

Учебный курс

Архитектура ЭВМ и язык ассемблера

Лекция 5

заместитель министра связи и массовых
коммуникаций РФ, старший преподаватель

Северов Дмитрий Станиславович

Часть оконных примитивов

- Структуры сложных данных
 - Точка
 - Область экрана
 - Системное сообщение
 - Класс окна
- Необходимые процедуры
 - `MessageBox` : Вывод текста в граф.режиме
 - `WinMain` : Инициализация граф.приложения
 - `WinProc` : Обработка системных событий
 - `ErrorHandler` : Обработка ошибок (необяз.)

Оконный сценарий

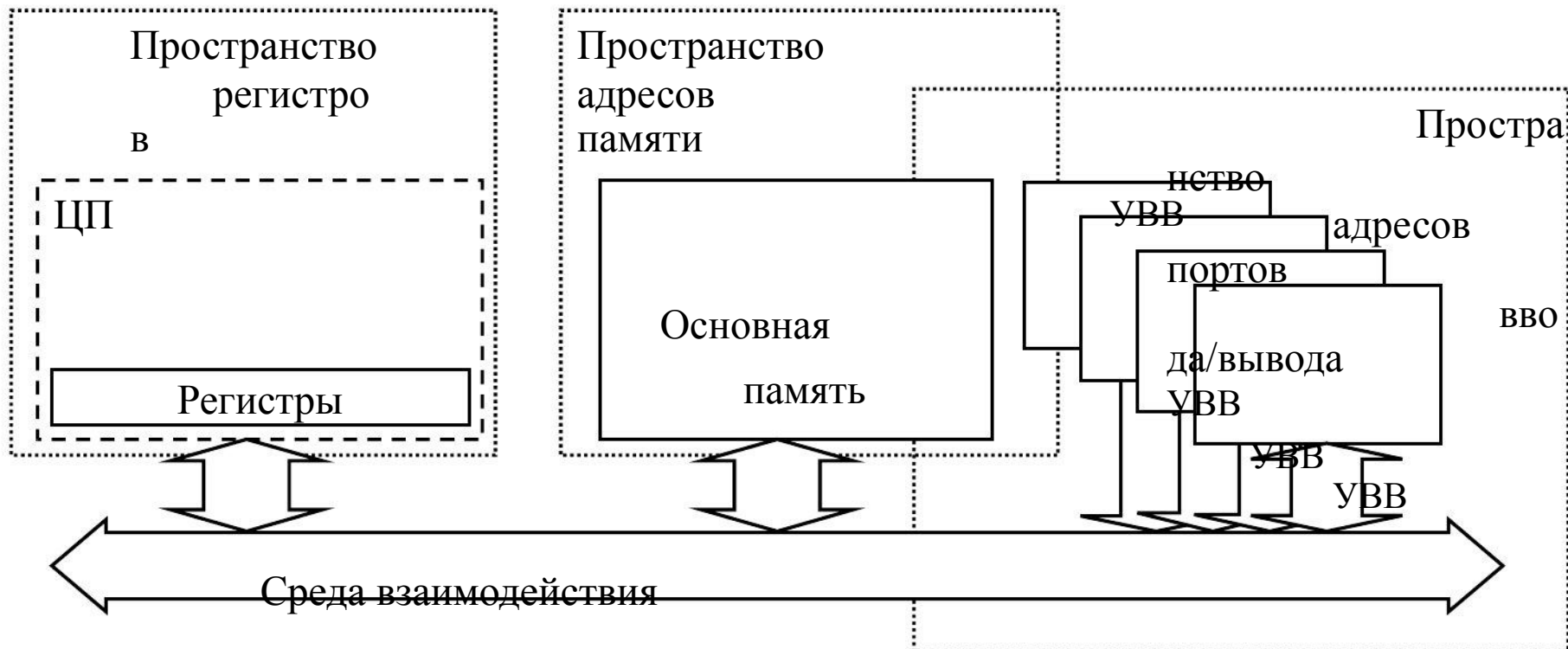
```
.386
.model flat, STDCALL
INCLUDE GraphWin.inc
.data
; Определим структурную переменную,
; описывающую класс окна
.code
WinMain PROC
; Определим дескриптор текущего процесса
; Загрузим образы пиктограммы и курсора
; программы.
; Зарегистрируем класс окна
; Создадим основное окно программы
; - если ошибка, отобразим сообщение и
; остановимся
; Отообразим окно на экране и обновим его
; содержимое
; Выведем приветственное сообщение
; Создадим цикл обработки сообщений
; Получим новое сообщение из
очереди
; Если в очереди больше нет
сообщений,
остановимся
```

```
WinProc PROC,
; Эта процедура обрабатывает
некоторые
сообщения, посылаемые
системой Windows
нашему приложению.
; Щелчок левой кнопкой мыши?
; Окно создано?
; Окно закрыто?
; Обработка остальных сообщений
выполняется
стандартной процедурой
системы Windows.
WinProc ENDP
-----
-
ErrorHandler PROC
; Получим код ошибки
; Определим адрес текстового
сообщения об
ошибке
```

Иерархия обращения к УВВ и «УВВ»

- Средства ЯВУ: операции, операторы
- Библиотеки функций `call/INVOKE`
 - ЯВУ: стандартные и «фирменные»
 - специальные
 - системные
- Системный вызов: `int`
- Обращение к BIOS (RM/VM): `int`
- Обращение к памяти и портам в/в: `in, out`

Порты ввода/вывода



IN аккумулятор, порт
OUT порт, аккумулятор

- Порт
 - 00h–FFh
 - DX
- Аккумулятор
 - AL
 - AX
 - EAX

```

        TITLE  Программа включения динамика   (Spkr.asm)
INCLUDE Irvine16.inc
timer    speaker    EQU    61h    ; Порт управления
        ; Порт управления таймером
        ; Динамиком
.code
        delay1    EQU    500    ;
        delay2    EQU    0D000h ; Задержка между сменой
        ; Определим состояние динамика
        in    al, speaker
        push ax    ; Сохраним байт состояния
        or    al, 00000011b ; Установим два младших бита
        out    speaker, al ; Включим динамик
L2:    mov    al, 60    ; Начальная высота тона
        out    timer, al ; Запустим таймер
L3:    push cx, delay1 ; Сохраним задержку в секундой циклов
        mov    cx, delay2
L3a:  loop L3a    ; Внутренний цикл
        pop    cx    ; задержки
        loop L3    ; Внешний цикл задержки
        sub    al, 1
        jnz    L2    ; Поведем тон
        pop    ax    ; Играем следующую ноту
        and    al, 11111100b ; Восстановим байт
        ; состояния
        out    speaker, al ; Сбросим 2 младшие бита
        ; Выключим динамик
        exit

```

Пример обращения к портам в/в

Прерывания (RM/VM)

- Вход прерываний и контроллер прерываний
- Таблица векторов прерываний
- Действия при поступлении прерывания
 - установка сигнала и передача процессору номера
 - сохранение флагов **FLAGS** и счётчика команд **CS:IP**
 - передача управления по вектору прерывания
 - <действия по обработке прерывания>
 - Восстановление флагов и счётчика команд **iret**
- Разрешение и запрет прерываний: **sti** и **cli**
- Программные прерывания **int <номер>**

```

TITLE Вывод цветного текста в окно (TextWin.asm)
; Отображает на экране цветное окно и выводит в него текст.
INCLUDE Irvine16.inc
.data
message "Текст, выводимый в окно", 0
.code
main PROC
    mov ax, @data
    mov ds, a
; Прокрутим окно

    mov ax, 0600h ; Номер функции
    mov bh, (blue SHL 4) OR yellow ; Атрибут
    mov cx, 050Ah ; Координаты левого верхнего
угла; Переместим курсор внутрь окна
    mov dx, 0A20h ; Номер функции; Координаты правого нижнего
угла; Строка 7, столбец 20
    mov dx, 0714h ; Строка 7, столбец 20
    mov bh, 0 ; Видеостраница 0
    int 10h
; Выведем текст в окно
    mov dx, OFFSET message
    call WriteString
; Ждем нажатия на любую клавишу
    mov ah, 10h
    int 16h
    exit
main ENDP
END main

```

Пример обращений к BIOS

Средства BIOS для работы с...

- клавиатурой – `int 16h` (`09h`, `1Bh`)
- экраном – `int 10h`
- мышью – `int 33h`
- параллельным интерфейсом – `int 17h`
- последовательным интерфейсом – `int 14h`
- списком устройств – `int 11h`
- системным таймером – `int 1Ah` (`1Ch`)
- эмулятором плавающей точки – `int 14h`
- диском на уровне секторов – `int 13h`
- ...

Пример обращения к DOS

```
TITLE Hello World Program(Hello.asm)
; Эта программа отображает "Hello, world!"

.model small
.stack 100h
.386

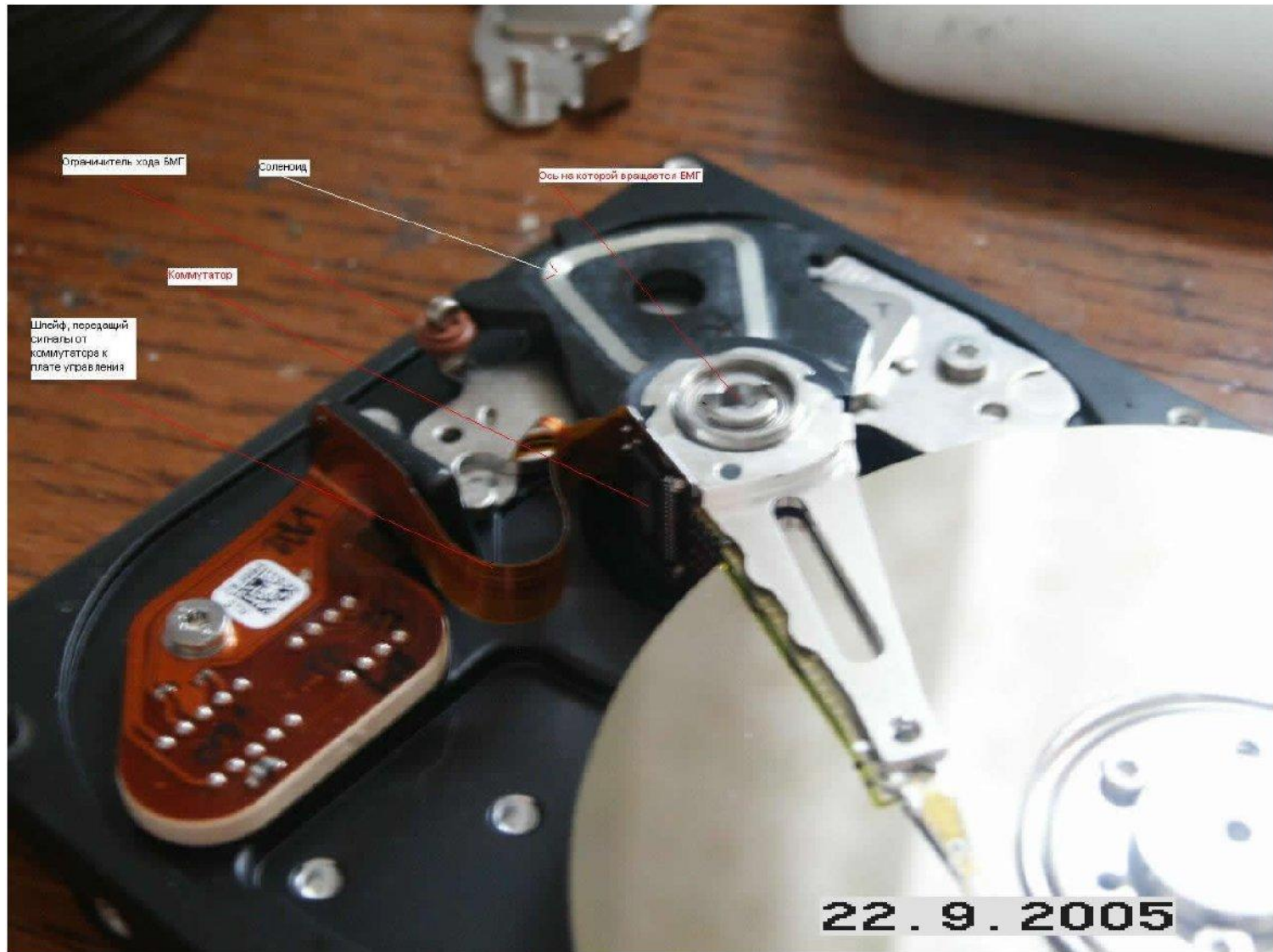
.data
message BYTE "Hello, world!",0dh,0ah

.code
main PROC
    mov ax,@data
    mov ah,40h ; функция вывода в канал
    mov bx,1 ; дескриптор стандартного вывода
    mov cx,SIZEOF message ; количество байт
    mov dx,OFFSET message ; адрес буфера
    int 21h
    .exit
main ENDP
END main
```

Средства DOS: int 21

- Чтение/запись
 - стандартные каналы
 - последовательный интерфейс
 - параллельный интерфейс
- клавиатура и экран
- Работа с файлами и каталогами
- Работа с системным временем
- Работа с памятью
- Работа с программами/процессами
- Работа с диском «напрямую»

«Жёсткие» диски



Адресация диска и старт системы

- Устройство типа «hard disk drive»
 - Сектор – «квант» обмена: 512 байт
 - Геометрия: цилиндр/дорожка/сектор, трансляция, LBA
- BIOS - Basic Input/Output System
 - POST-Power-On Self Test
 - логическая адресация секторов
 - выбор устройства
- Загрузка с диска: Master Boot Record/Partition Table
- Выбор раздела: Primary, Extended, Logical, Active
- Загрузка с раздела: Boot Sector/Volume
- Загрузка с тома: OS Loader/File system
- Загрузка из файлов: OS

Файловые системы

- Кластер – квант размещения данных
 - эффективность использования пространства
- FAT: Таблица размещения кластеров
 - разрядность элемента: 12, 16, 32 бит
 - секторов в кластере: $2^0, 2^1, \dots, 2^6$
 - кластеров в томе: $\leq 4087, \leq 65526, \leq 268\ 435\ 456$
- NTFS
 - Unicode
 - многодисковый том
 - права доступа, квоты
 - шифрование, сжатие
 - журналирование, зеркалирование

Форма с плавающей запятой

- Экспоненциальная форма $X = \pm m q^{\pm p}$



- Смещение порядка



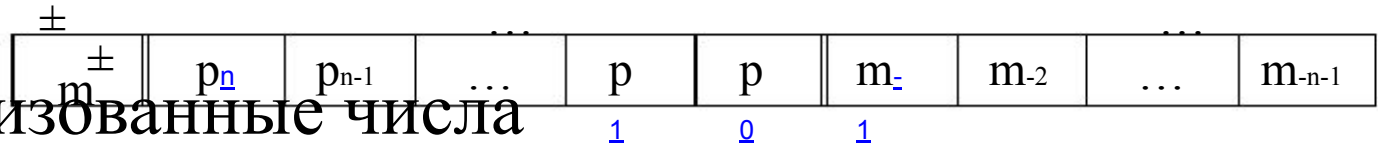
m

- Нормализация, скрытая единица

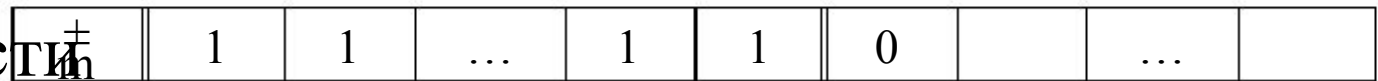


$$\begin{matrix} m_0 = 1 \\ m_0 = 0 \end{matrix}$$

- Денормализованные числа

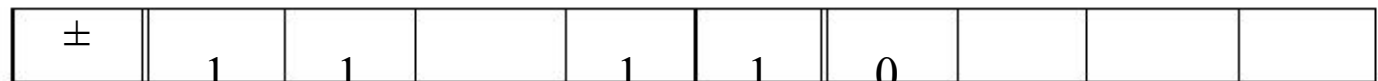


- Бесконечности



- Не числа

– гласные



– тихие



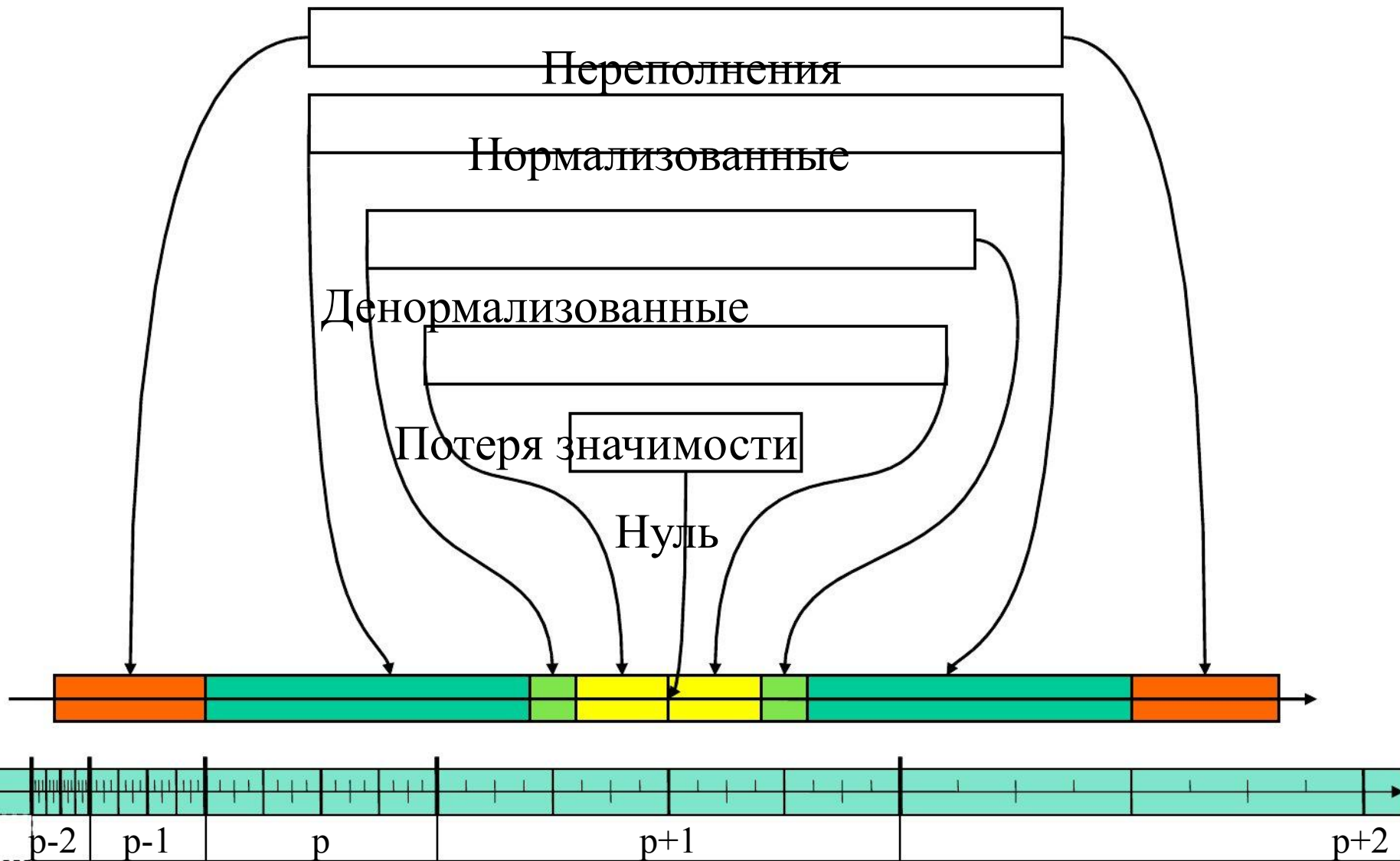
m

15

$m-2$

$m-n-1$

Что же представимо ?



Особенности операций

- Переполнение: сверхбольшой порядок
- Потеря значимости: сверхмалый порядок
- Неоднозначность (нуля)
- Особые случаи:
 - ненормализованные, бесконечности, «не числа»
- Бесконтрольная относительная погрешность
 - Вычитание близких
 - Сложение разномасштабных
- Неассоциативность
- Проблемы округления
 - <http://www.delphikingdom.com/asp/viewitem.asp?catalogid=374>
 - <http://www.delphikingdom.com/asp/viewitem.asp?catalogid=1217>

Стандарт iEEE 754

http://en.wikipedia.org/wiki/iEEE_floating-point_standard

	Одинарный	Одинарный расширенный	Двойной	Двойной расширенный
Слово (бит)	32	≥ 48	64	≥ 79
Порядок (бит)	8	≥ 11	11	≥ 15
Смещение порядка	127	-	10	-
Значения порядка	$-126 \div 127$	$\leq -1022 \div$ ≥ 1023	23	$\leq -16382 \div$ ≥ 16383
Мантисса (бит)	23/2	≥ 31	52/53	≥ 63

- Округления
 - round-down
 - round-half-up
 - round-ceiling
 - round floor

- Ещё округления
 - **round-half-even**
 - round-half-down
 - round-up

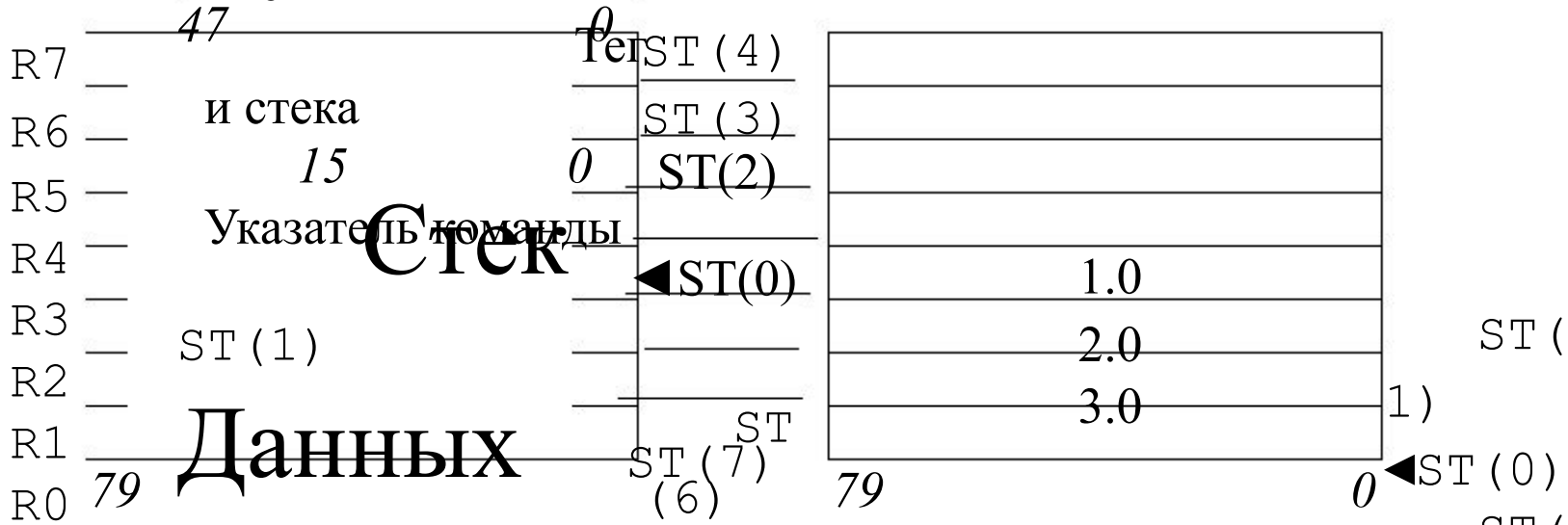
Модель «сопроцессора» FPU

Обмен с CPU через память

- Команду выбирает CPU
- Данные выбирает CPU
- CPU ждёт FPU



– FWAIT/WAIT



Детали «сопроцессора»

- Статус

Итоговая ошибка

Ошибка стека

Код условия

– Вершина стека

– Флаги исключений

- Неверная операция
- Денормал. операнд
- Деление на нуль
- Переполнение
- Потеря точности
- Ошибка

точности

- Управление

– Управление точностью

- Мантисса 24|53|64 бита

– Управление округлением

– Маски исключений

- Теги содержимого стека

– допустимое, нуль, прочее,
пуст

- Преобразование данных

– Целых, десятичных,
вещественных