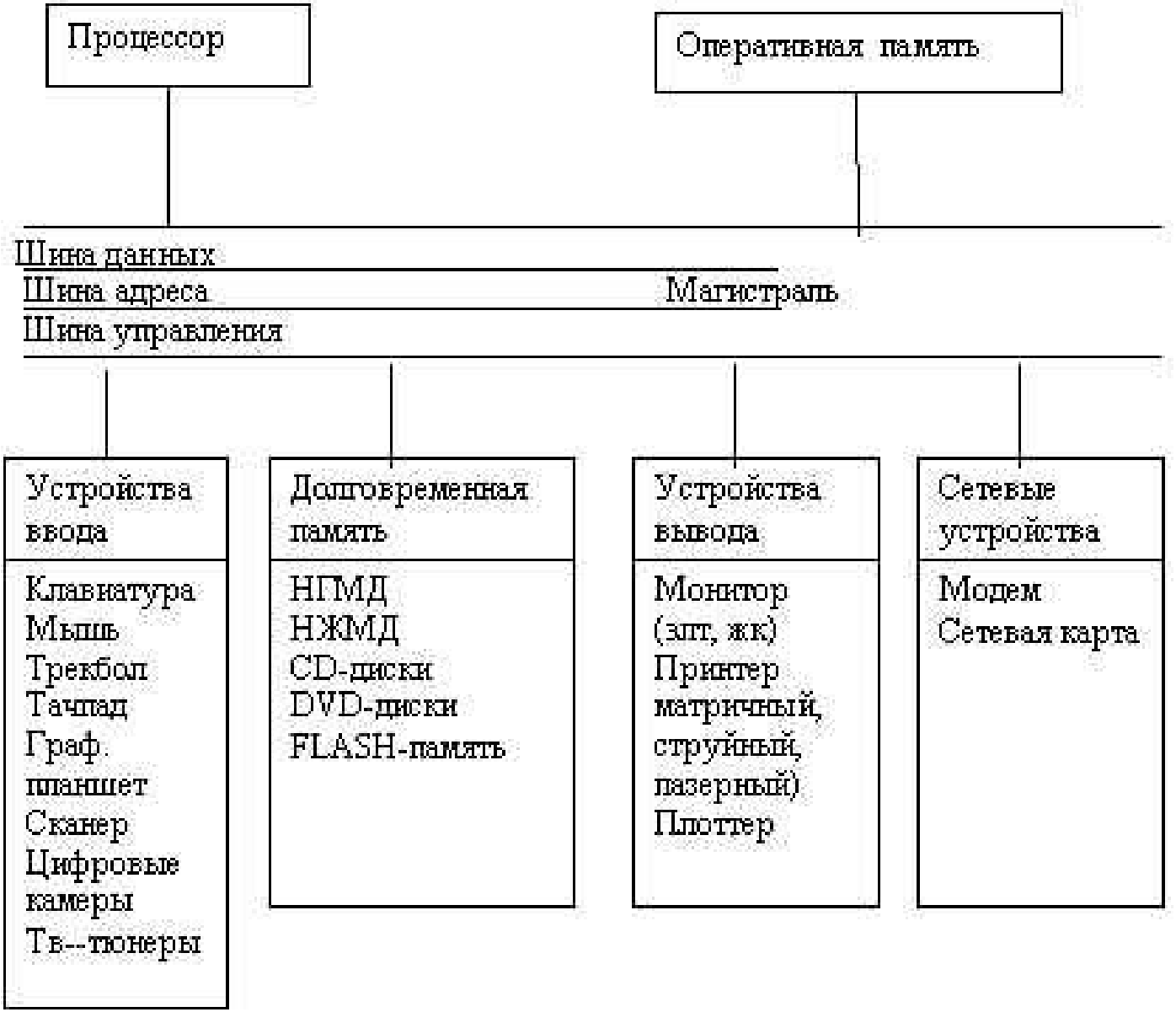
Взаимодействие с внешними устройствами

Структурная схема



Регистры внешних устройств

- Адресуемые «ячейки»
- Чтение — получение информации о состоянии внешних устройств
- Запись — изменение состояния внешнего устройства
- Режимы доступа определяются для каждого бита индивидуально

Порты ввода-вывода x86

- Отдельное адресное пространство: 2^16 доступных портов ввода-вывода
- Первые 256 портов доступны «прямой адресацией», остальные через BX
- IN REG, PORT — чтение из порта. REG: AL, AX, EAX. PORT: IMM8, BX.
- OUT PORT, REG — запись в порт.

Работа с клавиатурой

- Порт 0x60 — данные
- Порт 0x64 — статус
  - Бит 0 — клавиатура не готова принимать данные
  - Бит 1 — данные от клавиатуры готовы
- Отправка байта на клавиатуру:

```
Sendbyte:
    mov ecx, eax
w:  in   al, 0x64
    test al, 1
    jnz  w
    mov eax, ecx
    out  0x60, al
    ret
```

Работа с клавиатурой

- Включение CapsLock, ScrollLock, NumLock LED

```
mov al, 0xed
call sendbyte
mov al, 0x7  ; light all leds
call sendbyte
```

- Чтение скан-кода

```
w: in al, 0x64
   and al, 2
   jz w
   in al, 0x60
```

Проблемы

- Неудобный доступ
- 64K — слишком мало для современных компьютеров
  - Много устройств
  - Много управляющих регистров у каждого устройства
- Сложности с автоматическим конфигурированием и «горячим подключением»

Отображение в память

- Память устройства, конфигурационные параметры, регистры ввода-вывода отображаются на адресное пространство памяти
- PCI на 32-битных PC-совместимых компьютерах:
  - Верхний гигабайт физической памяти использовался для отображения устройств ввода-вывода
  - Память не доступна, даже если установлена

Проблемы (2)

- Постоянный опрос состояния внешнего устройства (i/o polling)
  - Устройство работают МНОГО медленнее процессора — очень много циклов тратится впустую — BUSY WAIT (активное ожидание)
  - Энергопотребление!
  - Процессор не может выполнять другие задачи

Прерывания

- На одном из входов процессора выставляется электрический уровень прерывания
- Процессор заканчивает выполнение текущей инструкции
- Запрещаются прерывания (флаг IF процессора)
- Сохраняется минимальная информация, необходимая для продолжения выполнения с текущей точки
- Выполняется переход на обработчик прерывания

Сохраняемая информация

- В реальном режиме: EFLAGS, CS, EIP
- Сохраняется в стек (SS, ESP)
- В конце обработчика прерывания состояние процессора восстанавливается с помощью инструкции IRET
- Обработчик прерывания начинает работу в режиме запрета прерываний. Как можно быстрее прерывания должны быть разрешены инструкцией STI

Источники прерываний

- Non-maskable interrupt (NMI)
- Обычные от внешних устройств (INTR)
- Исключения — специальные состояния при исполнении программы
  - Page fault
  - Division by zero
  - …
- Программные исключения — инициируются самой выполняющейся программой с помощью инструкции INT n

Номер прерывания

- X86 поддерживает 256 различных типов прерываний (для всех целей)
- 256 адресов начала обработчиков прерываний — таблица обработчиков прерываний
  - В реальном режиме располагается по адресу 0
  - В защищенном — где угодно
- Программное прерывание:
  - INT 3 — debug trap (занимает 1 байт вместо 2)
  - INTO — INT 4 if OF is set (1 байт вместо 2)
  - INT n

Список прерываний

- 0-31 — исключения процессора
  - 0 — divide error
  - 3 — breakpoint
  - 4 — overflow
  - 6 — invalid opcode
  - 13 — general protection fault
  - 14 — page fault
- 32-255 - пользовательские

APIC (advanced programmable interrupt controller)

- Отображает 16 (32, 64) входных линий прерываний от внешних устройств в линию INTR на процессор (или на ядра процессора)
- Отображает входные номера прерываний IRQ0 … IRQ15 на прерывания процессора (например, INT32 .. INT47)
- Маршрутизация прерываний на ядра процессора
- Приоретизация прерываний
- Блокировка прерываний на время обработки

Финализация прерываний

- Когда APIC отправляет сигнал прерывания INTR на ЦП, отправка дальнейших сигналов блокируется
- ЦП должен явно разблокировать отсылку прерываний

```
mov al, 0x20
out   0x20, al
```

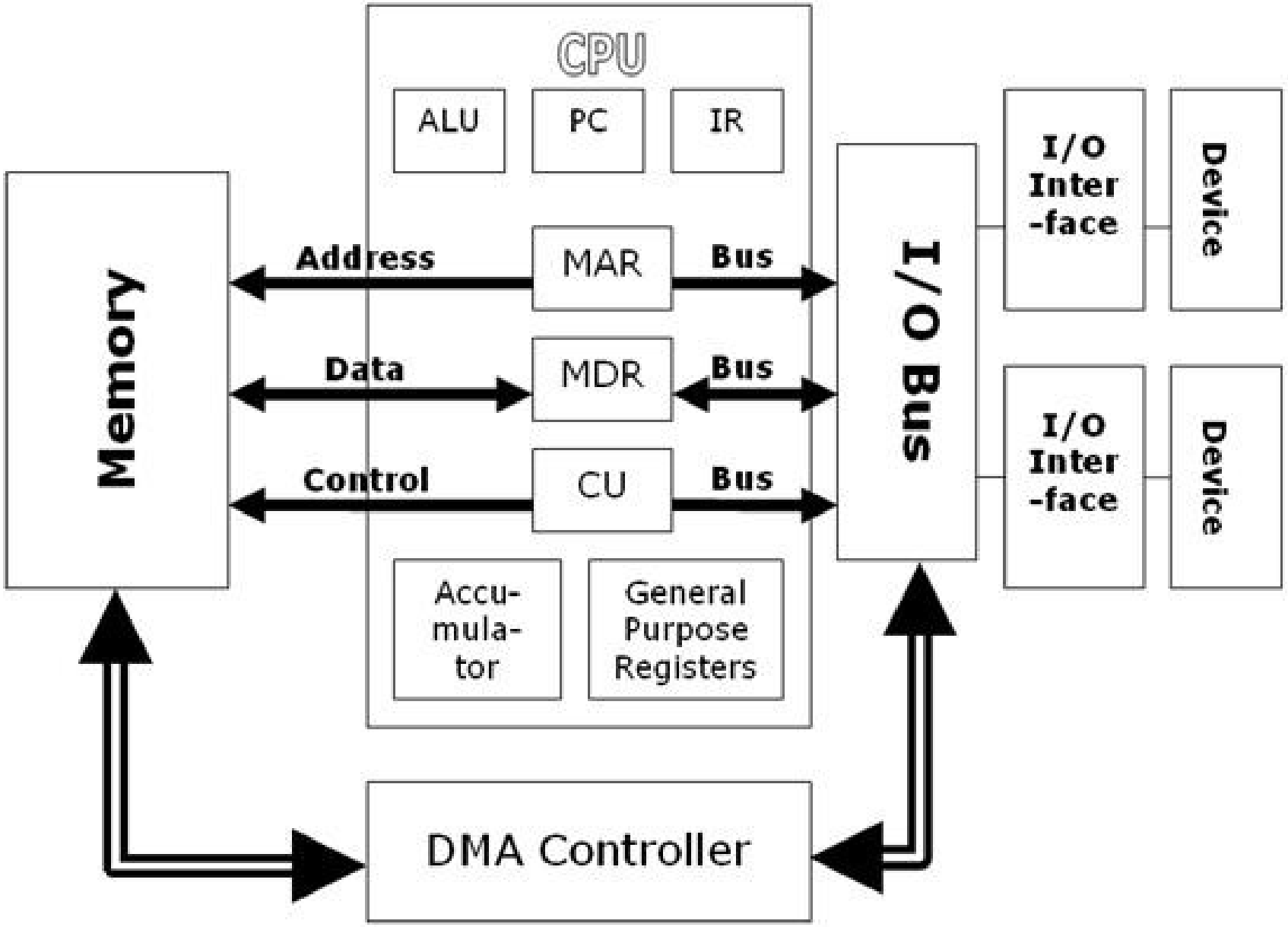
Прерывания от внешних устройств

```
           CPU0
 0:        134    XT-PIC-XT-PIC    timer
 1:        297    XT-PIC-XT-PIC    i8042
 2:          0    XT-PIC-XT-PIC    cascade
 8:          0    XT-PIC-XT-PIC    rtc0
 9:       5495    XT-PIC-XT-PIC    acpi, vboxguest, p7p1
10:       2088    XT-PIC-XT-PIC    p2p1
11:      22494    XT-PIC-XT-PIC    ahci, ohci_hcd:usb1
12:       1580    XT-PIC-XT-PIC    i8042
14:          0    XT-PIC-XT-PIC    ata_piix
15:       4186    XT-PIC-XT-PIC    ata_piix
```

DMA (Direct Memory Access)

- «Быстрые» устройства могут использовать DMA для пересылки данных в память в обход ЦП.
  - SATA-контроллер копирует считанный блок непосредственно в ОЗУ.
  - Ethernet-контроллер забирает пакет для передачи в сеть непосредственно из ОЗУ
- ЦП освобождается от пересылки данных
- Не «загрязняется» кеш процессора

DMA


CPU
ALU
PC
IR
Memory
Address
MAR
Bus
Data
MDR
Bus
Control
CU
Bus
I/O Bus
I/O Inter -face
Device
I/O Inter -face
Device
Accu- mula- tor
General Purpose Registers
DMA Controller

Алгоритм чтения сектора SATA

- ЦП выдает команду «считать сектор S по адресу в памяти M и выдать прерывание по готовности»
- SATA-контроллер считывает сектор S во внутренний буфер
- SATA-контроллер запрашивает DMA и копирует содержимое по адресу в памяти M
- После окончания копирования шлет запрос на прерывание на ЦП

Мультипрограммирование

- Организация выполнения нескольких программ одновременно (с перекрытием по времени) на одном компьютере
  - Распределение ОЗУ между работающими программами
  - Распределение времени ЦП (ядер ЦП) между программами
  - Управление внешними устройствами и организация доступа к ним

Требования к аппаратуре

- Поддержка прерываний
  - Для гарантированной корректной обработки событий от внешних устройств
- Таймер
  - Для гарантированного «справедливого» распределения времени между программами
- Защита памяти
  - Программа может работать только со своей памятью, доступ в чужую должен быть закрыт
- Режим супервизора
  - Чтобы ограничить несанкционированное использование пп. 1-3

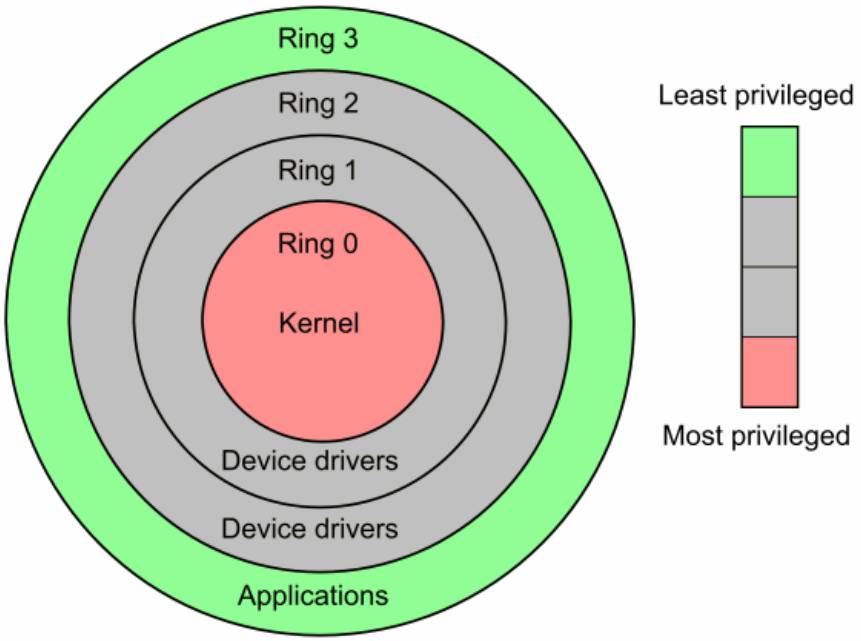
Защищенный режим x86

- При включении процессор работает в реальном режиме (real mode)
- Может быть переключен в защищенный режим (protected mode)
  - Регистр CR0 содержит бит включения (0)

```
mov eax, cr0
or  eax, 1
mov cr0, eax
```

Кольца защиты

- 0 — самое защищенное (kernel mode)
- 3 — наименее защищенное
- Уровень защиты текущей программы — биты IOPL регистра EFLAGS



Пользовательский режим

- Попытка выполнения привилегированной инструкции — исключение GPF (General Protection Fault) (#13)
- Доступ к портам ввода-вывода может быть разрешен с помощью карты доступа в TSS (Task State Segment)
- Векторы обработки прерываний располагаются с адреса в регистре IDTR (Interrupt Descriptor Table Register)
- При переключении уровня привилегий из TSS загружаются регистры SS, ESP и для стека используются они
- Все регистры процессора сохраняются в TSS

Защита памяти

- Ограничить доступ программы к памяти, которая ей не принадлежит
- Но! Очень часто можно использовать одну и ту же память для разных программ (разделяемые библиотеки (DLL, SO), сегмент кода программы)
- Облегчить перемещение программы по оперативной памяти

Виртуальная память

- Программно-аппаратный механизм трансляции адресов памяти программы (виртуальных адресов) в физические адреса.
- Каждая программа работает как если бы она была единственная загруженная в память
- Для x86 до 4GiB виртуальной памяти на процесс (обычно 2GiB или 3GiB)

Страничная виртуальная память

- Память разбивается на блоки фиксированного размера (страницы)
- Размер страницы на x86/x64 — обычно 4KiB (0x1000)
- Каждая страница начинается с кратного размеру страницы адреса
  - 0x00000000 — 0x00000fff — нулевая страница (zero page)

Отображение страниц

- Страница может быть не отображена (отсутствовать в виртуальном адресном пространстве) — при попытке обращения PageFault
- Страница может быть отображена на любую страницу физической памяти
- Каждая отображенная страница имеет права доступа «чтение», «запись», «выполнение»
- При нарушении прав - PageFault

Отображение страниц

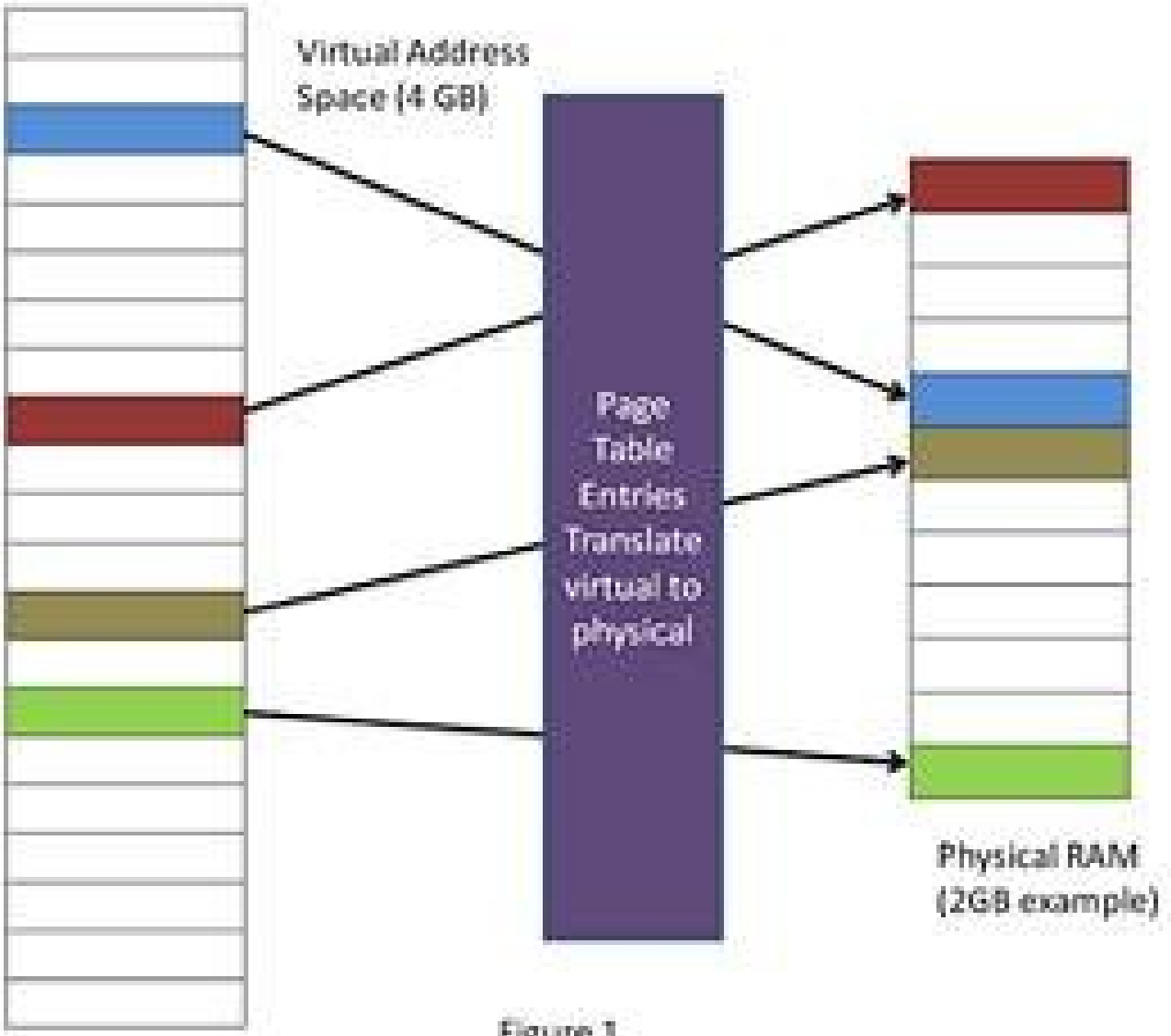


Figure 1

PageFault

- Исключение PageFault не обязательно ошибка
  - Страница данных откачана в swap
  - Страница кода не загружена из файла
  - Запись в страницу созданную для copy-on-write
- Обработчик исключения определяет причину PageFault. Если PageFault произошел из-за ошибки, ошибка передается в программу

Таблица страниц

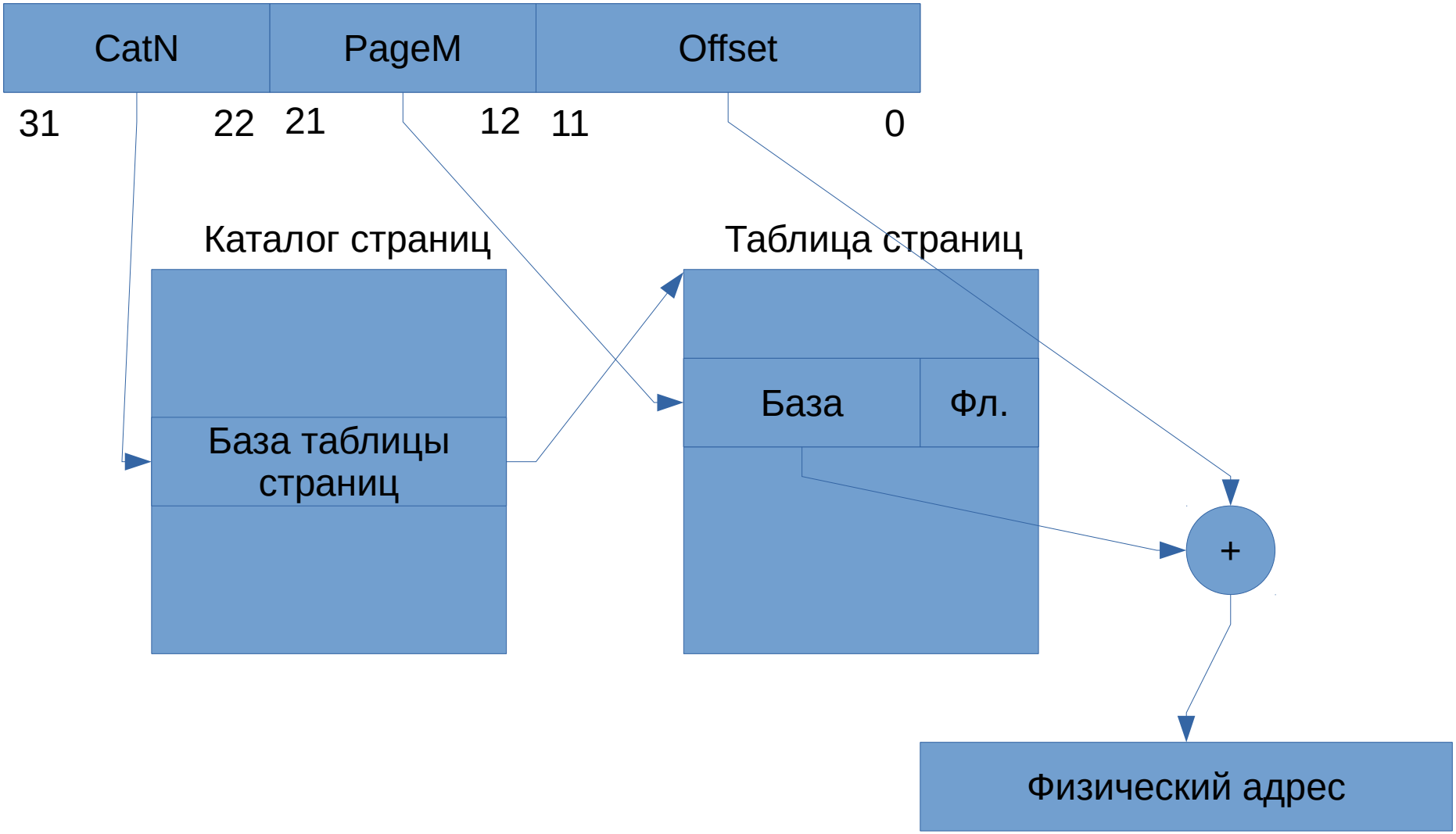


Таблица страниц

- CR2 указывает на начало каталога страниц
- X86 — двухуровневая таблица страниц, размер страницы — 4KiB, в каталоге страниц 1024 записи, в каждой таблице страниц 1024 записи, одна запись — 4 байта
- X64 — четырехуровневая таблица страниц, размер страницы — 4KiB, в таблице каждого уровня 512 записей, одна запись — 8 байт.

TLB (Translation Lookaside Buffer)

- Двухуровневая таблица страниц может потребовать 2 вспомогательных обращения к памяти!
- TLB — кэш-память для отображения виртуального адреса в физический
- TLB может быть многоуровневым и разделенным: для Intel Nehalem:
  - 64 записи в L1 DTLB
  - 128 записей в L1 ITLB
  - 512 записей в L2 TLB

Переключение контекста

- Происходит при прерываниях, системных вызовах, исключениях
- Влияние на производительность:
  - Время на сохранение контекста (все регистры)
  - Сброс кэшей процессора (предсказание переходов, микроинструкций)
  - Загрязнение/сброс кэша процессора
  - Очистка TLB
- Цена переключения от ~3000 тактов (1мкс)