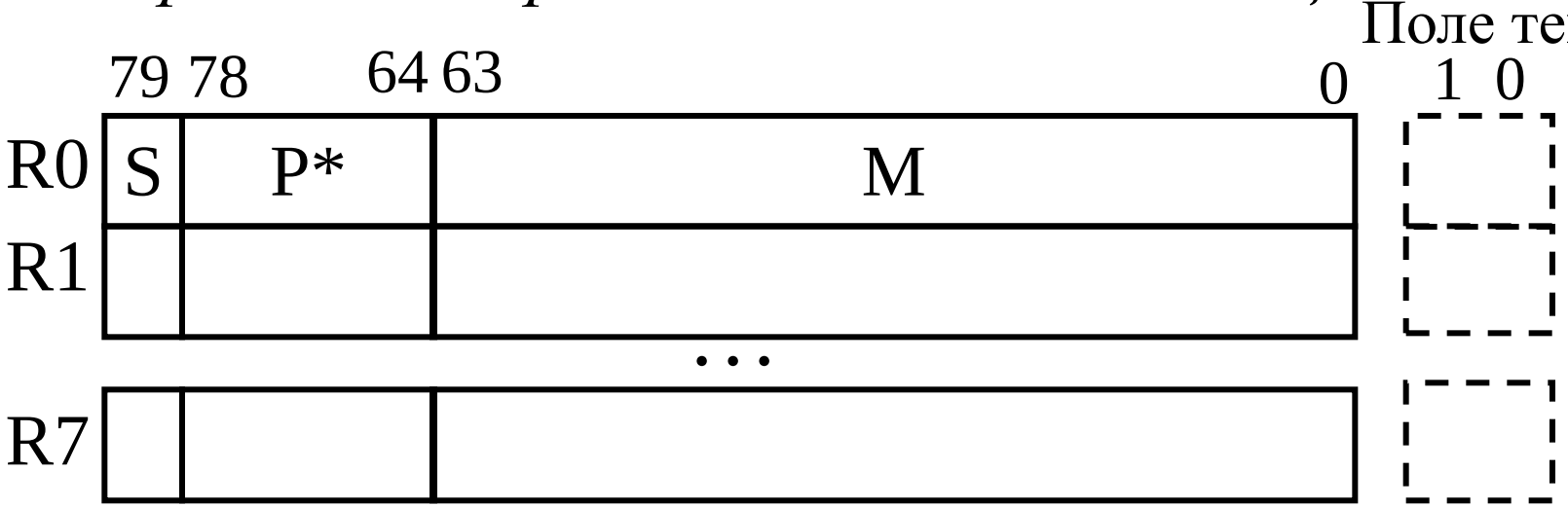
Регистры общего назначения

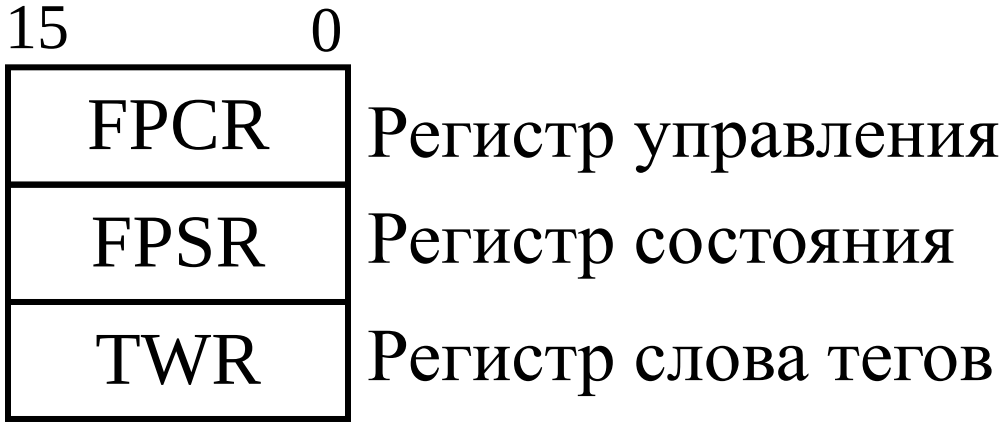


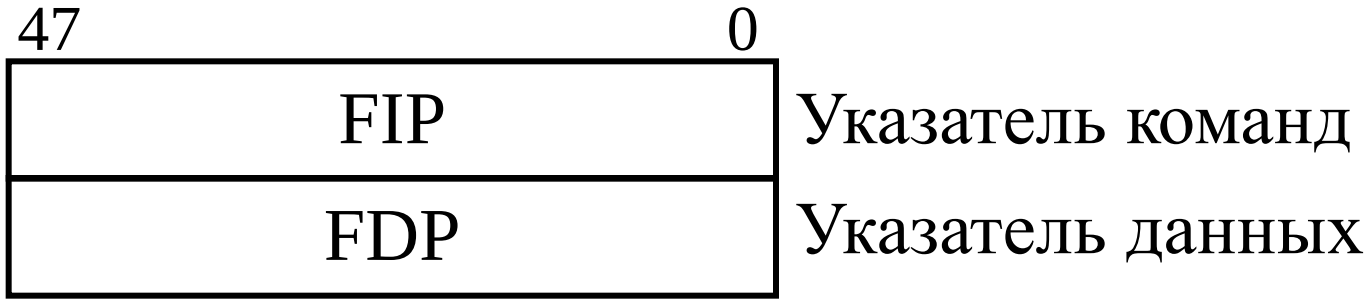
Сегментные регистры



Регистры блока обработки чисел с плавающей запятой

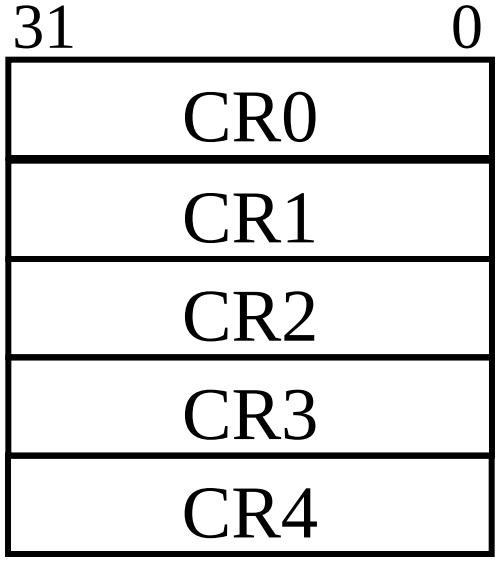




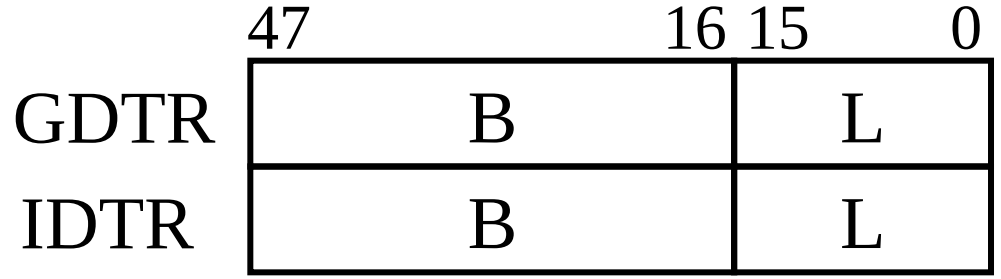


Регистры системного уровня

Управляющие регистры



Системные регистры адреса



GDTR – регистр адреса глобальной дескрипторной таблицы.
IDTR – регистр адреса таблицы дескрипторов прерываний.

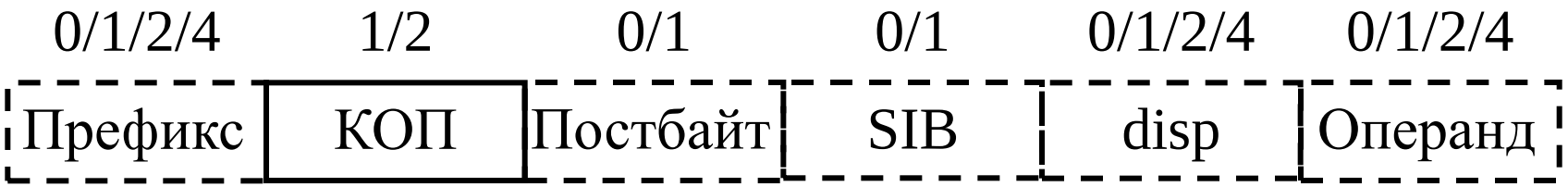
Регистры системных сегментов

LDTR – регистр селектора локальной дескрипторной таблицы.
TR – регистр селектора сегмента состояния задачи (TSS).

Регистры отладки (DR0-DR7)

3.2 Форматы команд и особенности формирования эффективного адреса

Обобщенный формат команды



Любое из полей (кроме КОП) в конкретной команде может отсутствовать. Цифры над полем указывают сколько байт оно может занимать. Например, поле кода операций (КОП) занимает либо один, либо два байта. Префиксы, если они есть, располагаются в младших, а поле «Операнд» (если оно есть) - в старших байтах команды.

Префиксы

- *Префикс команды*. Это либо префикс повторения REP, либо префикс блокировки LOCK.
- *Префикс замены сегмента*. Задает сегментный регистр, участвующий в формировании эффективного адреса.
- *Префикс размера операнда* (66h). Вместе с битом D в дескрипторе, загруженном в теневую часть регистра сегмента кода CS, определяет размерность используемых в команде операндов. Если этот префикс есть и D=0 или если этого префикса нет и D = 1, то команда работает с 32-разрядными операндами. В противных случаях команда работает с 16-разрядными операндами.
- *Префикс размера адреса* (67h). Аналогичен предыдущему префиксу, но определяет размерность эффективного адреса (16 или 32), формируемого командой.
- Порядок следования префиксов значения не имеет.

Код операции

- Код операции (КОП) чаще всего занимает один байт, но может дополняться битами из постбайта.
- В ряде команд КОП содержит поля reg (номер регистра общего назначения) и sreg (номер сегментного регистра), в которых принята кодировка регистров, приведенная ниже в таблицах.
- Первый бит команды определяет тип команды (одно- или двухадресная).
- КОП содержит также бит d, который задает выбор регистров, используемых в качестве источника и приемника информации при выполнении ряда арифметических и логических операций типа регистр-регистр.

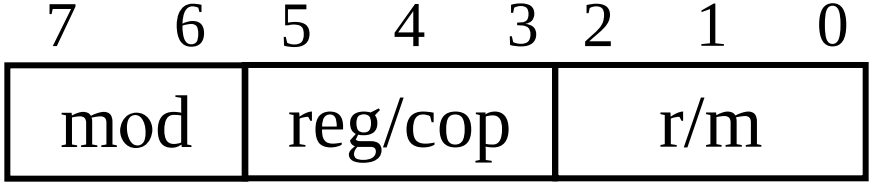
Кодирование регистров

Поле reg	**Разрядность операндов**		
	8	**16**	**32**
000	AL	AX	EAX
001	CL	CX	ECX
010	DL	DX	EDX
011	BL	BX	EBX
100	AH	SP	ESP
101	CH	BP	EBP
110	DH	SI	ESI
111	BH	DI	EDI

Поле sreg	**Сегментный регистр**
000 (00)	ES
001 (01)	CS
010 (10)	SS
011 (11)	DS
100	FS
101	GS

Постбайт адресации

- Постбайт адресации располагается после кода операции и имеет следующий формат.



mod – поле режима указывает режим адресации и определяет, как используется содержимое поля r/m при нахождении операнда;

r/m – поле «регистр/память», если mod=11, то поле содержит номер регистра, иначе адрес формируется с использованием 8-, 16- или 32-разрядного смещения;

reg – поле регистра определяет 8-, 16- или 32-разрядный регистр, в котором содержится операнд.

Формирование 16-разрядного адреса (EA)

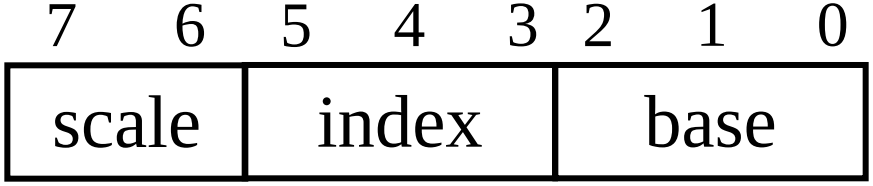
Поле r/m	**Поле mod**		
	00	**01**	**10**
000	DS:[BX+SI]	DS:[BX+SI+d8]	DS:[BX+SI+d16]
001	DS:[BX+DI]	DS:[BX+DI+d8]	DS:[BX+DI+d16]
010	SS:[BP+SI]	SS:[BP+SI+d8]	SS:[BP+SI+d16]
011	SS:[BP+DI]	SS:[BP+DI+d8]	SS:[BP+DI+d16]
100	DS:[SI]	DS:[SI+d8]	DS:[SI+d16]
101	DS:[DI]	DS:[DI+d8]	DS:[DI+d16]
110	DS:[d16]	DS:[BP+d8]	DS:[BP+d16]
111	DS:[BX]	DS:[BX+d8]	DS:[BX+d16]

Формирование 32-разрядного адреса (EA)

Поле r/m	**Поле mod (байт SIB отсутствует)**		
	00	**01**	**10**
000	DS:[EAX]	DS:[EAX+d8]	DS:[EAX+d32]
001	DS:[ECX]	DS:[ECX+d8]	DS:[ECX+d32]
010	DS:[EDX]	DS:[EDX+d8]	DS:[EDX+d32]
011	DS:[EBX]	DS:[EBX+d8]	DS:[EBX+d32]
100	См. табл. ниже	См. табл. ниже	См. табл. ниже
101	DS:[d32]	SS:[ESP+d8]	SS:[ESP+d32]
110	DS:[ESI]	DS:[ESI+d8]	DS:[ESI+d32]
111	DS:[EDI]	DS:[EDI+d8]	DS:[EDI+d32]

Байт SIB

- Байт адресации SIB располагается после кода операции (постбайта) и имеет следующий формат.



- При наличии байта SIB эффективный адрес вычисляется по формуле:

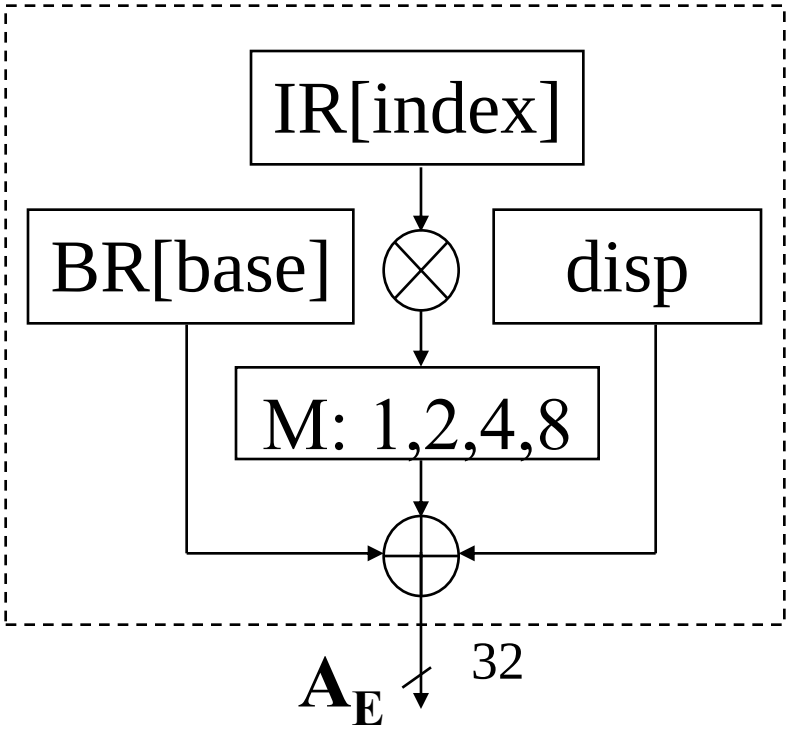
$$\mathbf{A_E=F\times(IR)+(BR)+disp,}$$

где $F=2^{scale}$, причем, поле scale имеет следующие значения: 00, 01, 10 и 11; (IR) – содержимое индексного регистра, определяемого полем index; (BR) – содержимое базового регистра, определяемого полем base; Смещение disp может отсутствовать.

Кодирование регистров в байте SIB

Поле index	**Ин-декс**	**Поле base**	**База**
000	EAX	000	EAX
001	ECX	001	ECX
010	EDX	010	EDX
011	EBX	011	EBX
100	-	100	ESP
101	EBP	101	EBP (d32)
110	ESI	110	ESI
111	EDI	111	EDI

Схема вычисления эффективного адреса



Формирование адреса EA (есть байт SIB)

Поле base	**Поле mod (байт SIB присутствует, r/m=100)**		
	00	**01**	**10**
000	DS:[EAX+IR*F]	DS:[EAX+IR*F+d8]	DS:[EAX+IR*F+d32]
001	DS:[ECX+IR*F]	DS:[ECX+IR*F+d8]	DS:[ECX+IR*F+d32]
010	DS:[EDX+IR*F]	DS:[EDX+IR*F+d8]	DS:[EDX+IR*F+d32]
011	DS:[EBX+IR*F]	DS:[EBX+IR*F+d8]	DS:[EBX+IR*F+d32]
100	SS:[ESP+IR*F]	SS:[ESP+IR*F+d8]	SS:[ESP+IR*F+d32]
101	DS:[d32+IR*F]	SS:[EBP+IR*F+d8]	SS:[EBP+IR*F+d32]
110	DS:[ESI+IR*F]	DS:[ESI+IR*F+d8]	DS:[ESI+IR*F+d32]
111	DS:[EDI+IR*F]	DS:[EDI+IR*F+d8]	DS:[EDI+IR*F+d32]

Поля disp и операнд

- Назначение этих полей очевидно и не требует пояснений.

Пример команды

- Благодаря использованию байта SIB возможных вариантов команд с 32-разрядным эффективным адресом значительно больше, чем при 16-разрядном эффективном адресе. Например возможна команда

  mov eax, [2*eax+eax+7].

- В шестнадцатеричной системе счисления эта команда имеет следующее представление:

  66 67 8B 44 40 07.

- Здесь 66 – префикс размера операнда, 67 – префикс размера адреса, 8B – КОП, 44 – постбайт, 40 – SIB, 07 – disp8.