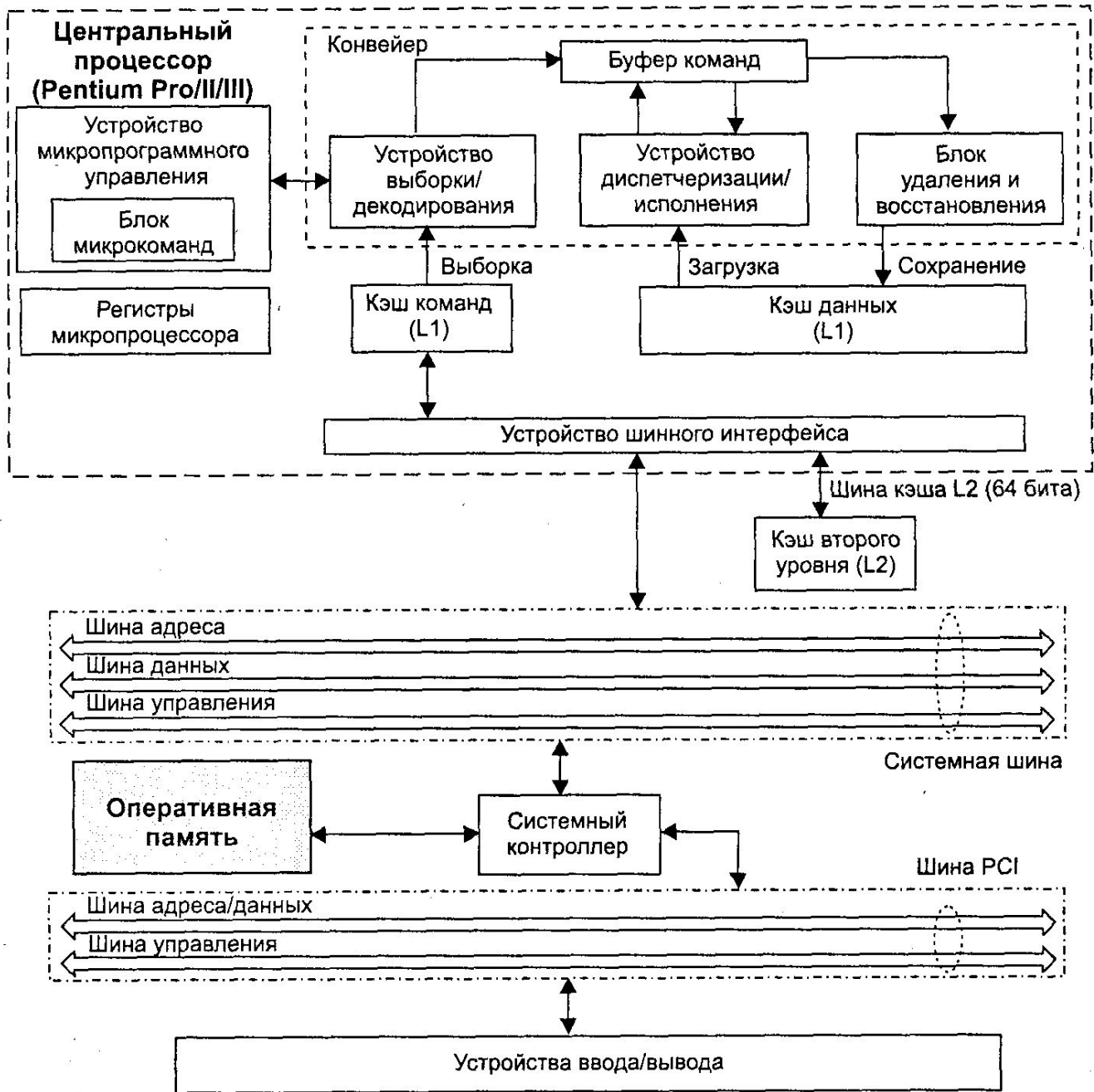
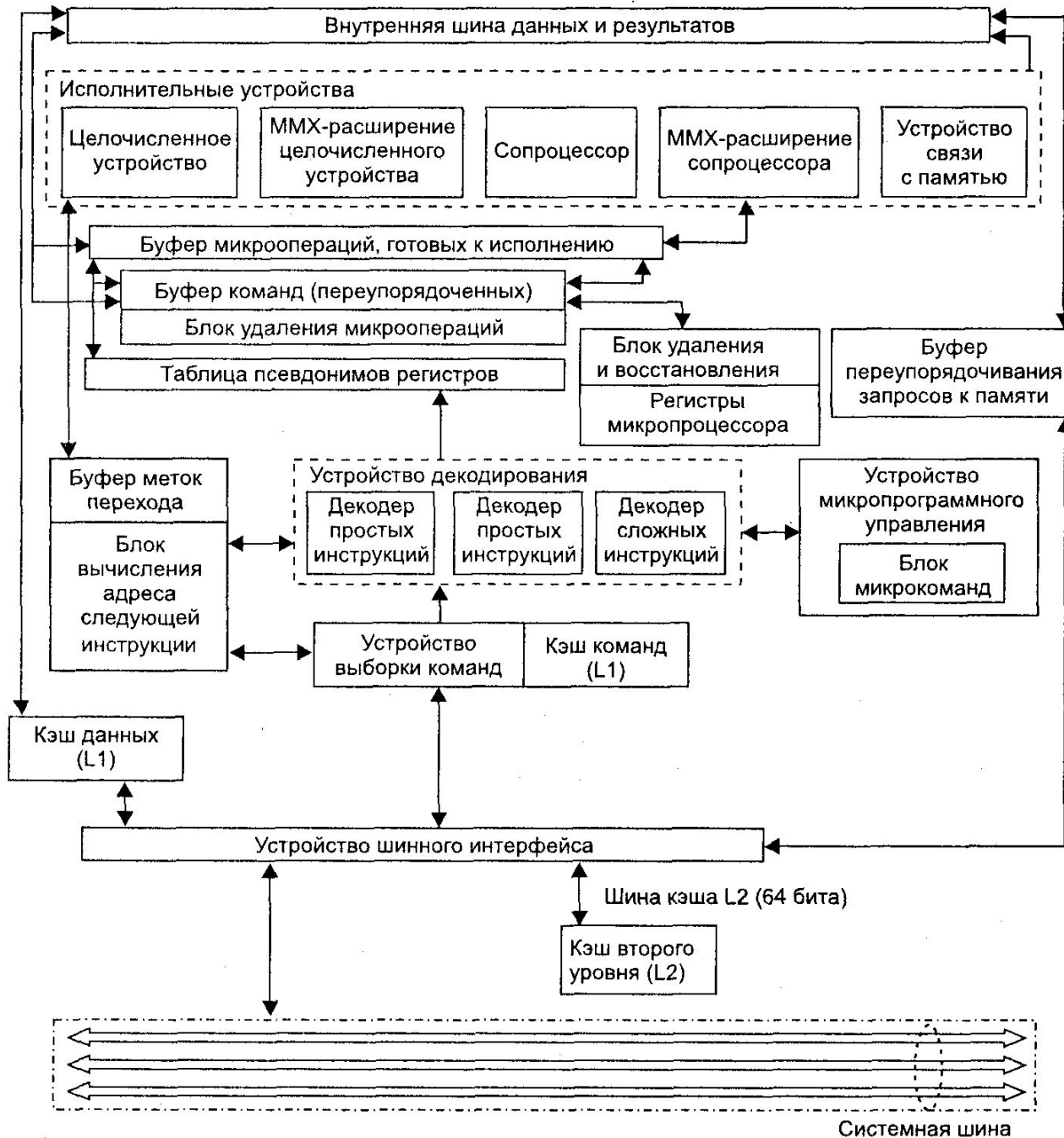


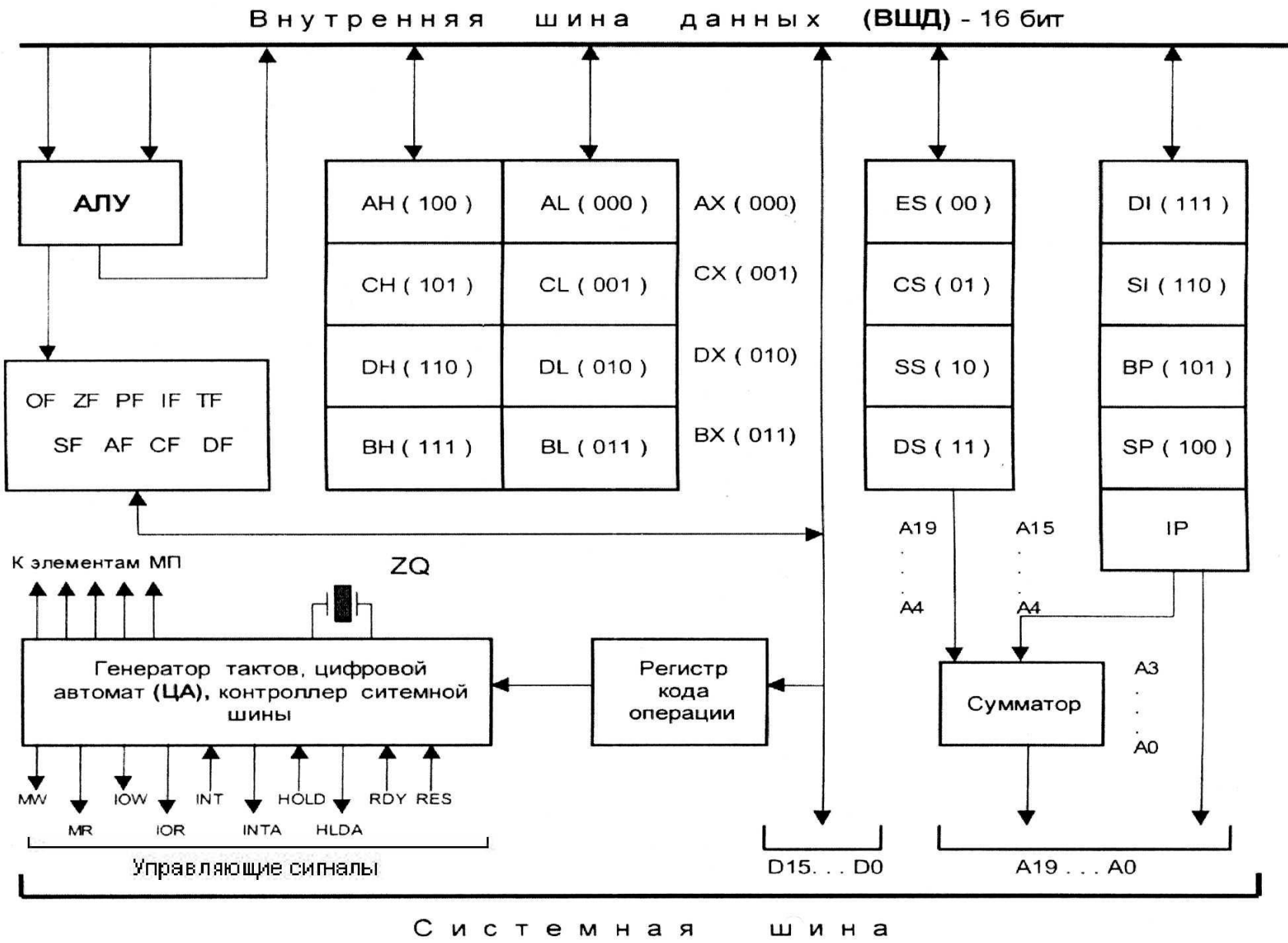
Архитектура процессора **i8086**

- 1. Архитектура центрального процессора.**
- 2. Оперативная память.**
- 3. Программная модель процессора i8086.**
- 4. Система команд процессора i8086.**
- 5. Безадресные команды.**
- 6. Одноадресная регистровая модификация.**
- 7. Одноадресная общая модификация.**
- 8. Двухадресная.**
- 9. Непосредственный операнд. Регистровая модификация.**
- 10. Непосредственный операнд. Общая модификация.**
- 11. Этапы выполнения машинной команды**



Центральный процессор (Pentium Pro/III/III)





RS: A₁₉A₁₈A₁₇A₁₆A₁₅A₁₄A₁₃A₁₂A₁₁A₁₀A₉A₈A₇A₆A₅A₄0000

+

EA: 0000 A₁₅A₁₄A₁₃A₁₂A₁₁A₁₀A₉A₈A₇A₆A₅A₄A₃A₂A₁A₀

A: A₁₉A₁₈A₁₇A₁₆A₁₅A₁₄A₁₃A₁₂A₁₁A₁₀A₉A₈A₇A₆A₅A₄A₃A₂A₁A₀

1CDO
+ F156

2BE56

Номера байтов оперативной памяти

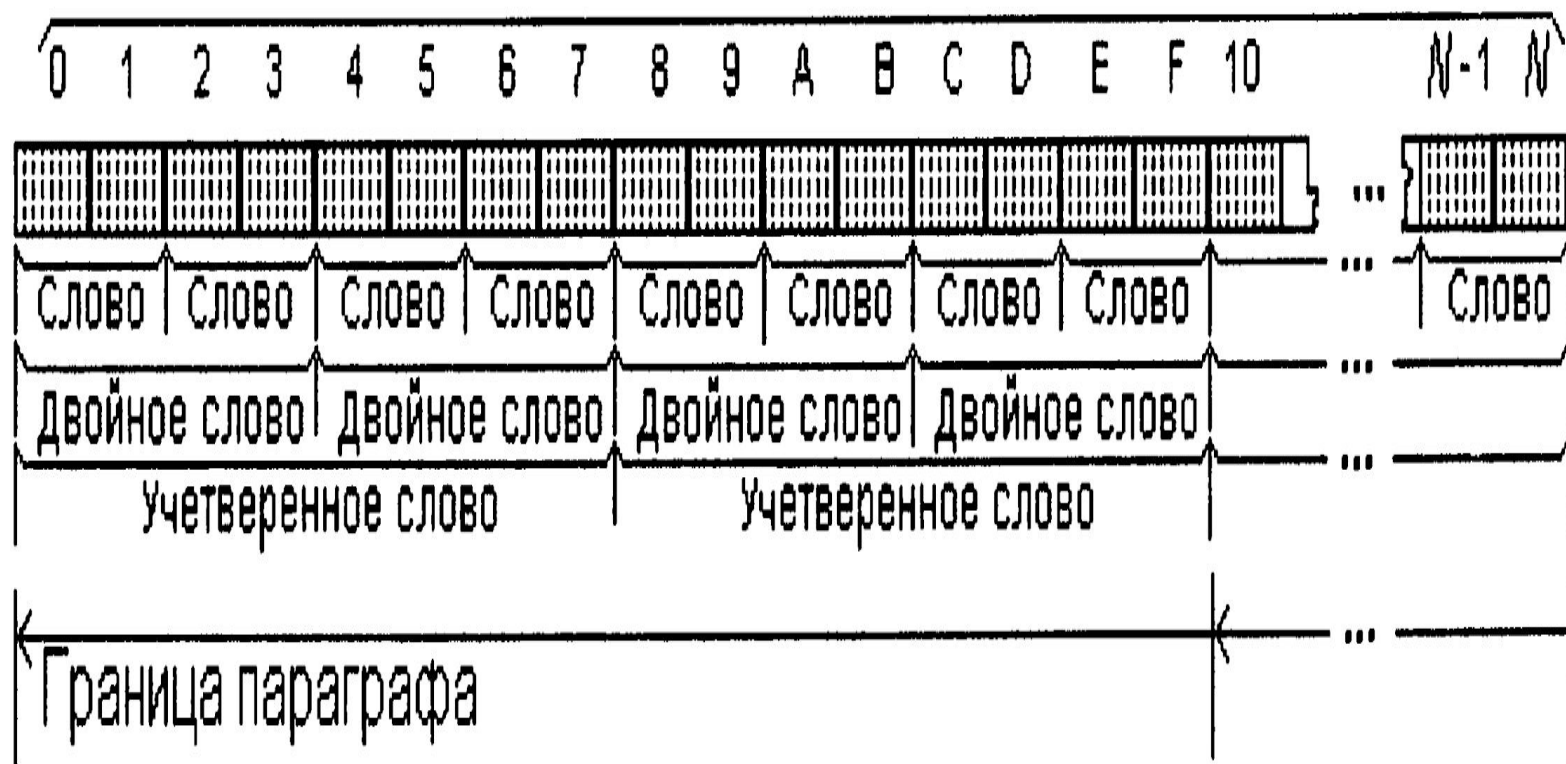


Рис. 4.1. Нумерация байтов и стандартные поля оперативной памяти

Длина поля 4 байта, номер начального байта 5

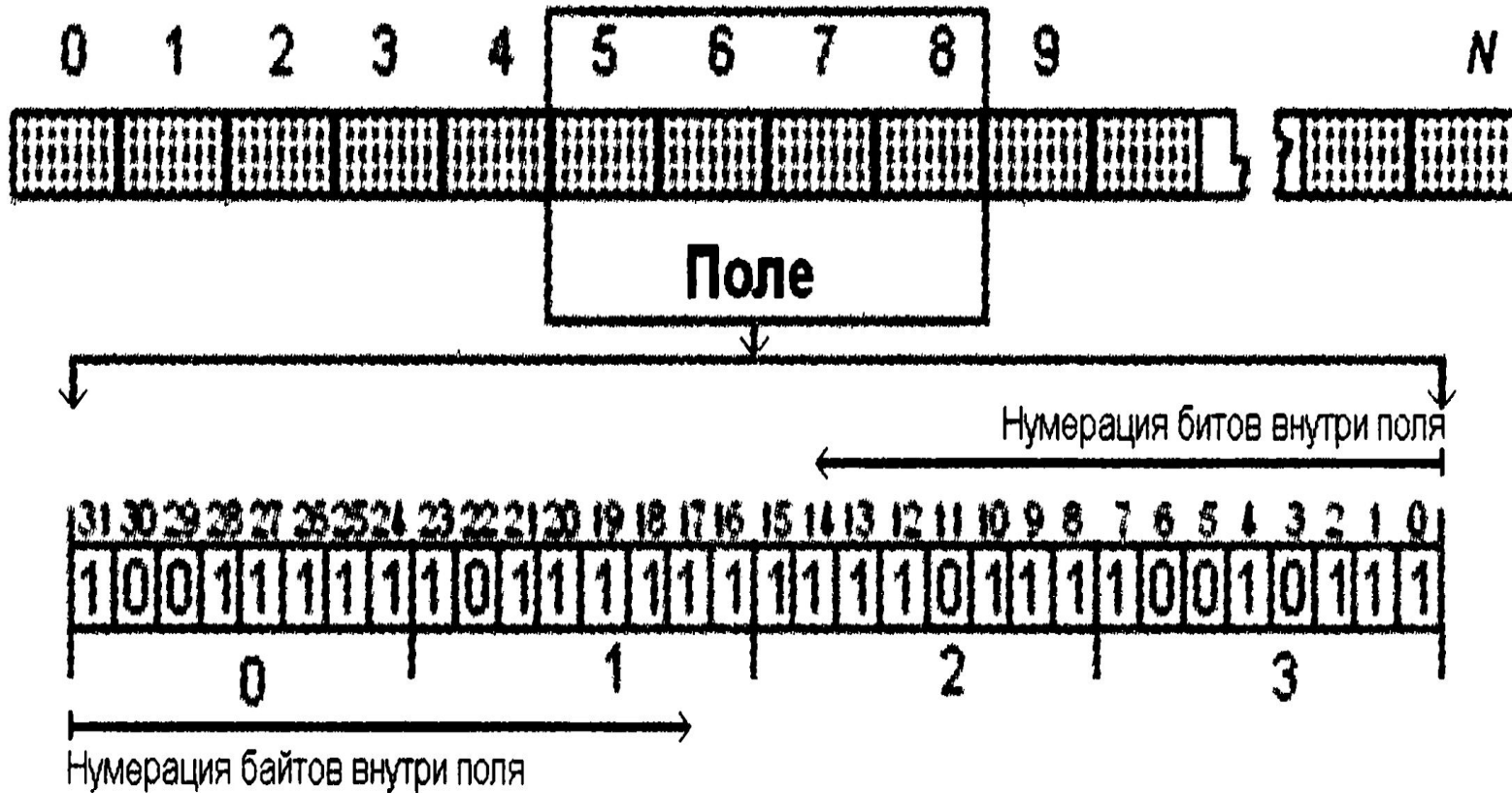


Рис. 4.2. Поле оперативной памяти



Рис. 4.3. Фрагмент двадцатиразрядной шины



Рис. 4.4. Адреса байтов памяти в компьютере с двадцатиразрядной адресной шиной

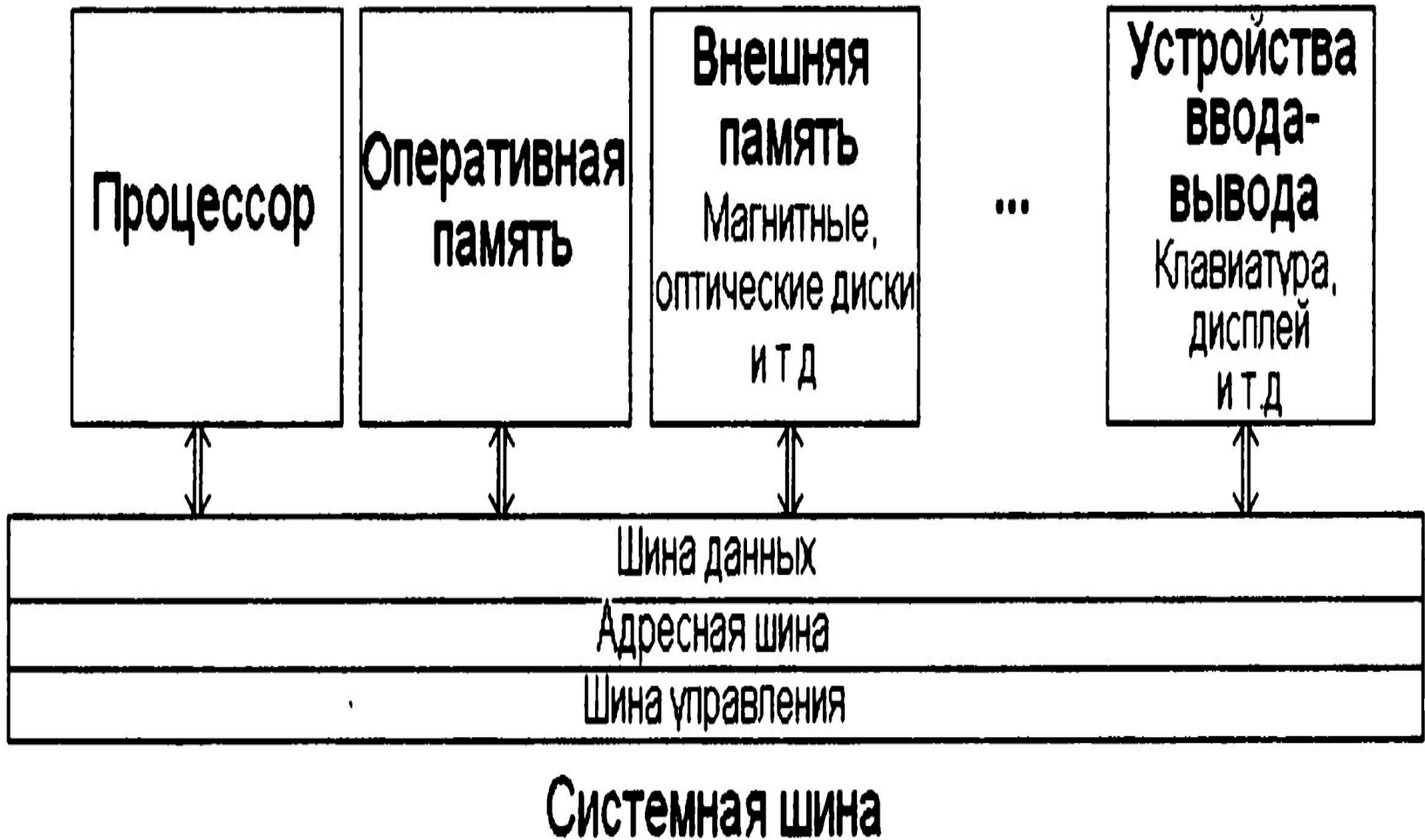


Рис. 4.5. Упрощенная схема компьютера с шинной архитектурой

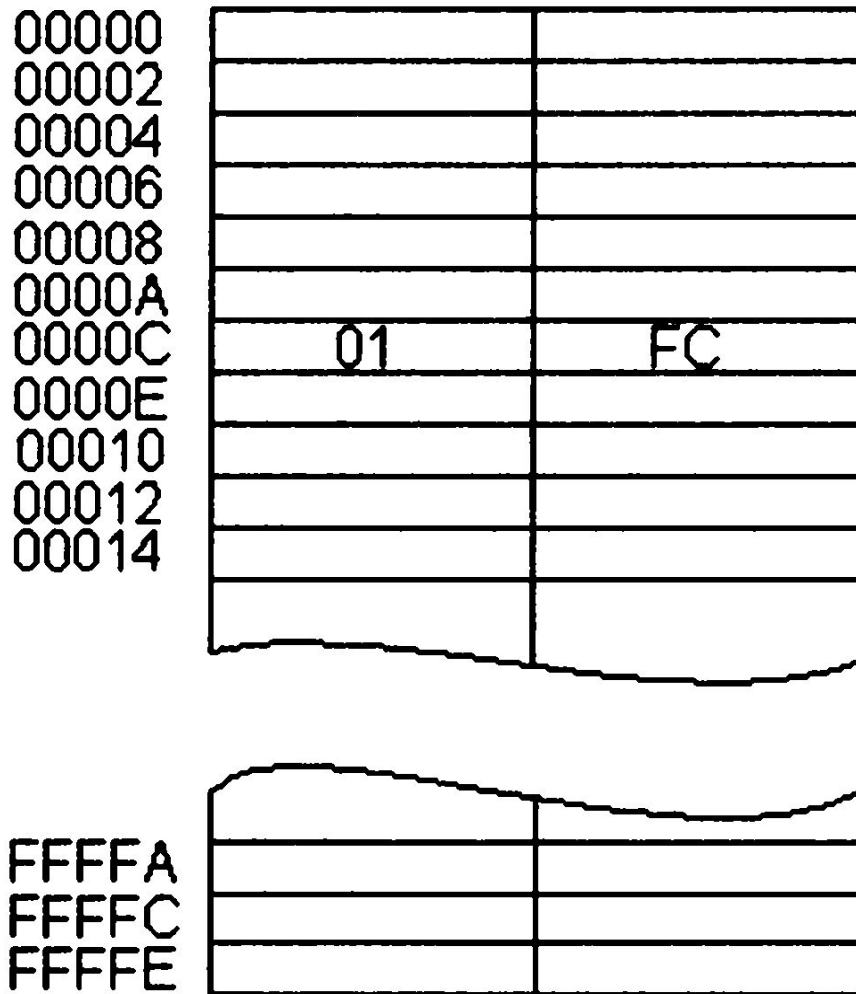
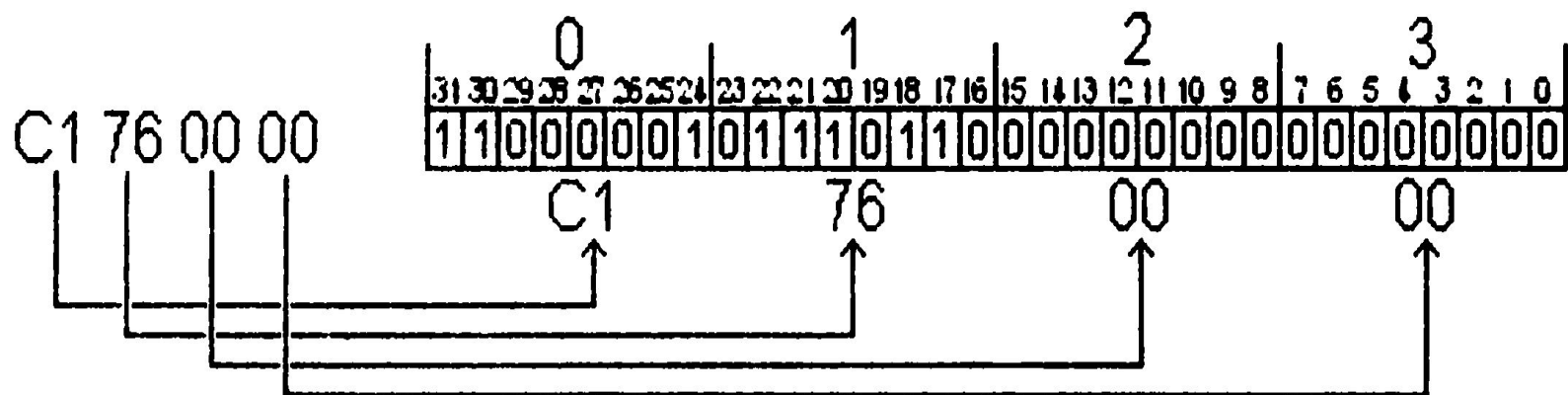
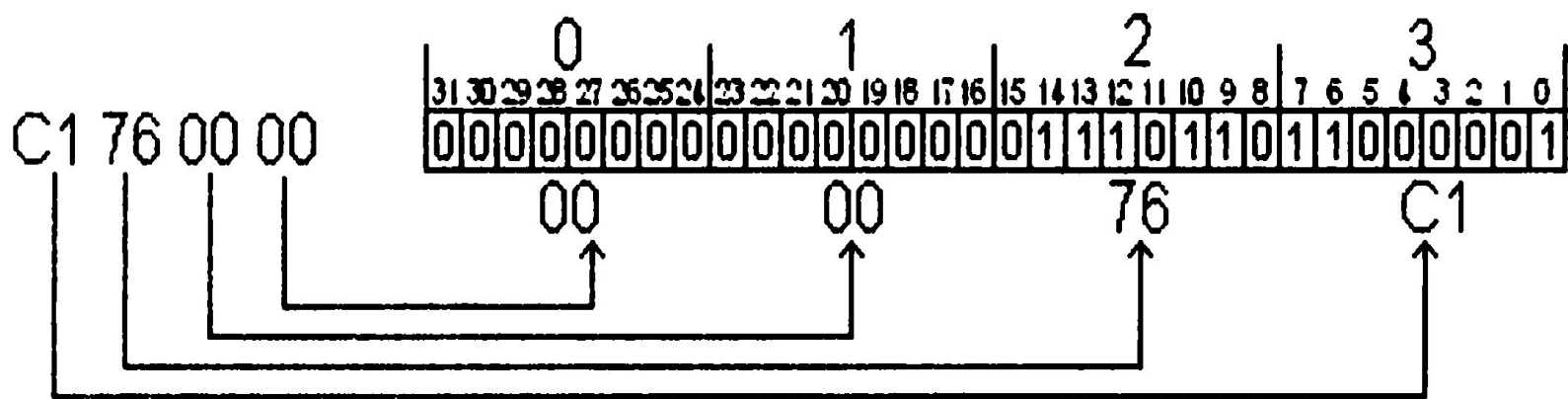


Рис. 4.6. Программная модель оперативной памяти



a



б

Рис. 4.7. Порядок записи кода в поле: а — прямой; б — обратный

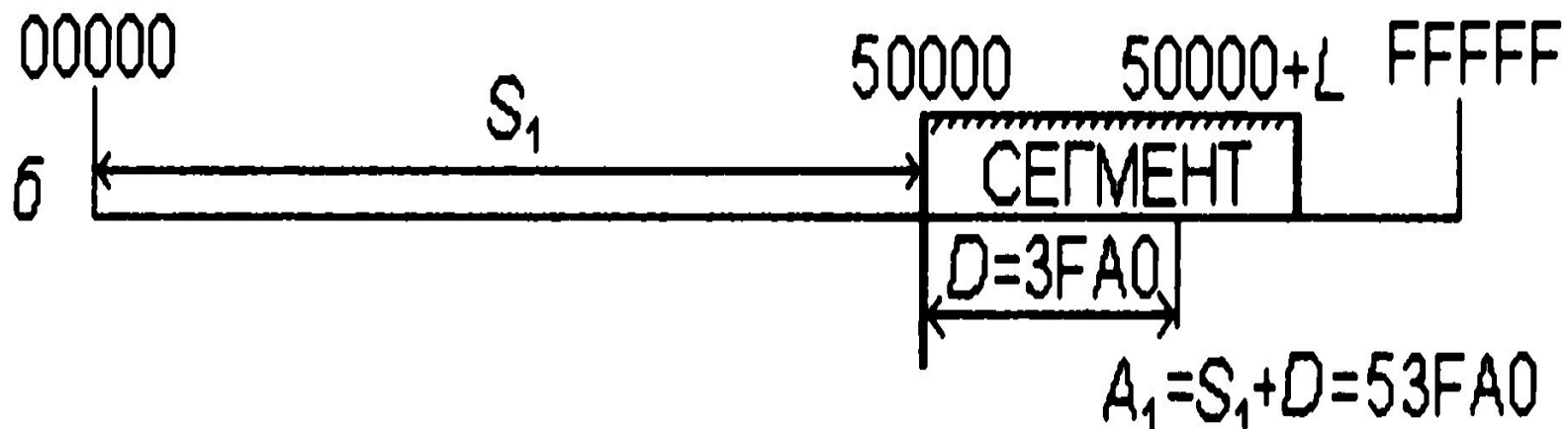
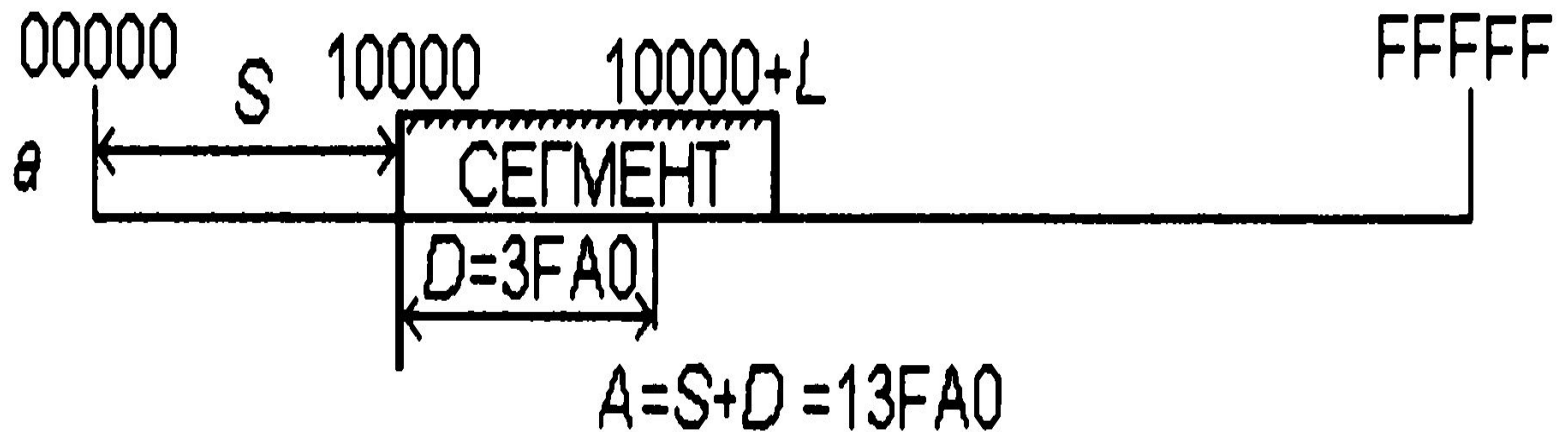


Рис. 4.8. Перемещение сегмента в оперативной памяти

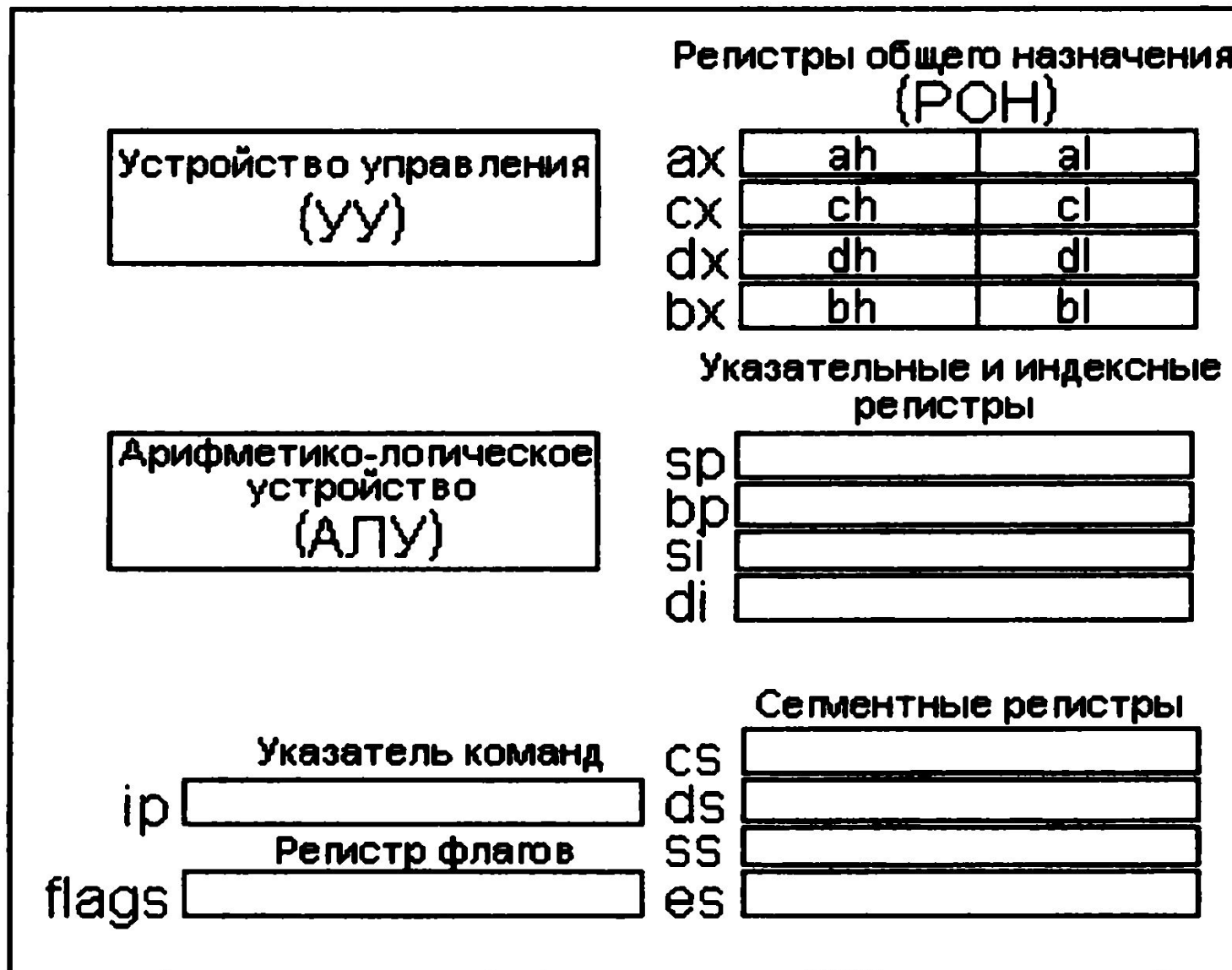
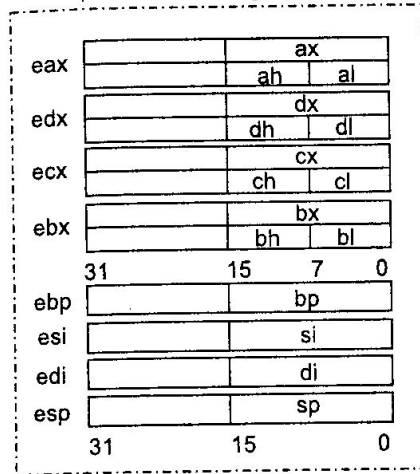
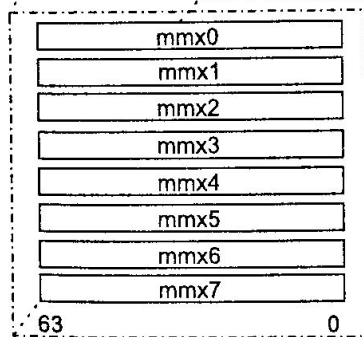
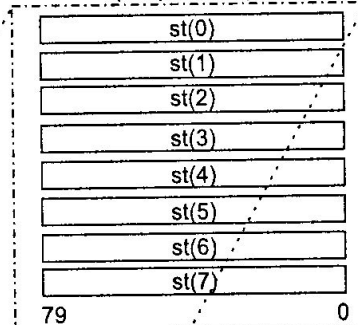


Рис. 4.9. Программная модель процессора i8086

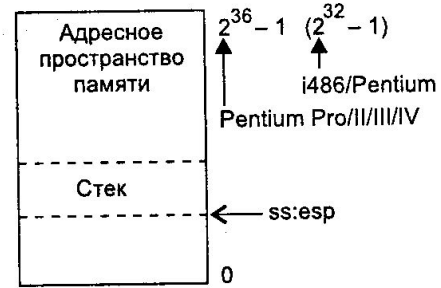
Регистры общего назначения
целочисленного устройства



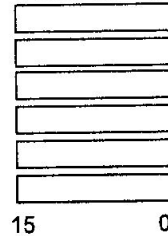
Регистры устройства
с плавающей точкой
(сопроцессора)



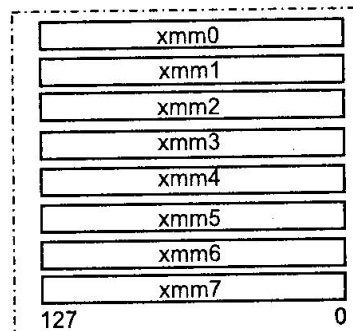
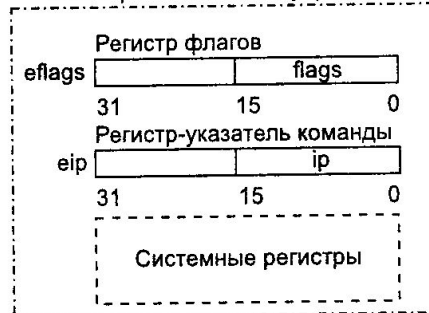
Регистры целочисленного MMX-расширения
(Pentium MMX/II/III/IV)



Сегментные регистры



Регистры состояния и управления



Регистры MMX-расширения с плавающей точкой
(Pentium III/IV)

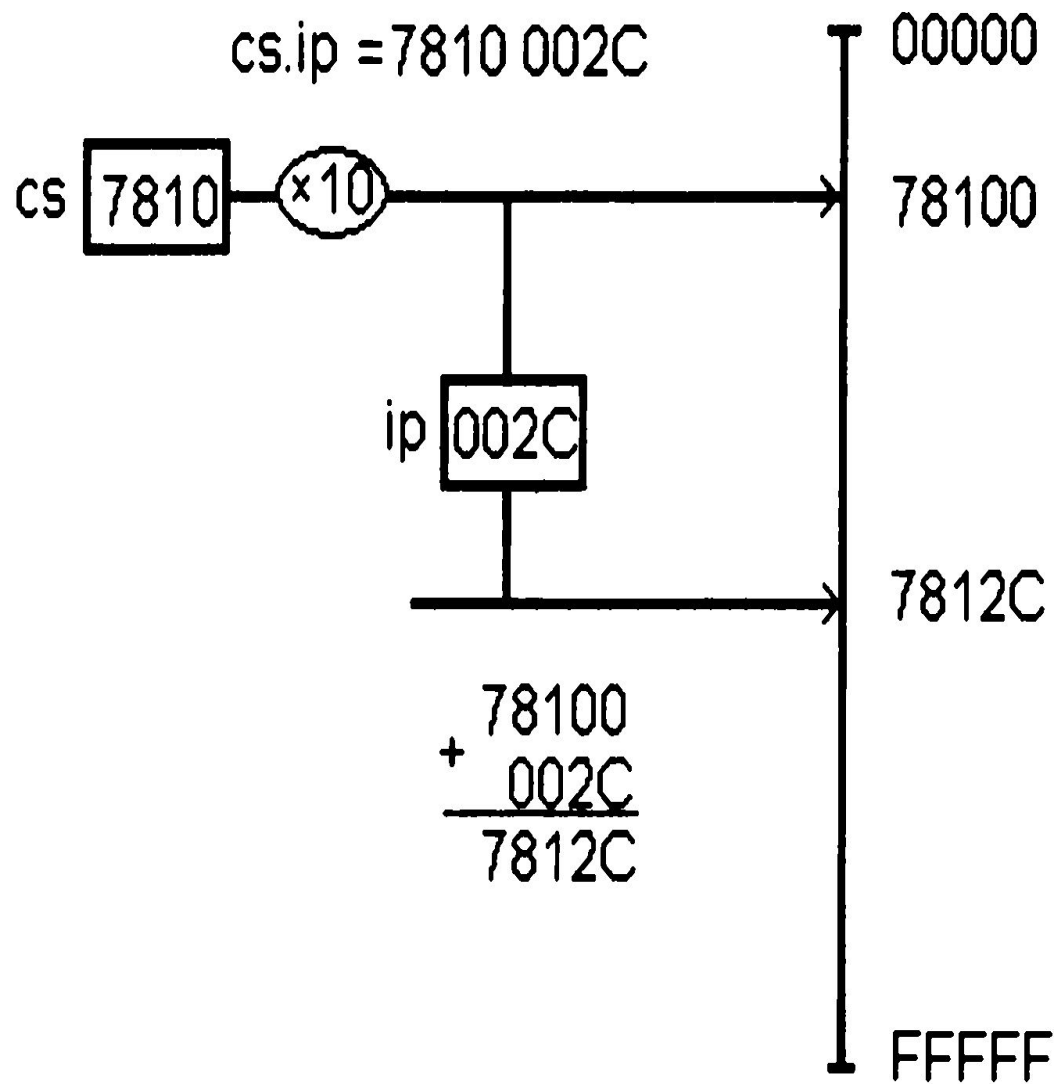


Рис. 4.10. Схема сегментной адресации процессора i8086

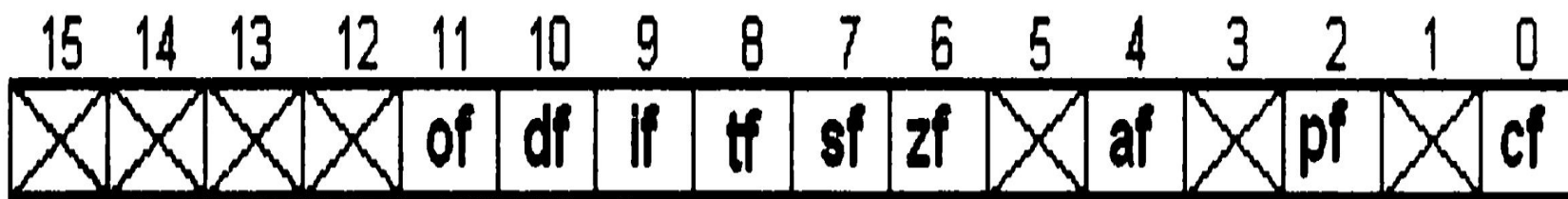


Рис. 4.11. Регистр флажков

Таблица 4.1. Флажки процессора i8086

Название	Разряд	Значение
cf (Carry Flag)	0	Флажок переноса из старшего бита результата. Равен 1, если был такой перенос. Является признаком выхода из диапазона представлений для беззнаковых чисел
pf (Parity Flag)	2	Флажок контроля четности. Равен 1, если младший байт результата содержит четное количество единиц
af (Auxiliary Flag)	4	Вспомогательный флажок переноса. Равен 1, если требуется коррекция сложения или вычитания BCD-данных
zf (Zero Flag)	6	Флажок нуля. Равен 1, если получен нулевой результат
sf (Sign Flag)	7	Флажок знака. Равен 1, если получен результат, который можно трактовать как отрицательное число
of (Overflow Flag)	11	Флажок переполнения. Равен 1, если был перенос из старшего или в старший бит результата и при этом состояние бита изменилось. Это признак выхода из диапазона представлений для знаковых чисел

Таблица 4.1 (продолжение)

Название	Разряд	Значение
tf (Trace Flag)	8	Флажок режима трассировки. Равен 1, если процессор находится в режиме трассировки
if (Interrupt Flag)	9	Флажок прерывания. Равен 1, если аппаратные прерывания разрешены
df (Direction Flag)	10	Флажок направления. Равен 1, если цепочка обрабатывается в направлении возрастания адресов

Таблица 4.2. Установки флажков cf и of

№	Двоичный код	Беззнаковый	cf	Знаковый	of
1	0000 0101 ₂	5 ₁₀	0	+5	0
	<u>0010 0100₂</u>	<u>36₁₀</u>		<u>+36</u>	
	0010 1001 ₂	41 ₁₀		+41	
2	0001 0001 ₂	17 ₁₀	1	+17	0
	<u>1111 1001₂</u>	<u>249₁₀</u>		<u>-7</u>	
	0000 1010 ₂	10 ₁₀		+10	
3	0000 1011 ₂	11 ₁₀	0	+11	1
	<u>0111 1100₂</u>	<u>124₁₀</u>		<u>+124</u>	
	1000 0111 ₂	135 ₁₀		-121	
4	1000 0111 ₂	124 ₁₀	1	-121	1
	<u>1111 0101₂</u>	<u>245₁₀</u>		<u>-11</u>	
	0111 1100 ₂	124 ₁₀		+124	

Таблица 4.3. Сравнение машинного и ассемблерного форматов

Машинный формат		Ассемблерный формат	Действие
Двоичный	Шестнадцатеричный		
0100 0101 ₂	45 ₁₆	inc bp	bp := bp + 1

Таблица 4.4. Фрагмент системы команд процессора i8086

Название	Мнемокод	Код операции	Характеристика
Пересылка	mov	100010	Двухадресная
		1011	Непосредственная регистровая
		1100011; 000	Непосредственная общая
Сложение	add	000000	Двухадресная
Вычитание	sub	001010	Двухадресная
Инкремент	inc	01000	Одноадресная регистровая
		1111111; 000	Одноадресная общая
Декремент	dec	01001	Одноадресная регистровая
		1111111; 001	Одноадресная общая
Возврат в ОС	retf	11001011	Безадресная

Таблица 4.5. Примеры безадресных команд

Машинный формат	Ассемблерный формат	Действие
$0110\ 0000_2$	60_{16} pusha	Записать все регистры в стек
$0110\ 0001_2$	61_{16} popa	Выбрать все из стека в регистры
$1111\ 0100_2$	$F4_{16}$ hlt	Остановить выполнение программы

Таблица 4.6. Кодировка параметра reg

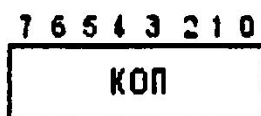
000_2	001_2	010_2	011_2	100_2	101_2	110_2	111_2
ax	cx	dx	bx	sp	bp	si	di

Таблица 4.7. Примеры одноадресных команд в регистровой модификации

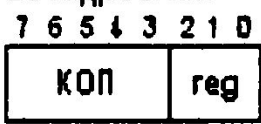
Машинный формат	Ассемблерный формат	Действие
01000 001 ₂ 41 ₁₆	inc cx	cx := cx + 1
01001 001 ₂ 49 ₁₆	dec cx	cx := cx - 1
01000 111 ₂ 47 ₁₆	inc di	di := di + 1
01001 111 ₂ 4F ₁₆	dec di	di := di - 1

Таблица 4.8. Сравнение модификаций одноадресных команд

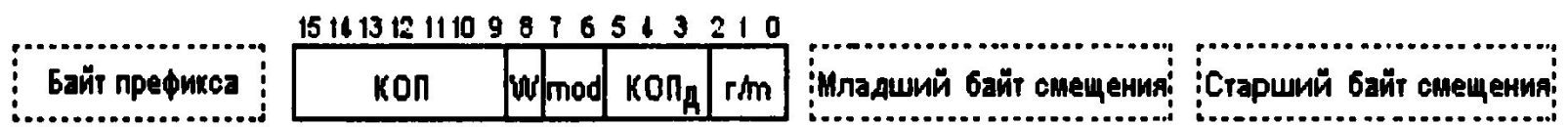
Модификация	Длина операнда	Местоположение операнда
Регистровая	Только 2 байта	Только регистр процессора, <i>reg</i> – код регистра
Общая	1 или 2 байта	<i>mod</i> =112 – регистр процессора, <i>r/m</i> – код регистра <i>mod</i> ≠ 112 – поле памяти, <i>r/m</i> – способ адресации



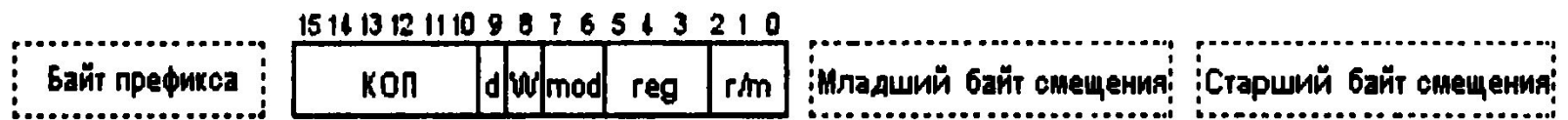
а Безадресная



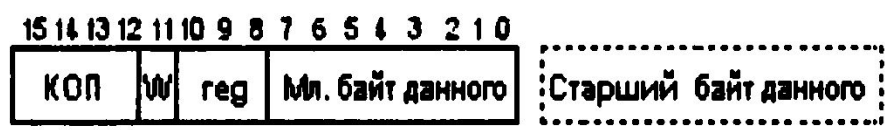
б Одноадресная регистровая модификация



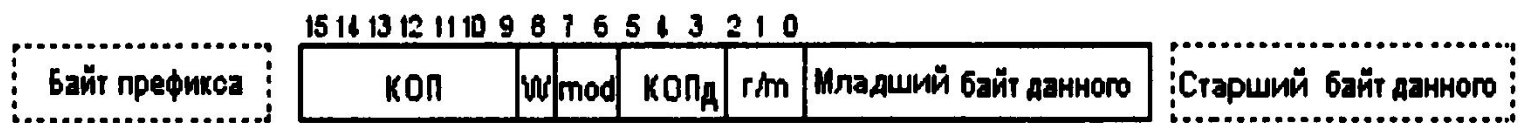
в Одноадресная общая модификация



г Двухадресная



д Непосредственный операнд, регистровая модификация



е Непосредственный операнд, общая модификация

Рис. 4.12. Форматы машинных команд процессора i8086

Случай $\text{mod} = 11_2$

Таблица 4.9. Кодировка параметра r/m при $\text{mod} = 11_2$

r/m	$W = 0$	$W = 1$	r/m	$W = 0$	$W = 1$
000	al	ax	100	ah	sp
001	cl	cx	101	ch	bp
010	dl	dx	110	dh	si
011	bl	bx	111	bh	di

$$A = [S] \cdot 10_{16} + D_s$$

$$D_s = [B] + [I] + D_k$$

Таблица 4.10. Кодировка параметра mod

mod	Длина команды	Пояснение
002	2 или 3 байта	Операнд в поле памяти, дополнительное смещение в команду не включается и слагаемое D_k в выражении (4.2) отсутствует
012	3 или 4 байта	Операнд в поле памяти, смещение в команде D_k занимает один дополнительный байт, обозначенный на рис. 4.12 как «Младший байт смещения»
102	4 или 5 байтов	Операнд в поле памяти, смещение в команде D_k занимает два дополнительных байта, обозначенных на рис. 4.12 как «Младший байт смещения» и «Старший байт смещения»
112	2 байта	Операнд в регистре процессора

Таблица 4.11. Кодировка параметра r/m при mod $\neq 11_2$

r/m	B	I	S
000 ₂	bx	si	ds
001 ₂	bx	di	ds
010 ₂	bp	si	ss
011 ₂	bp	di	ss
100 ₂	—	si	ds
101 ₂	—	di	ds
110 ₂	bp	—	ss
111 ₂	bx	—	ds



Рис. 4.13. Пример формирования команды

Таблица 4.12. Примеры формирования команд при $\text{mod} = 00_2$

r/m	W	16-ричный формат	Участвующие регистры	A	Ассемблерный формат
000_2	0	FE00	[ds], [bx], [si]	10030_{16}	inc byte ptr [bx+si]
000_2	1	FF00	[ds], [bx], [si]	10030_{16}	inc word ptr [bx+si]
001_2	0	FE01	[ds], [bx], [di]	10210_{16}	inc byte ptr [bx+di]
010_2	0	FE02	[ss], [bp], [si]	20120_{16}	inc byte ptr [bp+si]
011_2	0	FE03	[ss], [bp], [di]	20300_{16}	inc byte ptr [bp+di]
100_2	0	FE04	[ds], [si]	10020_{16}	inc byte ptr [si]
101_2	0	FE05	[ds], [di]	10200_{16}	inc byte ptr [di]
110_2^*	0	FE060201	[ds]	10102_{16}	inc byte ptr [0102]
111_2	0	FE07	[ds], [bx]	10010_{16}	inc byte ptr [bx]

Таблица 4.13. Сравнение регистровой и косвенной адресации

Команда	[bx]		Байт памяти по адресу 10010_{16}	
	До	После	До	После
inc bx	0010_{16}	0011_{16}	05_{16}	05_{16}
inc [bx]	0010_{16}	0010_{16}	05_{16}	06_{16}

Таблица 4.14. Примеры формирования команд при $\text{mod} = 01_2$

r/m	W	16-ричный формат	Участвующие регистры	A	Ассемблерный формат
000_2	1	$FE4002_{16}$	[ds], [bx], [si]	10032_{16}	inc byte ptr [bx+si+02]
000_2	2	$FF4002_{16}$	[ds], [bx], [si]	10032_{16}	inc word ptr [bx+si+02]
001_2	1	$FE4102_{16}$	[ds], [bx], [di]	10212_{16}	inc byte ptr [bx+di+02]
010_2	1	$FE4202_{16}$	[ss], [bp], [si]	20122_{16}	inc byte ptr [bp+si+02]
011_2	1	$FE4302_{16}$	[ss], [bp], [di]	20302_{16}	inc byte ptr [bp+di+02]
100_2	1	$FE4402_{16}$	[ds], [si]	10022_{16}	inc byte ptr [si+02]
101_2	1	$FE4502_{16}$	[ds], [di]	10202_{16}	inc byte ptr [di+02]
110_2	1	$FE4602_{16}$	[ss], [bp]	20102_{16}	inc byte ptr [bp+02]
111_2	1	$FE4702_{16}$	[ds], [bx]	10012_{16}	inc byte ptr [bx+02]

Таблица 4.15. Примеры формирования команд при mod = 10₂

r/m	W	16-ричный формат	Участвующие регистры	A	Ассемблерный формат
000 ₂	1	FE800201 ₁₆	[ds], [bx], [si]	10132 ₂	inc byte ptr [bx+si+0102]
000 ₂	2	FF800201 ₁₆	[ds], [bx], [si]	10132 ₂	inc word ptr [bx+si+0102]
001 ₂	1	FE810201 ₁₆	[ds], [bx], [di]	10312 ₂	inc byte ptr [bx+di+0102]
010 ₂	1	FE820201 ₁₆	[ss], [bp], [si]	20222 ₂	inc byte ptr [bp+si+0102]
011 ₂	1	FE830201 ₁₆	[ss], [bp], [di]	20402 ₂	inc byte ptr [bp+di+0102]
100 ₂	1	FE840201 ₁₆	[ds], [si]	10122 ₂	inc byte ptr [si+0102]
101 ₂	1	FE850201 ₁₆	[ds], [di]	10302 ₂	inc byte ptr [di+0102]
110 ₂	1	FE860201 ₁₆	[ss], [bp]	10202 ₂	inc byte ptr [bp+0102]
111 ₂	1	FE870201 ₁₆	[ds], [bx]	10132 ₂	inc byte ptr [bx+0102]

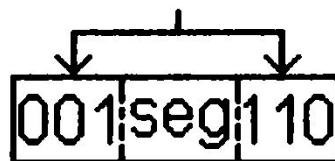


Рис. 4.14. Структура байта префикса замены сегмента

Таблица 4.16. Кодировка байта префикса

Используемый сегмент	Параметр seg	Байт префикса целиком
es	00_2	$001 \mid 00 \mid 110_2 = 26_{16}$
cs	01_2	$001 \mid 01 \mid 110_2 = 2E_{16}$
ss	10_2	$001 \mid 10 \mid 110_2 = 36_{16}$
ds	11_2	$001 \mid 11 \mid 110_2 = 3E_{16}$

Таблица 4.17. Сравнение форматов команд

Машинный формат	Ассемблерный формат	Действие	al		ch	
			До	После	До	После
00E8 ₁₆	add al, ch	al := al + ch	05 ₁₆	0C ₁₆	07 ₁₆	07 ₁₆
02E8 ₁₆	add ch, al	ch := ch + al	05 ₁₆	05 ₁₆	07 ₁₆	0C ₁₆

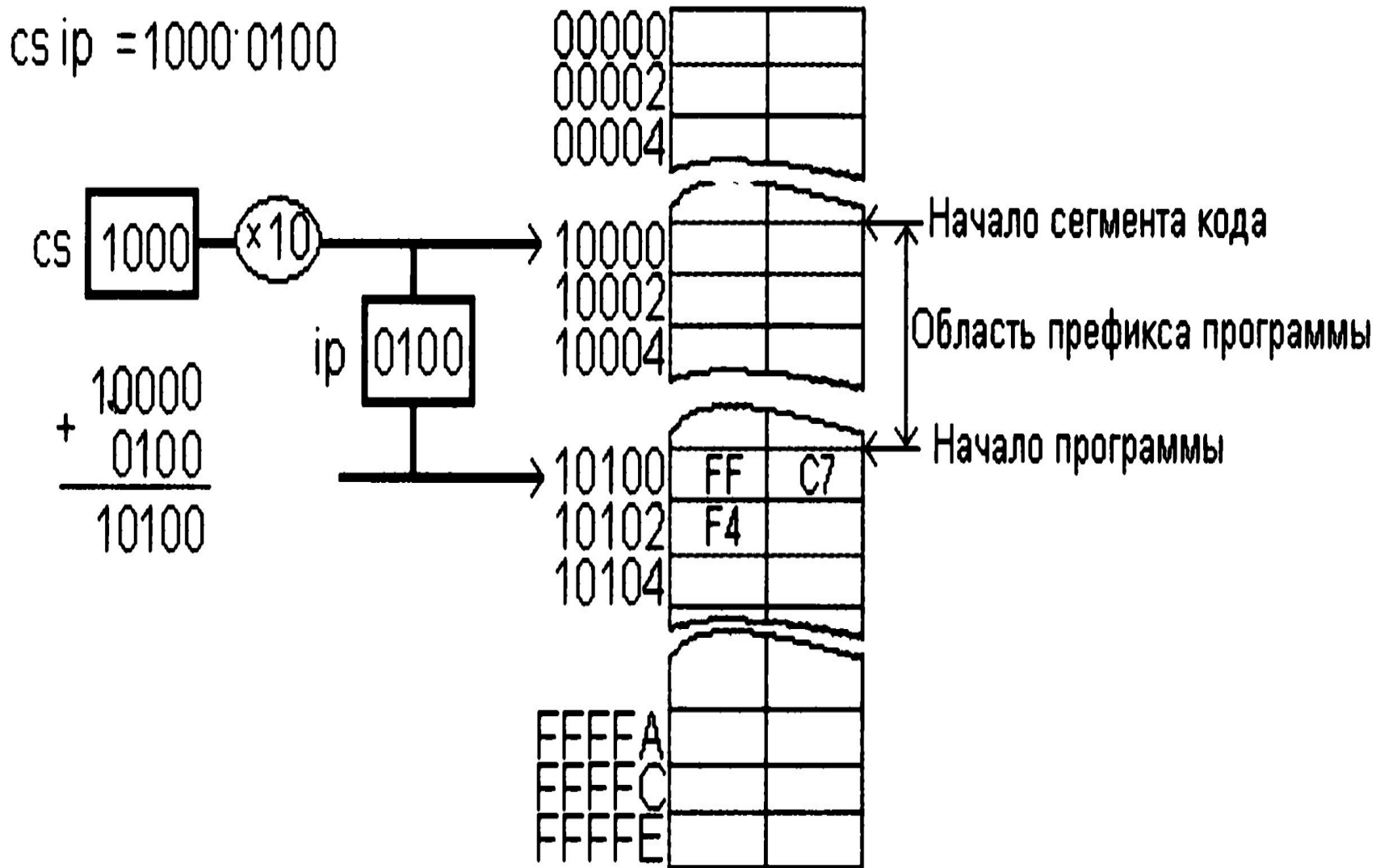


Рис. 4.15. Адресация начала машинной программы

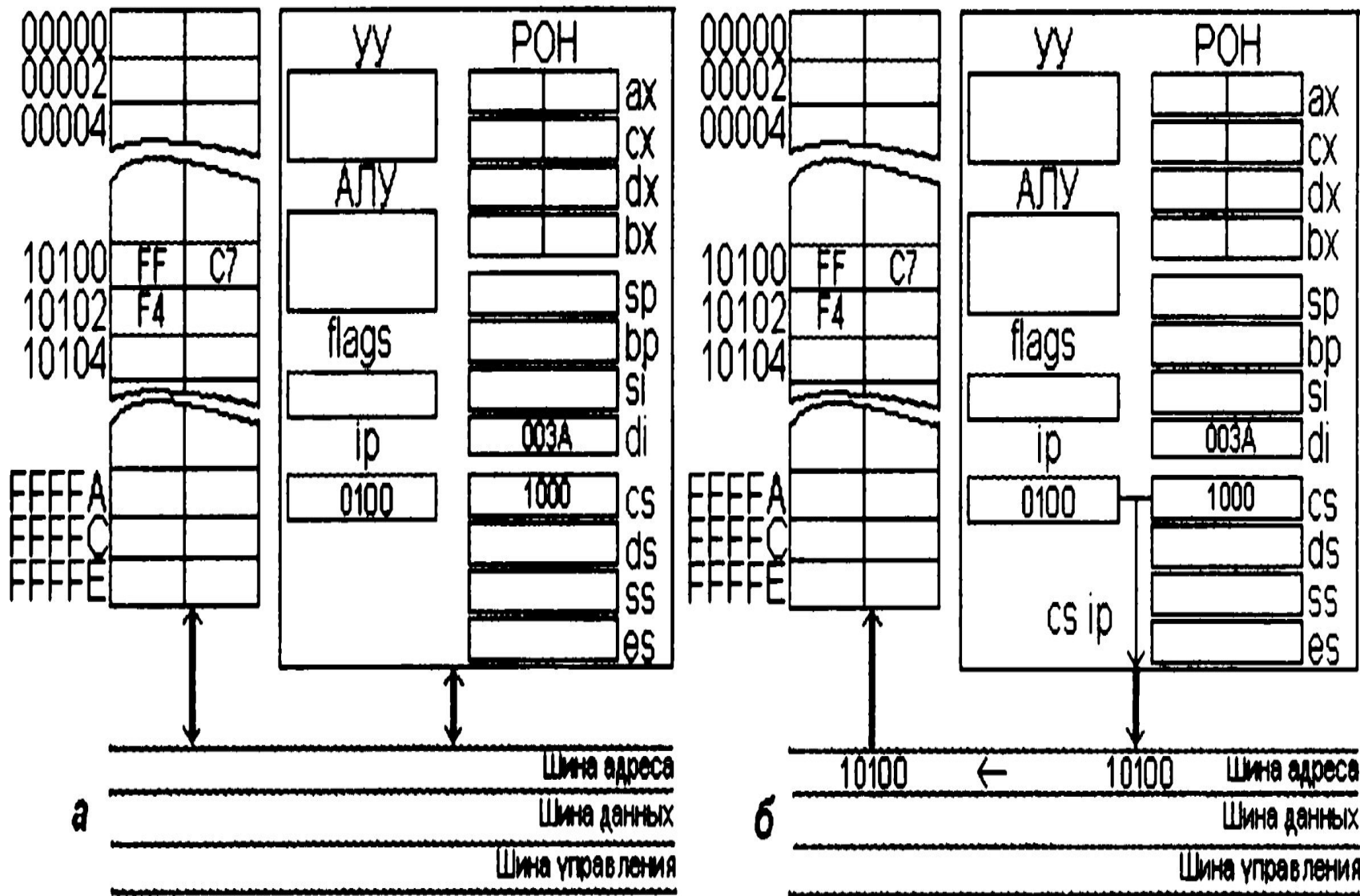


Рис. 4.16. Выполнение машинной команды: а — исходное состояние; б — первый этап

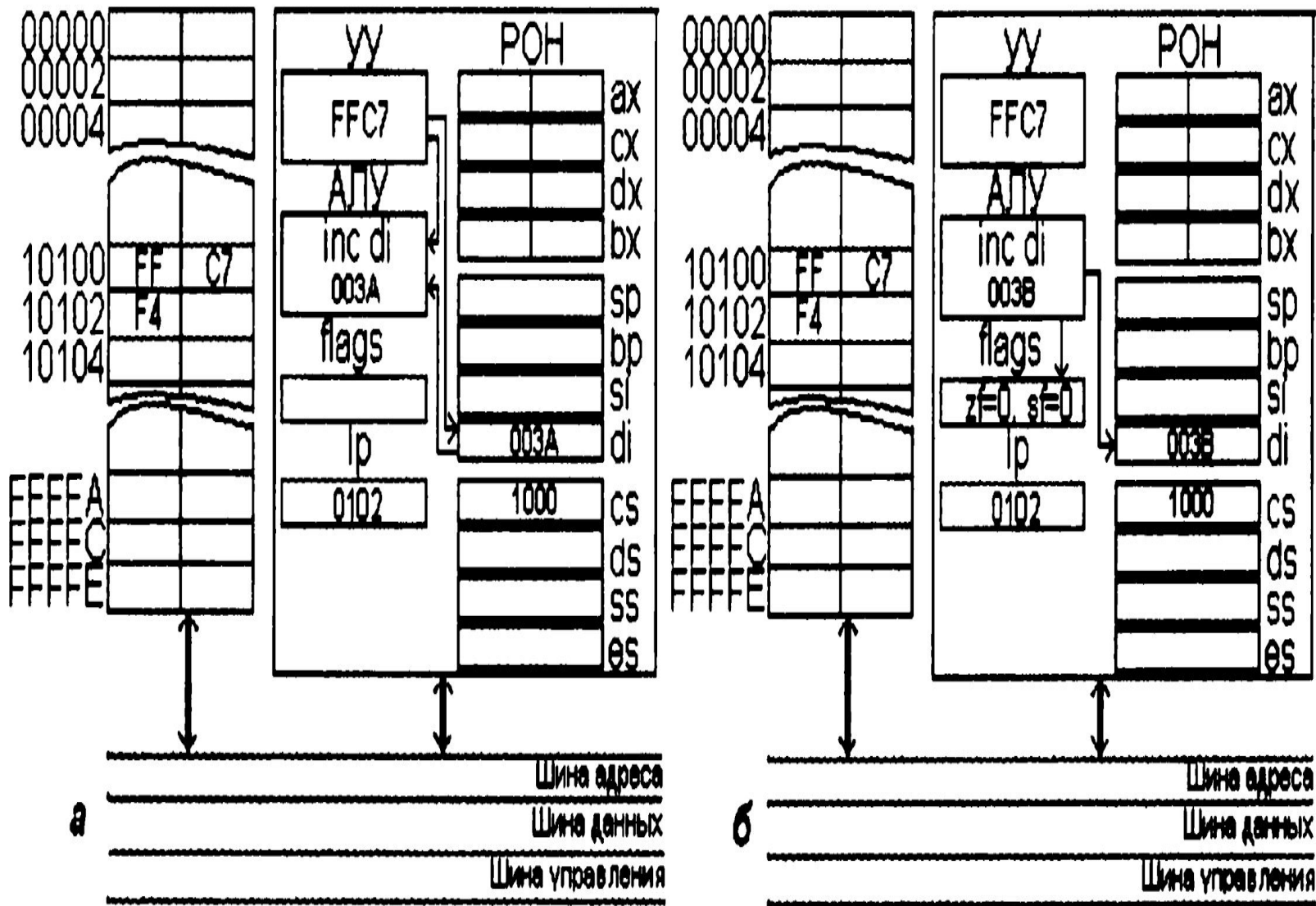


Рис. 4.17. Выполнение машинной команды: а — четвертый этап; б — пятый и шестой этапы



Рис. 4.18. Этапы выполнения машинной команды