

Содержание

От автора	16
Глава 1. Основы компьютерной техники	18
1.1. Из чего состоит компьютер?	20
1.2. Биты, байты, слова, параграфы... ..	21
1.3. Ячейки памяти, порты и регистры	23
1.4. Подсистемы памяти и хранения данных	24
1.4.1. Диски и файловые системы	28
1.5. Устройства ввода-вывода и коммуникаций	29
1.6. Адаптеры, контроллеры, системные и периферийные устройства	30
1.7. Программное обеспечение	30
Глава 2. Устройство персонального компьютера	34
2.1. Настольные компьютеры	35
2.2. Малогабаритные компьютеры	40
2.3. Промышленные компьютеры	42
2.4. Периферийные устройства	45
2.5. Карты, сокет, слоты, джамперы... ..	52
2.6. Кабели и разъемы	55
Глава 3. Архитектура IBM PC-совместимого компьютера	58
3.1. Структурная схема	59
3.2. Распределение памяти	60
3.2.1. Стандартная память — Conventional Memory	64
3.2.2. Верхняя память — UMA	65
3.2.3. Отображаемая и расширенная память — спецификации EMS и XMS	66

3.2.4.	Теневая память — Shadow ROM и Shadow RAM	68
3.2.5.	Архитектура унифицированной памяти — UMA	69
3.2.6.	Виртуальная память	69
3.2.7.	Оптимизация использования оперативной памяти	70
3.3.	Пространство ввода-вывода	75
3.4.	Аппаратные прерывания	78
3.4.1.	Немаскируемые прерывания NMI и SMI	78
3.4.2.	Маскируемые прерывания	79
3.4.3.	Разделяемое использование прерываний	83
3.5.	Прямой доступ к памяти — DMA	84
3.6.	Аксессуары системной платы IBM PC	87
3.6.1.	Системный порт PC/AT	88
3.6.2.	Системный таймер (8253/8254)	89
3.6.3.	Канал управления звуком (PC Speaker)	90
3.6.4.	Батарейная память и часы — CMOS Memory, RTC	91
3.6.5.	Системная поддержка таймеров — BIOS (Int 1Ah и Int 15h)	94
3.7.	Взаимодействие программ с периферийными устройствами	96
3.7.1.	Методы передачи данных	96
3.7.2.	Синхронизация программ и устройств	98
3.7.3.	Буферизация данных в устройствах	99
3.8.	Распределение системных ресурсов	102
3.9.	Функционирование компьютера	104
3.9.1.	Начальный запуск и самотестирование	104
3.9.2.	Загрузка ОС и прикладных программ	106
3.9.3.	«Засыпание» и «пробуждение» компьютера	110
Глава 4.	Системная плата	114
4.1.	Чипсет	115
4.2.	Компоненты: установка и конфигурирование	121
4.2.1.	Процессоры	122
4.2.2.	Оперативная память (DRAM)	138
4.2.3.	Вторичный кэш (SRAM)	141
4.2.4.	Слоты расширения	142
4.2.5.	Синхронизация и разгон	147
4.2.6.	BIOS	150
4.2.7.	Память CMOS: питание и обнуление	155
4.3.	Конструктивы и установка плат	157
4.4.	Подключение системной платы	161
4.5.	«Оживление» системной платы	168
4.6.	Конфигурирование компьютера — CMOS Setup	172
4.7.	Выбор системной платы PC	192

Глава 5.	Процессоры	194
5.1.	Исполнение программного кода	194
5.1.1.	Переключение задач и виртуальные машины	196
5.1.2.	Защищенный режим и виртуальная память	197
5.1.3.	Архитектура и микроархитектура процессоров	199
5.2.	Программная модель 32-разрядных процессоров	201
5.2.1.	Регистры процессора	204
5.2.2.	Организация памяти	206
5.2.3.	Прерывания и исключения	208
5.2.4.	Переключение между реальным и защищенным режимами	210
5.3.	Кэширование памяти	211
5.3.1.	Управление кэшированием	213
5.4.	Особые режимы работы процессора	215
5.4.1.	Запуск и инициализация процессоров	216
5.4.2.	Обновление микрокода	216
5.4.3.	Режим системного управления SMM	217
5.4.4.	Синхронизация и управление энергопотреблением	219
5.4.5.	Мультипроцессорные и избыточные системы	221
5.5.	Совместимость и идентификация процессоров	223
5.5.1.	Совместимость процессоров	223
5.5.2.	Идентификация процессоров	225
5.5.3.	Основные характеристики процессоров	227
5.6.	Процессоры фирмы Intel	230
5.6.1.	Процессоры P6	230
5.6.2.	Процессор Pentium 4	241
5.7.	Процессоры AMD и других фирм	244
5.7.1.	Процессоры фирмы AMD для сокетов 5 и 7	244
5.7.2.	Процессоры Athlon и Duron фирмы AMD	246
5.7.3.	Процессоры фирм Cyrix, VIA, IBM и другие	247
Глава 6.	Электронная память	250
6.1.	Построение оперативной памяти	252
6.1.1.	Быстродействие и производительность памяти	252
6.1.2.	Достоверность хранения данных	254
6.1.3.	Кэширование оперативной памяти	256
6.1.4.	Пакетный режим передачи данных	260
6.2.	Динамическая память	261
6.2.1.	Основы работы DRAM	262
6.2.2.	Регенерация	265

6.2.3.	Асинхронная память: FPM, EDO и BEDO DRAM	267
6.2.4.	Синхронная память: SDRAM и DDR SDRAM	271
6.2.5.	Память Rambus DRAM	276
6.2.6.	Память с виртуальными каналами — VC DRAM	282
6.2.7.	Сравнительная характеристика и перспективные типы динамической памяти	284
6.2.8.	Микросхемы динамической памяти	287
6.2.9.	Модули динамической памяти	292
6.3.	Применение DRAM в оперативной памяти	304
6.3.1.	Нюансы применения DRAM	307
6.3.2.	Рекомендации по выбору модулей динамической памяти	309
6.3.3.	Тестирование оперативной памяти	311
6.4.	Статическая память	313
6.4.1.	Разновидности статической памяти	314
6.4.2.	Применение статической памяти для кэширования ОЗУ	316
6.5.	Энергонезависимая память	322
6.5.1.	Постоянная и полупостоянная память — ROM, PROM, EPROM	324
6.5.2.	Флэш-память и EEPROM	328
6.5.3.	Энергонезависимая память с последовательными интерфейсами	339
Глава 7.	Устройства хранения данных	342
7.1.	Общие свойства устройств внешней памяти	343
7.1.1.	Основные характеристики устройств внешней памяти	343
7.1.2.	Конструктивы и питание устройств	346
7.1.3.	Принципы хранения информации	347
7.1.4.	Хранение информации на магнитных дисках	349
7.2.	Накопители на гибких магнитных дисках	353
7.2.1.	Дискеты: размеры, форматы и правила использования	357
7.2.2.	Интерфейс НГМД	361
7.2.3.	Контроллеры НГМД	364
7.2.4.	Проблемы с накопителями на гибких дисках	367
7.3.	Накопители на жестких магнитных дисках — винчестеры	370
7.3.1.	Конструкция НЖМД	371
7.3.2.	Производительность и оптимизация дисков	378
7.3.3.	Параметры винчестеров	382
7.3.4.	Функционирование винчестера	386

10 Содержание

7.4.	Устройства массовой памяти на сменных носителях	392
7.4.1.	Магнитные диски	392
7.4.2.	Магнитооптические диски	395
7.4.3.	Оптические диски — CD, DVD, PD	400
7.5.	Логическая структура дисков	425
7.5.1.	Разделы и логические диски	425
7.5.2.	Логический диск с файловой системой FAT	429
7.6.	Системная поддержка дисковой памяти	431
7.6.1.	Сервисы BIOS	433
7.6.2.	Системная поддержка CD-ROM	443
7.7.	Установка, обслуживание и тестирование дисков	446
7.7.1.	Интерфейс подключения устройств	446
7.7.2.	Установка новых устройств	448
7.7.3.	Проблемы использования больших дисков	452
7.7.4.	Конфигурирование, форматирование и обслуживание дисков	459
7.7.5.	Основные причины отказов дисков	462
7.7.6.	Повышение надежности и производительности дисков (RAID-массивы)	464
Глава 8.	Видеосистема	466
8.1.	Принципы вывода изображений	468
8.1.1.	Графический режим	471
8.1.2.	Текстовый режим	478
8.1.3.	Трехмерная графика	480
8.1.4.	Обработка видеоизображений	487
8.2.	Дисплей	498
8.2.1.	Параметры монитора	500
8.2.2.	Настройка монитора	502
8.2.3.	Синхронизация и цифровое управление	504
8.2.4.	Управление энергопотреблением	505
8.2.5.	Эргономические характеристики	506
8.2.6.	Плоские дисплеи	507
8.2.7.	Интерфейсы дисплеев	510
8.3.	Дисплейные адаптеры	517
8.3.1.	Компоненты дисплейного адаптера	518
8.3.2.	Динамическая память для дисплейных адаптеров	526
8.3.3.	Интерфейсы дисплейного адаптера	528
8.3.4.	Типы графических адаптеров: MDA, HGC, CGA, EGA, PGA, MCGA, VGA, IBM 8514, SVGA, XGA	529
8.3.5.	Программные модели стандартных адаптеров	531
8.3.6.	Адаптеры с интерфейсами PCI и AGP	532

8.4.	Видео BIOS	536
8.4.1.	Видеосервис BIOS	538
8.5.	Мультидисплейные системы	539
8.6.	Параметры видеосистемы	541

Глава 9. Устройства ввода-вывода

	и их интерфейсы	546
9.1.	Клавиатура	546
9.1.1.	Интерфейс клавиатуры	548
9.1.2.	Контроллер интерфейса клавиатуры и мыши 8042/8242	550
9.1.3.	Скан-коды	552
9.1.4.	Системная поддержка и программный интерфейс	553
9.2.	Манипуляторы-указатели — мышь, трекбол	557
9.2.1.	Последовательные мыши — MS Mouse и PC Mouse	559
9.2.2.	Мышь PS/2	560
9.2.3.	Мышь Bus Mouse	561
9.3.	Принтеры и плоттеры	562
9.3.1.	Буквопечатающие принтеры	563
9.3.2.	Матричные игольчатые принтеры	564
9.3.3.	Термопринтеры	569
9.3.4.	Струйные принтеры	570
9.3.5.	Лазерные принтеры	570
9.3.6.	Плоттеры	572
9.3.7.	Форматы данных	574
9.3.8.	Интерфейсы принтеров и плоттеров	577
9.3.9.	Системная поддержка принтера	583
9.4.	Параллельный порт — LPT	584
9.4.1.	Традиционный LPT-порт	585
9.4.2.	Расширения параллельного порта	589
9.4.3.	Стандарт IEEE 1284	589
9.4.4.	Системная поддержка LPT-порта	597
9.4.5.	Параллельный порт и PnP	597
9.4.6.	Применение LPT-порта	598
9.4.7.	Конфигурирование LPT-портов	599
9.4.8.	Неисправности и тестирование параллельных портов	601
9.5.	Игровые устройства — джойстик, руль, педали	602
9.6.	Коммутаторы устройств ввода-вывода	605

Глава 10. Коммуникационные устройства	608
10.1. Интерфейс RS-232C — COM-порт	608
10.1.1. Протокол RS-232C	610
10.1.2. Управление потоком данных	613
10.1.3. Микросхемы асинхронных приемопередатчиков	615
10.1.4. Системная поддержка COM-портов	616
10.1.5. Конфигурирование COM-портов	618
10.1.6. Использование COM-портов	620
10.1.7. COM-порт и PnP	623
10.1.8. Неисправности и тестирование COM-портов	623
10.2. Беспроводные интерфейсы периферийных устройств	627
10.2.1. Инфракрасный интерфейс IrDA	627
10.2.2. Радиointерфейс Bluetooth	632
10.3. Модемы и факс-модемы	638
10.3.1. Конструкции модемов	640
10.3.2. Технологии xDSL и кабельные модемы	642
10.3.3. Модемы для выделенных линий	644
10.4. Подключение к локальным сетям	645
10.4.1. Сетевые адаптеры	646
10.5. Подключение компьютера к Интернету	654
10.5.1. IP-телефония и передача факсов по IP-сетям	656
Глава 11. Аудиосистема PC	660
11.1. Краткий экскурс в прикладную звукотехнику	663
11.1.1. Основы цифровой обработки сигналов	665
11.1.2. Методы компрессии звуковой информации	668
11.1.3. Методы синтеза звуков	670
11.1.4. Стерефоническое и объемное воспроизведение	675
11.1.5. Трехмерный звук	676
11.2. Звуковая карта PC	678
11.2.1. Аналоговые звуковые карты	680
11.2.2. Цифровые технологии в звуковых картах	681
11.2.3. Аудиокодек AC'97	684
11.2.4. Интерфейсы звуковых карт	688
11.2.5. «Исторические» модели звуковых карт	693
11.3. Интерфейс MIDI	698
Глава 12. Шины расширения	704
12.1. Шина ISA	706
12.2. Шина PC/104	710

12.3.	Шина EISA	711
12.4.	Конфигурирование интерфейсных карт ISA и EISA	712
12.4.1.	Спецификация Plug and Play для шины ISA	714
12.5.	Шина PCI	715
12.5.1.	Шины, логические устройства и функции PCI	716
12.5.2.	Протокол шины PCI	717
12.5.3.	Таймеры, задержки и буферы	720
12.5.4.	Команды шины и адресация	721
12.5.5.	Прерывания	721
12.5.6.	Эмуляция DMA (PC/PCI)	722
12.5.7.	Электрический интерфейс, слоты и карты PCI	723
12.5.8.	Мосты PCI	727
12.5.9.	Программный доступ к конфигурационному пространству и генерация специальных циклов	730
12.5.10.	Конфигурирование устройств	731
12.5.11.	Классы устройств PCI	731
12.5.12.	PCI BIOS	732
12.6.	Магистральный интерфейс AGP	733
12.7.	Интерфейс LPC	740
12.8.	Устаревшие шины PC	741
12.8.1.	Шина MCA	741
12.8.2.	Локальная шина VLB	742
12.9.	Шины и карты расширения блокнотных ПК	742
12.9.1.	Интерфейсы PCMCIA, PC Card и CardBus	743
12.9.2.	CompactFlash	746
12.9.3.	SmartMedia Card	747

Глава 13. Параллельные шины периферийных

	устройств — ATA и SCSI	748
13.1.	Интерфейс ATA (IDE)	749
13.1.1.	Электрический интерфейс	752
13.1.2.	Регистры устройств	758
13.1.3.	Адресация данных	761
13.1.4.	Система команд	763
13.1.5.	Протоколы и режимы передачи данных	768
13.1.6.	Средства многозадачности (ATA-4)	771
13.1.7.	Пакетный интерфейс ATAPI	772
13.1.8.	Адаптеры и контроллеры шины ATA	773
13.1.9.	Категории устройств IDE	779
13.1.10.	Конфигурирование устройств	779

14 Содержание

13.2. Шина SCSI	781
13.2.1. Параллельный интерфейс SCSI	784
13.2.2. Типы периферийных устройств	795
13.2.3. Адресация и система команд	796
13.2.4. Выполнение команд	796
13.2.5. Конфигурирование устройств SCSI	798

Глава 14. Последовательные шины:

USB, FireWire, FCAL, ACCESS.Bus	808
14.1. Шина USB	809
14.1.1. Организация шины USB	810
14.1.2. Модель передачи данных	815
14.1.3. Протокол	816
14.1.4. Типы передач данных	817
14.1.5. Синхронизация при изохронной передаче	818
14.1.6. Устройства и хабы	819
14.1.7. Хост	822
14.1.8. Применение шины USB	824
14.2. Шина IEEE 1394 — FireWire	827
14.2.1. Физический уровень сети	828
14.2.2. Протокол IEEE 1394	830
14.2.3. Устройства и адаптеры 1394	833
14.2.4. Использование шины 1394	834
14.3. Интерфейс Fibre Channel (FCAL)	835
14.4. Последовательная шина ACCESS.Bus и интерфейс I ² C	836

Глава 15. BIOS — базовая система ввода-вывода

15.1. Системная BIOS — System ROM BIOS	839
15.2. Тест начального включения — POST	840
15.3. Начальная загрузка — bootstrap	845
15.4. Сервисы и другие векторы прерываний BIOS	848
15.5. 32-разрядные вызовы — BIOS32	851
15.6. Области данных ROM BIOS — BDA	851
15.7. Расширения ROM BIOS	851
15.8. PnP BIOS	853
15.9. DMI BIOS	853
15.10. Интерфейс ACPI	855
15.11. Флэш-BIOS	859

Глава 16. Питание, охлаждение и электробезопасность

16.1. Схемотехника блоков питания	862
16.2. Блок питания PC	865

16.3. Питание процессоров	872
16.4. Охлаждение компонентов системного блока	873
16.5. Общие вопросы электропитания и заземления	876
16.6. Средства улучшения качества электропитания	879
 Список сокращений	 884
 Литература	 896
 Алфавитный указатель	 897

Архитектура IBM PC-совместимого компьютера

Архитектурный облик IBM PC-совместимого компьютера определяется рядом свойств, обеспечивающих возможность функционирования программного обеспечения, управляющего подключенным оборудованием. Программы могут взаимодействовать с устройствами разными способами:

- используя вызовы функций операционной системы (прерывания DOS, API Windows и т. п.);
- используя вызовы функций базовой системы ввода-вывода (BIOS);
- непосредственно взаимодействуя с известным им «железом» – портами и памятью устройств или контроллеров интерфейсов.

Такой «толстый пирог» из слоев совместимости существует благодаря изначальной открытости архитектуры первых IBM PC и сохранения имеющихся решений (пускай иногда и не самых лучших) в последующих моделях, обрастающих новыми узлами.

Облик PC-совместимого компьютера в значительной степени определяется разработчиками из фирм Microsoft и Intel. Для этих фирм стало уже традицией выпускать объемистый документ, диктующий разработчикам аппаратуры требования для получения вождя логотипа «Designed for Microsoft Windows».

В настоящее время действует спецификация *PC 99 System Design Guide*, опубликованная Microsoft Press в 1998 году. Провозглашенные в ней требования вступили в действие с 1 июля 1999 года и относятся к компьютерам, предназначенным для работы под ОС Windows 98 или Windows 2000 (в спецификации последняя называлась NT Workstation 5.0). Эти требования не распространяются на специализированные серверы, а также на крохотные компьютеры PalmTop с ОС Windows CE. Следующей стала более категоричная спецификация PC'2001, но пока еще не все компьютеры соответствуют и предыдущей.

В спецификациях определяются требования к функциональности и производительности всех подсистем компьютера, включая и периферийные устройства. Отдельные положения этих спецификаций упоминаются в разделах, посвященных конкретным подсистемам ПК.

3.1. Структурная схема

Структурная схема современного IBM PC-совместимого компьютера приведена на рис. 3.1. Ядром компьютера является процессор (CPU), один или несколько; ОЗУ (RAM); ПЗУ с BIOS (ROM BIOS) и интерфейсные средства, связывающие их между собой и с остальными компонентами. Эти средства на рисунке изображены в виде «облака», поскольку их формы разнообразны (шины, хабы). Это «облако» обычно имеет интерфейсы одной или нескольких шин расширения (PCI, ISA...), а также порта AGP. Стандартная архитектура PC определяет набор обязательных средств ввода-вывода и поддержки периферии – системы аппаратных прерываний (PIC 8259A), системы прямого доступа к памяти (DMA 8237A), трехканальный счетчик (8254), интерфейс клавиатуры и управления (KBC 8042), канал управления звуком, память и часы (CMOS RTC). На рисунке изображены лишь логические связи между этими устройствами; подразумевается, что с помощью средств того же «облака» они представлены своими стандартизованными регистрами в общедоступном пространстве ввода-вывода. Также подразумевается, что все компоненты получают требуемое питание, что превращает весь этот набор компонентов в работоспособный компьютер. Конечно же, он должен быть дополнен периферией: дисплеем со своим адаптером, подключаемым к порту AGP, шине расширения или прямо в «облако», контроллерами шин периферийных устройств (ATA, SCSI, USB...), интерфейсов портов (COM, LPT, GAME...), дисководов, аудиосредств и прочего. «Облако» вместе со средствами ввода-вывода и поддержки периферии реализуется чипсетом системной платы (см. п. 4.1), который обычно включает в себя и вышеперечисленные интерфейсы.

Любой PC-совместимый компьютер имеет следующие характерные черты:

- процессор, программно совместимый с семейством x86 фирмы Intel;
- специфическая система распределения пространства адресов памяти;
- традиционное распределение адресов пространства ввода-вывода с фиксированным положением обязательных портов и совместимостью их программной модели;

- систему аппаратных прерываний, позволяющую периферийным устройствам сигнализировать процессору о необходимости исполнения некоторых обслуживающих процедур;
- систему прямого доступа к памяти, позволяющую периферийным устройствам обмениваться массивами данных с оперативной памятью, не отвлекая на это процессор;
- набор системных устройств и интерфейсов ввода-вывода;
- унифицированные по конструктиву и интерфейсу шины расширения (ISA, EISA, MCA, VLB, PCI, PC Card, Card Bus), состав которых может варьироваться в зависимости от назначения и модели компьютера;
- базовую систему ввода-вывода (BIOS), выполняющую начальное тестирование и загрузку операционной системы, а также имеющую набор функций, обслуживающих системные устройства ввода-вывода.

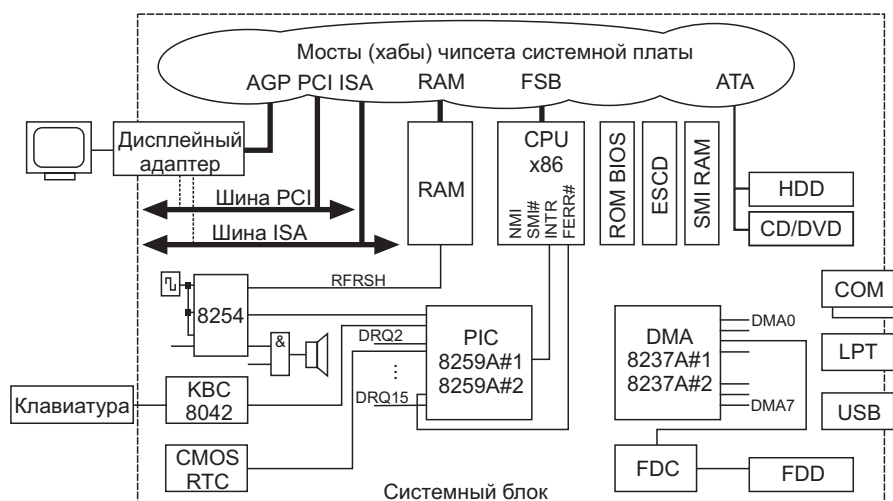


Рис. 3.1. Структурная схема компьютера

3.2. Распределение памяти

Логическая структура памяти PC обусловлена особенностями системы адресации процессоров семейства x86. Процессоры 8086/88, применявшиеся в первых моделях PC, имели доступное адресное пространство 1 Мбайт (20 бит шины адреса). Эти процессоры использовали сегментную модель памяти, унаследованную и следующими моделями в реальном режиме. Согласно этой модели исполнительный (линейный) адрес вычисляется по формуле $Addr = Seg \times 16 + Offset$, где *Seg* и *Offset* – содержимое сегментного и адресного регистров. Таким образом, обеспечивался доступ к адресному пространству $Addr = 00000 - FFFFFh$ при помощи пары 16-битных регистров. Заметим, что при $Seg = FFFFh$ и

$\text{Offset} = \text{FFFFh}$ данная формула дает адрес 10FFEFh , но ввиду 20-битного ограничения на шину адреса эта комбинация в физической памяти указывает на 0FFEFh . Таким образом, адресное пространство как бы сворачивается в кольцо с небольшим «нахлестом». Начиная с процессора 80286, шина адреса была расширена до 24 бит, а впоследствии (386DX, 486 и выше) до 32 и даже 36 (у процессоров P6). В реальном режиме процессора, используемом в DOS, применяется та же сегментная модель памяти и формально доступен лишь 1 Мбайт памяти, что является недостаточным для большинства современных приложений. Однако выяснилось, что процессоры 80286 в реальном режиме эмулируют 8086 с ошибкой: та самая единица в бите A20, которая отбрасывалась в процессорах 8086/88, теперь попадает на шину адреса, и в результате максимально доступный линейный адрес в реальном режиме достиг 10FFEFh . За эту ошибку с радостью ухватились разработчики PC, поскольку дополнительные (64 К - 16) байты оперативной памяти, адресуемой в реальном режиме, оказались подарком, позволяющим освободить дефицитное пространство оперативной памяти для прикладных программ. В эту область ($100000\text{h} - 10\text{FFEFh}$), названную «высокой памятью» – High Memory Area (HMA), стали помещать часть операционной системы и небольшие резидентные программы. Однако для обеспечения полной совместимости с процессором 8086/88 в схему PC ввели вентиль линии A20 шины адреса – *GateA20*, который либо пропускает сигнал от процессора, либо принудительно обнуляет линию A20 системной шины адреса. Более старшие биты такой «заботы» не требуют, поскольку переполнение при суммировании 16-битных компонентов адреса по данной схеме до них не распространяется. Управление этим вентиляем подключили к свободному программно-управляемому выходному биту 1 контроллера клавиатуры 8042, ставшего стандартным элементом архитектуры PC, начиная с AT. Предполагалось, что этим вентиляем часто пользоваться не придется. Однако жизнь внесла свои поправки, и оказалось, что переключение вентиля в многозадачных ОС, часто переключающих процессор между защищенным режимом, реальным режимом и режимом V86, контроллером клавиатуры выполняется слишком медленно. Так появились альтернативные методы быстрого переключения вентиля, специфичные для различных реализаций системных плат (например, через порт 92h). Кроме того, иногда использовали и аппаратную логику быстрого декодирования команды на переключение бита, поступающую к контроллеру клавиатуры. Для определения способа переключения в утилиту CMOS Setup ввели соответствующие параметры (см. п. 4.6), позволяющие выбрать между стандартным, но медленным способом и менее стандартизованным, но быстрым, в зависимости от используемого ПО.

Поскольку ошибка эмуляции 8086 была радостно принята и широко использована, ее повторили и в 386, и в следующих моделях процессоров. А для упрощения внешних схем в процессоры, начиная с 486, ввели и вентиль *GateA20* с соответствующим внешним управляющим выводом.

32-разрядные процессоры позволяют организовать режим, иногда называемый «нереальным» или «большим реальным» (см. п. 5.2), в котором инструкции выполняются как в реальном, но доступны все 4 Гбайт памяти. Этот режим часто используется в игровых программах, целиком захватывающих все ресурсы ком-

62 Глава 3. Архитектура IBM PC-совместимого компьютера

пьютера, не заботясь о «правилах хорошего тона» по отношению к другим исполняемым программам.

Основную часть адресного пространства занимает оперативная память. Объем установленной памяти определяется тестом POST при начальном включении (перезагрузке) компьютера, начиная с младших адресов. Натолкнувшись на отсутствие памяти (ошибку), тест останавливается на достигнутом и сообщает системе объем реально работающей памяти.

Распределение памяти PC, непосредственно адресуемой процессором, приведено на рис. 3.2 и представляется следующим образом.

- 00000h–9FFFFh – *Conventional (Base) Memory*, 640 Кбайт – стандартная (базовая) память, доступная DOS и программам реального режима. В некоторых системах с видеоадаптером MDA верхняя граница сдвигается к AFFFFh (704 Кбайт). Иногда верхние 128 Кбайт стандартной памяти (область 80000h–9FFFFh) называют Extended Conventional Memory.
- A0000h–FFFFFFh – *Upper Memory Area (UMA)*, 384 Кбайт – *верхняя* память, зарезервированная для системных нужд. В ней размещаются области буферной памяти адаптеров (например, видеопамять) и постоянная память (BIOS с расширениями). Эта область, обычно используемая не в полном объеме, ставит непреодолимый архитектурный барьер на пути непрерывной (нефрагментированной) памяти, о которой мечтают программисты.
- Память выше 100000h – *Extended Memory* – *дополнительная* (расширенная) память, непосредственно доступная только в защищенном (и в «большом реальном») режиме для компьютеров с процессорами 286 и выше. В ней выделяется область 100000h–10FFEFh – *высокая* память, НМА, – единственная область расширенной памяти, доступная 286+ в реальном режиме при открытом вентиле *Gate A20*.

Область памяти выше первого мегабайта в различных источниках называется по-разному. Ее современное английское название – Extended Memory – пересекается с названием одной из спецификаций ее использования – Extended Memory Specification. Но название другой спецификации использования – Expanded Memory Specification – в прямом переводе на русский язык неотличимо от перевода предыдущего термина (и Extended и Expanded переводятся как «расширенный»). Будем придерживаться терминологии, укрепившейся в литературе, выпущенной издательством «Питер», и область всей физической памяти, расположенной в адресном пространстве выше 1 Мбайта, будем называть *дополнительной памятью*. Ее объем у современных компьютеров указывается строкой **Extended Memory xxxxx Kbyte** в таблице, выводимой после прохождения теста POST, и в меню стандартной конфигурации CMOS Setup.

Вышеприведенное разделение памяти актуально только для приложений и операционных систем реального режима типа MS-DOS. Для ОС защищенного режима (в том числе Windows 9x/NT/2000) доступна вся оперативная память, причем без каких-либо ухищрений вроде EMS и XMS, описанных ниже. Однако область UMA с ее традиционными «жителями», сохраняемая ради совместимости, остается барьером на пути к единой однородной памяти.

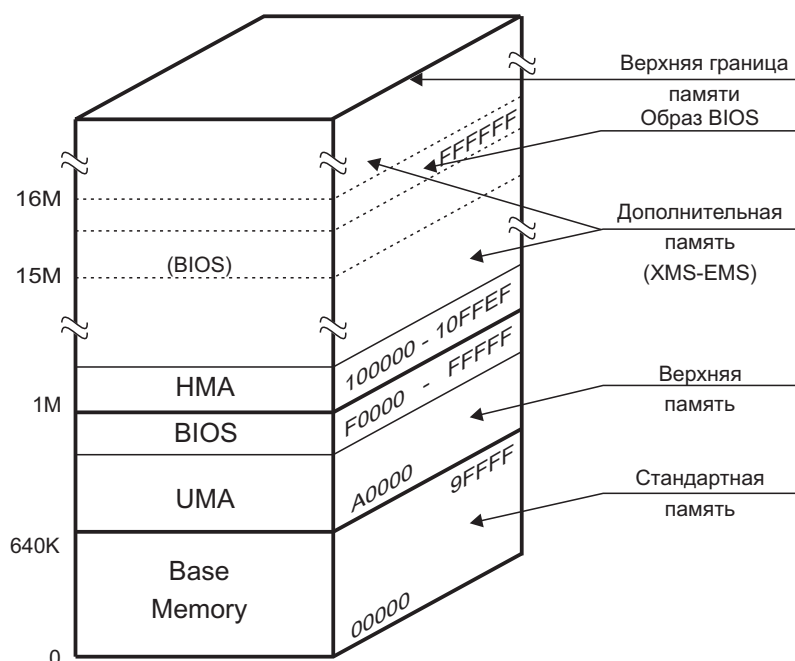


Рис. 3.2. Распределение памяти PC

Для компьютеров класса AT-286 с 24-битной шиной адреса верхняя граница оперативной памяти – $FDFFFFh$ (максимальный размер 15,9 Мбайт). Область $FE0000h-FFFFFFFh$ содержит ПЗУ BIOS (ROM BIOS Area), обращение к этой области эквивалентно обращению к ROM BIOS по адресам $0E0000h-0FFFFFFh$.

Для 386+ процессоров и 32-битной шины адреса теоретическая верхняя граница – 4 Гбайт, а для P6 – 64 Гбайт (36-битная шина адреса). В компьютерах с 32-разрядной шиной адреса образ BIOS дополнительно проецируется в адреса $FFE0000h-FFFFFFFFh$, хотя для процессоров P6 это и необязательно. Однако иногда используется и проекция BIOS в область $FE0000h-FFFFFFFh$, что не позволяет задействовать более 16 Мбайт ОЗУ, поскольку система воспринимает только найденную непрерывную область оперативной памяти. Если 32-разрядный компьютер имеет отображение области BIOS под границей 16 Мбайт, это отображение обычно можно запретить установкой соответствующего параметра CMOS Setup. Иногда для использования специфических адаптеров ISA, имеющих буфер с адресами в 16-м мегабайте памяти, предусматривают параметр **Memory Hole At 15-16M**. Его установка также не позволяет использовать оперативную память свыше 16 Мбайт.

Реально современные системные платы позволяют установить до 512-2048 Мбайт ОЗУ, для мощных серверных платформ и это не предел. Обращение по адресам, превышающим границу установленной оперативной памяти (или максимально возможного объема), транслируется на шину PCI, которая имеет 32-битную адресацию.

Подробнее с иерархической подсистемой электронной памяти можно познакомиться в главе 6. Здесь же отметим, что в логическом распределении памяти фигурирует только оперативная и постоянная память, а кэш является программно-прозрачным средством повышения ее производительности.

Компьютеры, использующие режим системного управления *SMM* (System Management Mode), имеющийся у большинства процессоров последних поколений, имеют еще одно адресное пространство памяти – *SMRAM*. Это адресное пространство «параллельно» пространству обычной памяти и при работе доступно процессору только в режиме обработки SMI. Память *SMRAM* может представлять собой часть физической оперативной памяти (DRAM), а может быть реализована и специальной микросхемой энергонезависимой памяти. Ее размер может варьироваться в диапазоне от 32 Кбайт (минимальные потребности SMM) до 4 Гбайт. *SMRAM* располагается, начиная с адреса *SMIBASE* (по умолчанию 30000h), и распределяется относительно адреса *SMIBASE* следующим образом.

- FE00h–FFFFh (3FE00h–3FFFFh) – область сохранения контекста (распределяется, начиная со старших адресов по направлению к младшим). По прерыванию SMI сохраняются почти все регистры процессора, но сохранение регистров FPU не производится.
- 8000h (38000h) – точка входа в обработчик (SMI Handler).
- 0–7FFFh (30000h–37FFFh) – свободная область.

Память *SMRAM* должна быть схемотехнически защищена от доступа прикладных программ. Процессор генерирует специальный выходной сигнал *SMIACK#* во время обработки SMI, который и должен являться «ключом» доступа к этой памяти. Если *SMRAM* не является энергонезависимой, то системная логика должна обеспечить возможность ее инициализации (записи программного кода обработчика) процессором из обычного режима работы до разрешения появления сигнала *SMI#*.

3.2.1. Стандартная память — Conventional Memory

Стандартная память является самой дефицитной в PC, когда речь идет о работе в среде операционных систем типа MS-DOS. На ее небольшой объем (типичное значение 640 Кбайт) претендуют и BIOS, и ОС реального режима, а остатки отдаются прикладному ПО. Стандартная память распределяется следующим образом:

- 00000h–003FFh – *Interrupt Vectors* – векторы прерываний (256 двойных слов);
- 00400h–004FFh – *BIOS Data Area* – область переменных BIOS;
- 00500h–00xxxh – *DOS Area* – область DOS;
- 00xxxh–9FFFFh – *User RAM* – память, предоставляемая пользователю (до 638 Кбайт); при использовании PS/2 Mouse область 9FC00h–9FFFFh используется как расширение BIOS Data Area, и размер User RAM уменьшается.

3.2.2. Верхняя память — UMA

Верхняя память имеет области различного назначения, которые могут быть заполнены буферной памятью адаптеров, постоянной памятью или оставаться незаполненными. Раньше эти «дыры» не использовали из-за сложности «фигурного выпиливания» адресуемого пространства. С появлением механизма страничной переадресации (у процессоров 386 и выше) их стали по возможности заполнять «островками» оперативной памяти, названными блоками верхней памяти *UMB* (Upper Memory Block). Эти области доступны DOS для размещения резидентных программ и драйверов через драйвер *EMM386*, который отображает в них доступную дополнительную память.

Стандартное распределение верхней памяти выглядит следующим образом (рис. 3.3).

- A0000h–BFFFFh – *Video RAM*, 128 Кбайт – видеопамять (обычно используется не полностью).
- C0000h–DFFFFh – *Adapter ROM, Adapter RAM*, 128 Кбайт – резерв для адаптеров, использующих собственные модули ROM BIOS или (и) специальное ОЗУ, разделяемое с системной шиной.
- E0000h–EFFFFh – свободная область, 64 Кбайт, иногда занятая под System BIOS.
- F0000h–FFFFFFh – *System BIOS*, 64 Кбайт – системная BIOS.
- FD000h–FDFFFFh – *ESCD* (Extended System Configuration Data) – область энергонезависимой памяти, используемая для конфигурирования устройств Plug and Play. Эта область имеется только при наличии PnP BIOS, ее положение и размер жестко не заданы.

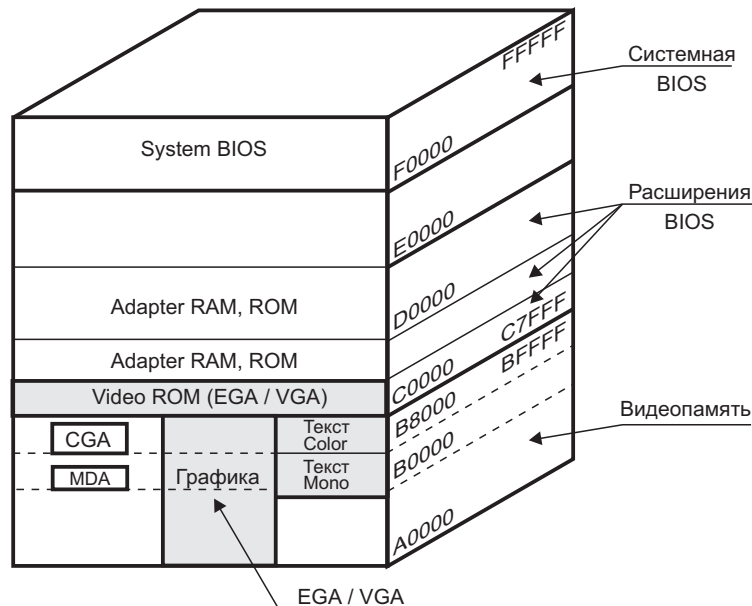


Рис. 3.3. Распределение верхней памяти (UMA)

В области UMA практически всегда присутствует графический адаптер. В зависимости от модели он занимает следующие области:

- MDA RAM – B0000h–B0FFFh;
- CGA RAM – B8000h–BBFFFh;
- EGA ROM – C0000h–C3FFFh/C7FFFh;
- VGA ROM – C0000h–C7FFFh;
- EGA, VGA RAM – A0000h–BFFFFh, в зависимости от видеорежима используются следующие области:
 - Graphics – A0000h–AFFFFh;
 - Color Text – B8000h–BFFFFh;
 - Mono Text – B0000h–B7FFFh.

Также распространенным потребителем UMA являются расширения ROM BIOS, расположенные на платах дисковых контроллеров и микросхемы удаленной загрузки (Boot ROM) на платах адаптеров ЛВС. Обычно они занимают область C8000h – CBFFFh/C9FFFh/C8FFFh (для дисковых контроллеров), но могут и перемещаться при конфигурировании адаптеров.

Размер области, занимаемой системной ROM BIOS, колеблется от 8 Кбайт у PC/XT до 128 Кбайт, однако разумное значение – 64 Кбайт. Большая область использовалась «на радостях» от появления микросхем ROM и флэш-памяти объемом 1 Мбит (128К × 8), но при этом размер доступной UMA сократился. Тогда стали микросхемы того же (и большего) объема отображать только на область F0000h–FFFFFFh (64 Кбайт), а иногда и меньшую. Это оказалось возможным, поскольку не все содержимое микросхемы ROM BIOS должно быть доступно одновременно. Таким способом удалось примирить интересы пользователей UMB с необходимостью расширения объема BIOS, связанной с усложнением технических средств.

3.2.3. Отображаемая и расширенная память — спецификации EMS и XMS

Отображаемая память EMS (Expanded Memory Specification) – программная спецификация использования дополнительной памяти DOS-программами реального режима. Спецификация *LIM EMS* – соглашение фирм Lotus, Intel, Microsoft на использование EMS. С помощью специальных аппаратных или программных средств любая область дополнительной памяти может быть отображена на небольшие страницы, расположенные в области UMA. В первоначальном варианте можно было использовать 4 страницы по 16 Кбайт, примыкающие друг к другу, обычно начиная с адреса D0000h (положение страниц можно менять в пределах свободных областей UMA). Обращение прикладных программ к памяти EMS осуществляется через диспетчер памяти, вызываемый по прерыванию Int 67h. Программа, нуждающаяся в дополнительной памяти, должна сначала запросить выделение области, указав ее размер в 16-килобайтных страницах. В ответ на этот запрос (если имеется свободная память) диспетчер сообщает программе номер дескриптора EMS (EMS handler), по которому программа в дальнейшем бу-

дет ссылаться на выделенную ей область при управлении отображением. Далее программа через диспетчер назначает отображение требуемой логической страницы из выделенной ей области дополнительной памяти на выбранную физическую страницу, расположенную в области UMA. После этого любые программные обращения процессора к физической странице, расположенной в пределах первого мегабайта, будут в действительности работать с логической страницей дополнительной памяти, расположенной выше первого мегабайта, причем без переключения в защищенный режим. Для работы с иной логической страницей требуется вызов диспетчера для переназначения отображения. В EMS 4.0, эмулируемой на процессорах 386+, появилась возможность увеличения числа доступных физических страниц и отображения дополнительной памяти не только на фиксированные области UMA, но и на любые области памяти.

Для поддержки EMS поначалу требовались специальные аппаратные средства. В компьютерах на процессорах 386 и выше появилась возможность программной эмуляции EMS, которую в MS-DOS 5+ выполняет драйвер EMM386.EXE.

Система EMS в основном предназначена для хранения данных – для исполняемого в данный момент программного кода она неудобна, поскольку требует программного переключения страниц через каждые 16 Кбайт. EMS используется в основном старым ПО, фирма Lotus продвигала эту спецификацию для хранения своих больших электронных таблиц. Ее используют для создания виртуальных дисков, хранения очередей заданий для печати, а также и для хранения данных и даже программного кода некоторых резидентных программ (в целях экономии стандартной памяти).

Расширенная память XMS (eXtended Memory Specification) – иная программная спецификация использования дополнительной памяти DOS-программами, разработанная компаниями Lotus, Intel, Microsoft и AST для компьютеров на процессорах 286 и выше. Эта спецификация позволяет программе получить в распоряжение одну или несколько областей дополнительной памяти, а также использовать область НМА. Распределением областей ведает диспетчер расширенной памяти – драйвер HIMEM.SYS. Диспетчер позволяет захватить или освободить область НМА (65 520 байт, начиная с 100000h), а также управлять вентилем линии адреса A20. Функции собственно XMS позволяют программе:

- определить размер максимального доступного блока памяти;
- захватить или освободить блок памяти;
- копировать данные из одного блока в другой, причем участники копирования могут быть блоками как стандартной, так и дополнительной памяти в любых сочетаниях;
- запретить блок памяти (запретить копирование) и отпереть его;
- изменить размер выделенного блока.

В ответ на запрос выделения области диспетчер выдает номер дескриптора блока (16-битное число XMS handler), по которому выполняются дальнейшие манипуляции с этим блоком. Размер блока может достигать 64 Мбайт. Спецификация XMS позволяет программам реального режима устраивать «склады» данных в дополнительной памяти, которая им непосредственно недоступна, копируя в нее и из нее данные доступных областей первого мегабайта памяти. До-

ступ к диспетчеру XMS осуществляется через прерывание Int 2Fh. Заботу о переключении в защищенный режим и обратно для получения доступа к дополнительной памяти берет на себя диспетчер. По умолчанию HIMEM.SYS позволяет использовать до 32 дескрипторов блоков, но это число можно увеличить, задав параметр /NUMHANDLES=xx в строке загрузки драйвера HIMEM.SYS.

Кроме работы с дополнительной памятью спецификация XMS определяет пару функций и для работы с блоками UMB – захватить блок требуемого размера (или определить максимально доступный блок) и освободить его.

Как видно, спецификации EMS и XMS отличаются по принципу действия: в EMS для доступа к дополнительной памяти выполняется отображение (страничная преадресация) памяти, а в XMS – копирование блоков данных. На компьютерах с процессорами 386+ эти спецификации мирно сосуществуют при использовании драйвера HIMEM.SYS, поверх которого может быть загружен и драйвер EMM386.EXE, пользующийся памятью XMS для эмуляции EMS-памяти. Память, доступная EMS и XMS, может выделяться динамически из числа дополнительной. Ключ NOEMS в строке запуска EMM386 запрещает выделение памяти под использование по спецификации EMS.

3.2.4. Теневая память — Shadow ROM и Shadow RAM

В области верхней памяти UMA обычно располагаются устройства с медленной памятью: системная BIOS (System ROM BIOS), расширения BIOS на графическом адаптере (Video ROM BIOS), на контроллерах дисков и интерфейсов (Adapter ROM), ПЗУ начальной загрузки на сетевой карте (Boot ROM), видеопамять (Video Memory Buffer). Они, как правило, реализованы на 8- или 16-битных микросхемах с довольно большим временем доступа. Обращение к полноразрядному системному ОЗУ выполняется гораздо быстрее. Для ускорения обращений к памяти этих устройств применяется *теневая память* (Shadow Memory) – подмена ее системным ОЗУ. Теневая память появилась на развитых моделях AT-286, где она была реализована аппаратно. Процессоры класса 386+ позволяют ее реализовать программно, с помощью страничной преадресации (см. п. 5.2.2). Затенение ОЗУ и ПЗУ устройств выполняется по-разному.

При инициализации теневого ПЗУ (*Shadow ROM*) содержимое затеняемой области копируется в ОЗУ, и при дальнейшем чтении по этим адресам подставляется ОЗУ, а запись в эту область блокируется.

При использовании теневого ОЗУ (*Shadow RAM*) запись производится одновременно в физическую память затеняемой области и в системное ОЗУ, наложенное на эту область. При чтении затененной области обращение идет только к системной памяти, что происходит гораздо быстрее. Особенно велик эффект от затенения видеопамати старых графических адаптеров, которая по чтению бывает доступна только во время обратного хода развертки, и процессору приходится долго ждать этого момента. Однако затенение областей разделяемой памяти, модифицируемых со стороны адаптеров, недопустимо – эти изменения не будут восприняты процессором. К разделяемой относится буферная память сетевых адаптеров, видеопамати адаптеров с графическими сопроцессорами (аксе-

лераторами). Из этого следует, что затенение видеопамати применимо только к примитивным графическим картам, устанавливаемым в слот ISA, и то не во всех режимах.

Обычно тневая память включается через CMOS Setup отдельными областями размером по 16 Кбайт или более крупными, и для каждой области указывают режим затенения (Shadow ROM или Shadow RAM). Возможно ее включение и драйверами ОС (например, драйвером EMM386). На современных системных платах затенение области системной BIOS выполняется всегда, на старых платах затенением этой области можно было управлять. Затенение BIOS видеоадаптера (Video BIOS Shadowing) для работы в среде Windows с «родными» драйверами графического адаптера может и не давать прироста производительности.

3.2.5. Архитектура унифицированной памяти — UMA

Видеопамать графического адаптера является особой областью памяти, к которой во время непрерывного процесса регенерации экрана интенсивно обращается и центральный процессор, и графический акселератор (если таковой имеется). Видеопамать традиционно является физически выделенной памятью сравнительно (по сравнению с ОЗУ) небольшого объема, и для нее разными способами обеспечивают максимальную производительность – увеличивают разрядность до 128 бит, повышают частоту, применяют специализированные, в том числе и двухпортовые, микросхемы памяти. Это, конечно же, приводит к удорожанию компьютера. Для современных графических акселераторов требуется доступ к большому объему памяти, причем с высокой производительностью. Вместо предоставления локальной памяти адаптера была предложена *архитектура унифицированной памяти UMA* (Unified Memory Architecture). Здесь для видеопамати (и других нужд акселератора) выделяется область в общем пространстве единой физической оперативной памяти. За этот способ снижения стоимости приходится расплачиваться снижением производительности как видеосистемы, так и основной памяти. Архитектура UMA применяется в чипсетах системной платы с интегрированной графикой для недорогих компьютеров. При этом может предоставляться возможность установки и дополнительного специализированного модуля видеопамати, позволяя за дополнительные деньги отказаться от UMA. Если с графического адаптера AGP убрать локальную память, этот высокопроизводительный адаптер вырождается в систему с UMA.

3.2.6. Виртуальная память

Виртуальная память (Virtual Memory) представляет собой программно-аппаратное средство расширения пространства памяти, предоставляемой программе в качестве оперативной. Как уже было сказано в главе 1, эта память физически реализуется в оперативной и дисковой памяти под управлением соответствующей операционной системы. Виртуальное пространство памяти разбито на страницы фиксированного размера, а в физической оперативной памяти в каждый

момент времени присутствует только часть из них. Остальные страницы хранятся на диске, откуда операционная система может «подкачать» их в физическую память, предварительно выгрузив на диск часть не используемых в данный момент модифицированных страниц. Обращение процессора к ячейке виртуальной памяти, присутствующей в физической памяти, происходит обычным способом. Если же затребованная область в данный момент не отображена в физической памяти, процессор вырабатывает исключение (внутреннее прерывание), по которому операционная система программно организует замещение страниц, называемое свопингом (Swapping). Виртуальную память поддерживают процессоры, работающие в защищенном режиме, начиная с 80286, но реально ее широко стали использовать только в операционных системах и оболочках для 32-разрядных процессоров (80386+). Виртуальная память используется лишь при наличии дополнительной памяти, а ее максимальный объем определяется размером файла подкачки (Swap File), выделяемом на жестком диске на нужды виртуальной памяти. В принципе файл подкачки может располагаться и на сетевом диске, но при этом трафик сети будет напряженным. Вопросы организации виртуальной памяти в основном относятся к области системного программного обеспечения (ядра операционной системы), здесь же мы ограничимся приведенным описанием.

3.2.7. Оптимизация использования оперативной памяти

Оптимизация использования памяти может преследовать две цели – увеличение объема памяти, доступной приложению, и повышение быстродействия обращений к памяти. На старых машинах, когда объем установленной физической памяти не превышал единиц мегабайт, зачастую приходилось чем-то жертвовать; в современных системах достижение обеих целей уже не противоречит друг другу. Нет сомнений в том, что чем больше установленный объем ОЗУ, тем лучше – компьютер будет мощнее: он позволит загружать приложения, особо «жадные» до памяти; увеличить число одновременно работающих приложений (в многозадачной ОС); ускорить работу. Однако для обеспечения возможности использования памяти в ряде случаев требуются некоторые дополнительные действия по конфигурированию компьютера.

В системах с размером установленной памяти более 640 Кбайт возможны различные варианты использования последних 384 Кбайт из первого мегабайта физической памяти:

- память не используется;
- область (или часть ее) перемещается в конец дополнительной памяти;
- область (или часть ее) используется в качестве теневой (*Shadow*) памяти адаптеров и ROM BIOS.

Перемещение неиспользуемого остатка первого мегабайта в конец дополнительной памяти (разрешается установкой параметра **Memory Relocation** в CMOS Setup) возможно не всегда. Обычно такое перемещение становится невозможным, если хоть часть из этого кусочка используется в качестве теневой памяти.

Также перемещение может предлагаться лишь при небольших объемах установленной памяти, и на современных системных платах эта возможность почти не встречается. Поэтому не стоит удивляться сообщению об объеме памяти, обнаруженном тестом POST, в котором относительно установленного «не хватает» 384 Кбайт.

Иногда BIOS предлагает такое распределение памяти, при котором под стандартную память выделяется 512 Кбайт, а остальная память идет как расширенная. Пользу такого распределения оценить трудно. При этом вышесказанное про верхние 384 Кбайт становится справедливым для оставшихся 512 Кбайт, правда, на возможность их перемещения в конец дополнительной памяти ограничения будут мягче.

Острее всего проблема обеспечения доступности памяти стоит для приложений MS-DOS, исполняемых в среде этой системы. Обычно «битва» идет за килобайты стандартной памяти (conventional memory), доступной приложениям. Из 640 Кбайт после загрузки ОС и необходимых резидентных драйверов на долю приложений может остаться около 500 Кбайт, а то и меньше, что для ряда приложений неприемлемо. Несмотря на повсеместное внедрения ОС типа Windows 9x, интерес к запуску больших приложений MS-DOS (например, бухгалтерских программ, в том числе и принудительно распространяемых налоговыми органами, а также игр) сохраняется и поныне. Сообщение о нехватке памяти на компьютере с ОЗУ размером, например, 32 Мбайт для приложения, скромно просящего всего 590 Кбайт, неприятно удивляет неискушенных пользователей. Однако эта нехватка не фатальна, если правильно выбрать версию и способ загрузки ОС. Все программные настройки, влияющие на объем доступной памяти, кроются в файлах CONFIG.SYS и AUTOEXEC.BAT, размещенных в корневом каталоге загрузочного диска.

Понимая проблемы с использованием памяти различными процессорами, для DOS-приложений можно рекомендовать следующие версии MS-DOS, оптимальные по размеру свободной стандартной памяти:

- для компьютеров класса XT на процессорах 8086/88 – MS-DOS 3.30;
- для компьютеров класса AT на процессорах 80286 – MS-DOS 5.x;
- для компьютеров класса AT на процессорах 80386 и выше – MS-DOS 6.2x (правда, одно время встречались некачественные системные платы для 80386, на которых MS-DOS 6.2x отказывалась загружаться в HMA, хотя MS-DOS 5.x работала нормально).

Кроме MS-DOS существуют и другие операционные системы реального режима, совместимые с MS-DOS (например, PC DOS, DR DOS, COMPAQ DOS). Каждая из них имеет свои преимущества и недостатки по сравнению с соответствующим поколением MS-DOS, но их обсуждение выходит за рамки данной книги. Операционные системы (и оболочки) защищенного режима на компьютеры с процессором класса ниже 386 пытаться устанавливать почти бессмысленно.

Система MS-DOS 3.3 (более ранние рассматривать не будем) загружалась целиком в стандартную память, но была довольно компактной и оставляла приемлемое место для приложений своего времени. Когда вышла MS-DOS версии 4 с более развитыми возможностями, появившимися ценой ее разрастания, мно-

72 Глава 3. Архитектура IBM PC-совместимого компьютера

гие приложения отказывались работать в ее среде именно из-за нехватки памяти, и эта версия ОС широкого распространения не получила. Более удачной стала версия 5, которая «научилась» использовать «высокую память» (НМА) на компьютерах с процессором 286 и выше, если таковая присутствовала. Для этого был введен специальный драйвер высокой памяти, и в файле CONFIG.SYS должны присутствовать строки:

```
DEVICE=[<path>]HIMEM.SYS (загрузка драйвера высокой памяти)
DOS=HIGH (указание на загрузку ОС в высокую память)
```

Конечно, в верхнюю память загружается не вся ОС – часть все-таки попадает и в стандартную память. Кроме того, в стандартную память загружаются и резидентные драйверы – например, русификаторы клавиатуры и экрана, драйвер мыши и т. п. Все они отрывают свой кусок от памяти, которую могли бы использовать приложения. На компьютерах с 32-разрядными процессорами (386 и выше), имеющих механизм страничной переадресации, появилась возможность использования «верхней памяти» (UMA) с помощью драйвера EMM386.EXE. Этот драйвер отыскивает в области UMA (A0000–FFFFFh) регионы, не занятые памятью устройств, и отображает их на области доступной дополнительной памяти. В эти регионы, нормально адресуемые процессором в реальном режиме, можно помещать модули операционной системы и загружаемые драйверы; их же могут использовать и приложения.

Для наиболее компактной загрузки MS-DOS версий 5 и выше (а также Windows 9x, которая может представляться как MS-DOS 7) в файле CONFIG.SYS должны быть следующие директивы:

```
DEVICE=[<path>]HIMEM.SYS (загрузка драйвера высокой памяти)
DEVICE=[<path>]EMM386.EXE (загрузка диспетчера расширенной памяти)
DOS=HIGH, UMB (указание на загрузку ОС в высокую и верхнюю память)
```

Резидентные драйверы (русификаторы клавиатуры, экрана и принтера, драйвер мыши) по умолчанию обычно загружаются в стандартную память. Если приложениям MS-DOS не хватает свободной памяти, то по крайней мере часть резидентных драйверов можно загрузить в верхнюю память UMA. Для драйверов, загружаемых из файла CONFIG.SYS, вместо команды DEVICE= следует использовать команду DEVICEHIGH [/L:n[,m]]=, которая попытается загрузить драйвер в n-регион UMB. Необязательный параметр m задает требуемый объем памяти (он может отличаться от размера файла с драйвером). Ключ загрузки /L вместе с параметрами n, m используется для ручной оптимизации памяти. Если драйверу требуется выделить несколько областей памяти, то они перечисляются в списке вида /L:n1[,m1];n2[,m2][;...]. Для резидентных программ, загружаемых из файла AUTOEXEC.BAT, тем же целям служит команда LOADHIGH (LH) с аналогичными необязательными параметрами, задающими размер одной или нескольких областей UMA. Строка запуска будет иметь вид

```
LH[/L:n[,m]] <путь>\<файл>[<параметры>]
```

Посмотреть текущее положение свободных блоков памяти можно командой MEM /F из командной строки DOS/Windows. Чтобы не заниматься ручной оптимизацией памяти, в состав DOS/Windows введена диалоговая утилита MEMMAKER.EXE, которая за несколько этапов расставит требуемые команды в фай-

лах CONFIG.SYS и AUTOEXEC.BAT. До ее запуска в этих файлах должны присутствовать ссылки на все необходимые драйверы и программы. Файл CONFIG.SYS должен начинаться с вышеприведенных трех строк (если не указать DOS=HIGH, утилита может «постесняться» использовать высокую память для загрузки DOS). Запуск нерезидентных приложений, а также оболочки типа Norton Commander из файла AUTOEXEC.BAT на время работы MEMMAKER следует отменить (MEMMAKER несколько раз перезагружает компьютер и автоматически запускается после обработки файла AUTOEXEC.BAT). При начальном запуске MEMMAKER спрашивает, требуется ли память EMS для приложений, и если нет, то установит ключ NOEMS в строке запуска EMM386.

Для того чтобы в верхнюю память поместилось как можно больше модулей, ее следует оптимизировать. Оптимизация UMA подразумевает такое конфигурирование базовых адресов буферной и постоянной памяти адаптеров, при котором свободные области UMA получились бы максимально возможного размера. Нужно стремиться к плотному соединению областей UMA, занимаемых адаптерами. Тогда максимальный размер блоков UMB увеличится, и драйвер типа EMM386 сможет разместить в нем более крупные модули, что, в свою очередь, позволит освободить дополнительное место в остродефицитной стандартной памяти. Если на конфигурирование адаптеров не обращать внимания, то может оказаться, что два компьютера с одинаковым составом аппаратных средств и программным обеспечением будут после загрузки ОС иметь значительно различающиеся размеры стандартной памяти.

Многие адаптеры (контроллеры SCSI, адаптеры локальных сетей и др.) позволяют задавать адреса областей встроенных RAM и ROM, отображаемых в пространство памяти компьютера. Это позволяет разрешать (или, наоборот, создавать) конфликты использования UMA, а также оптимизировать использование ее блоков. При конфигурировании устанавливаемых адаптеров необходимо исключить перекрытия занимаемых адресов, поскольку из-за этого, скорее всего, ни одно из конфликтующих устройств работать не будет. Самый неприятный (трудно устранимый) конфликт возникает, если при конфигурировании какого-либо адаптера его память перекрывает буфер или BIOS графического адаптера. Если конфигурирование осуществляется джамперами, то вернуть нормальную конфигурацию труда не составит. А если адаптер конфигурируется только с помощью специальной утилиты, изменяющей содержимое его энергонезависимой памяти, то изменить его конфигурацию можно, лишь загрузив и выполнив соответствующую утилиту. Но при конфликте с графическим адаптером это сделать не так-то просто – в лучшем случае удастся загрузить компьютер со «слепым» экраном, а в худшем – POST откажется продолжать тестирование и загрузку, обнаружив ошибку графического адаптера и сообщив об этом попискиванием динамика. Но эта ситуация не так безнадежна: есть еще в природе графический адаптер MDA, у которого видеобуфер не совпадает по адресам с распространенными адаптерами EGA и VGA, а расширение BIOS у него отсутствует. Вставив неудачно сконфигурированную плату в компьютер с адаптером MDA (и, конечно, соответствующим монитором), можно утилитой задать правильную конфигурацию. Однако этот способ для новых компьютеров, не имеющих слотов шины ISA, уже не пригоден.

Кроме занимаемого пространства оптимизация использования UMA касается, как ни странно, и быстродействия. Для областей памяти адаптеров часто бывает полезным применение теневой памяти. Управление теневой памятью осуществляется через CMOS Setup для определенных областей. Поэтому, размещая конфигурируемые области памяти, иногда следует учитывать возможности задания границ теневой памяти в CMOS Setup. Необходимо помнить, что Shadow ROM блокирует запись, а Shadow RAM игнорирует возможность изменения со стороны адаптера затеняемой области памяти, что приведет к ошибкам при некорректном использовании теневой памяти. Для процессоров 386 и выше теневую память может организовать и драйвер EMM386, но этой его функцией пользуются редко.

Работу приложений в среде MS-DOS, в которой активно используется программный код BIOS, значительно ускоряет затенение ROM BIOS, как системной, так и BIOS графического адаптера и дискового контроллера. Для многозадачных ОС защищенного режима (Windows и др.) затенение ROM BIOS ускоряет только начальный процесс загрузки ОС, поскольку в рабочем режиме здесь в основном используются драйверы, загружаемые в ОЗУ.

Приложения ОС Windows пользуются *виртуальной памятью* и никакие старые спецификации EMS и XMS им не нужны. Суммарный объем виртуальной памяти, доступной всем приложениям, определяется размером ОЗУ и файлов подкачки (их может быть и несколько). В Windows 9x размер файла подкачки изменяется динамически, по мере потребностей системы. Для того чтобы приложениям хватало памяти, на диске, несущем файл подкачки, должно быть достаточно свободного пространства (десятки и сотни мегабайт). Конечно же, важен и объем установленной физической памяти – ее малый объем может быть принципиальным ограничением на запуск ряда приложений или установку операционных систем. При малом объеме ОЗУ свопинг (подкачка страниц) будет слишком интенсивным, в результате чего скорость работы приложений существенно снижается (обращения к диску выполняются на несколько порядков медленнее, чем к ОЗУ). Приложения реального времени (например, аудио- и видеопроигрыватели и тем более кодеры) могут стать неработоспособными именно из-за малого объема ОЗУ. Поскольку файл подкачки изменяет свой размер в процессе работы, важно следить за фрагментацией диска, несущего этот файл – обращение к фрагментированному файлу выполняется медленнее, чем к нефрагментированному. При выборе диска для размещения файла подкачки следует учитывать его быстродействие – время доступа и скорость передачи данных. При использовании приложений реального времени, интенсивно обменивающихся с дисками (те же проигрыватели и кодеры, а также программы, записывающие компакт-диски), по возможности файл подкачки следует размещать на других дисках.

ВНИМАНИЕ

Если на компьютере под управлением ОС защищенного режима (Windows, Unix, OS/2...) перестают запускаться приложения с сообщениями о недостаточном объеме оперативной памяти – проверьте наличие свободного места на жестких дисках, используемых для подкачки. Если памяти не хватает приложениям MS-DOS – проверьте файлы AUTOEXEC.BAT и CONFIG.SYS и запустите утилиту MEMMAKER или выполните ручную оптимизацию памяти.

Увеличение физического объема оперативной памяти в ряде случаев может привести к неожиданному снижению производительности компьютера. Это возможно, когда системная плата (или процессор со вторичным кэшем) не способна кэшировать весь объем ОЗУ. У многих системных плат для процессоров Pentium кэшируются только первые 64 Мбайт ОЗУ; у первых процессоров Pentium II кэшировались только 512 Мбайт. Память, выходящая за размеры кэшируемой области, конечно же доступна, но ее производительность гораздо ниже кэшируемой. ОС Windows 9x распределяет память, начиная с верхней границы доступной памяти, причем наверх попадает ее ядро, скорость работы которого существенна для работы многих приложений. Если после увеличения ОЗУ ядро попадает в некэшируемую область, можно наблюдать снижение производительности. Для лечения этого недуга можно воспользоваться условно-бесплатной программой **W2CACHE.COM**, которая запускается в начале загрузки Windows и, оставаясь резидентной, «съедает» верхнюю часть памяти, заставляя ядро Windows загружаться в нижнюю, кэшируемую область. После окончания загрузки Windows программа освобождает занимаемую память, и ОС отдает ее в распоряжение приложений.