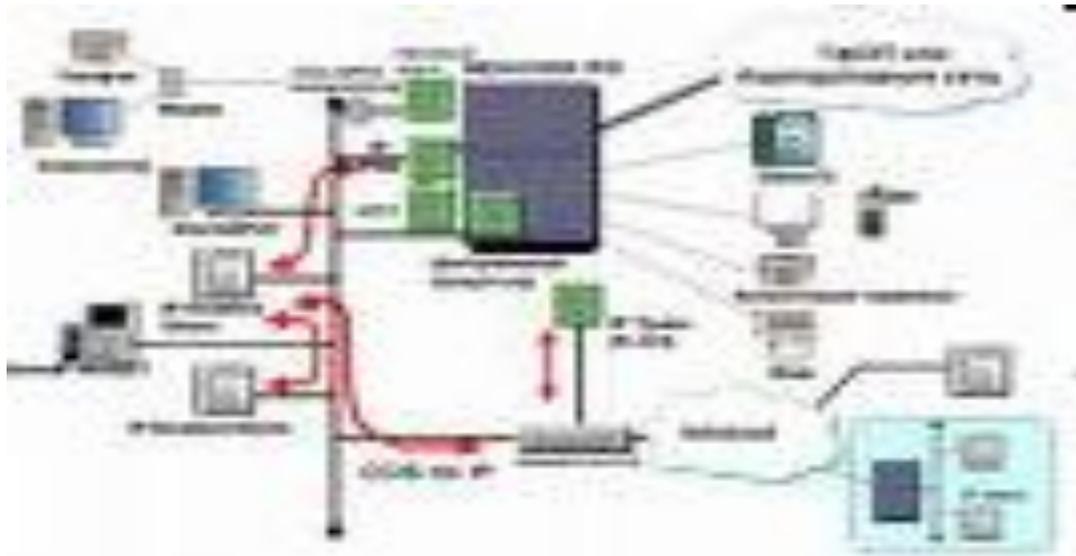




*Математический сопроцессор.
Форматы данных FPU.
Программная модель FPU.
Исключения и интерфейс сопроцессора*





Математический сопроцессор



Математический сопроцессор — сопроцессор для расширения командного множества центрального процессора и обеспечивающий его функциональностью модуля операций с плавающей запятой, для процессоров, не имеющих интегрированного модуля. Он может как представлять собой устройство в отдельном корпусе, так находиться вместе с центральным процессором на одном кристалле.





Основные понятия

Основным исполнительным устройством математического сопроцессора является устройство обработки чисел с плавающей запятой (FPU – floating point unit).

- Как правило, он не может самостоятельно выполнять программный код и получает поток инструкций от центрального процессора. Таким образом программы для центрального процессора и математического сопроцессора находятся в одном потоке инструкций.
- Однако сопроцессор при этом обладает собственным микропрограммным автоматом, набором регистров и исполнительными устройствами, таким образом являясь функционально независимым.





Схема взаимодействия центрального процессора и сопроцессора

- Сопроцессор подключен к шинам центрального процессора, а также имеет несколько специальных сигналов для синхронизации процессоров между собой.
- Часть командных кодов центрального процессора зарезервирована для сопроцессора, он следит за потоком команд, игнорируя другие команды. Центральный процессор, наоборот, игнорирует команды сопроцессора, занимаясь только вычислением адреса в памяти, если команда предполагает к ней обращение. Центральный процессор делает цикл фиктивного считывания, позволяя сопроцессору считать адрес с адресной шины. Если сопроцессору необходимо дополнительное обращение к памяти (для чтения или записи результатов), он выполняет его через захват шины.





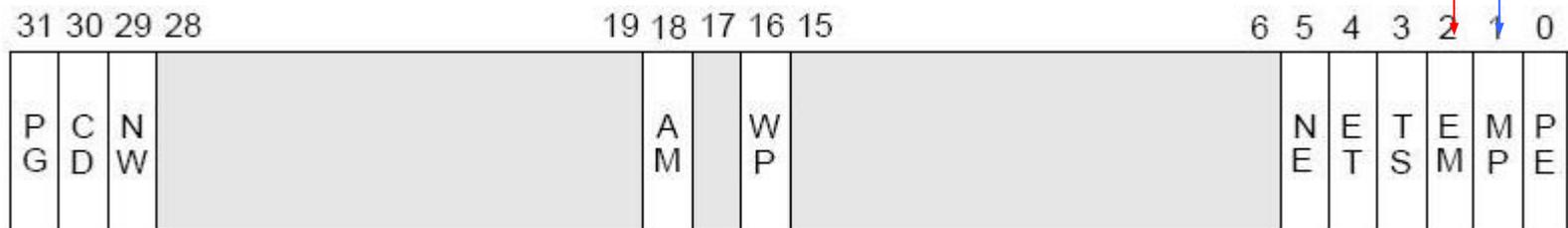
Схема взаимодействия центрального процессора и сопроцессора

- После получения команды и необходимых данных сопроцессор начинает её выполнение. Пока сопроцессор выполняет команду, центральный процессор выполняет программу дальше, параллельно с вычислениями сопроцессора. Если следующая команда также является командой сопроцессора, процессор останавливается и ожидает завершения выполнения сопроцессором предыдущей команды.
- Также существует специальная команда ожидания (FWAIT), принудительно останавливающая процессор до завершения вычислений (если для продолжения программы необходимы их результаты).



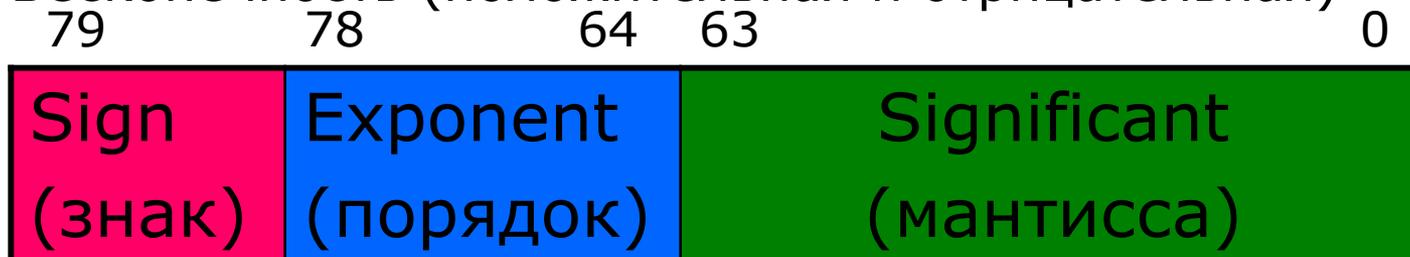
В случае отсутствия сопроцессора

- Прикладная программа использует инструкции FPU, но для этого ОС должна поддерживать эмуляцию сопроцессора. Эмулятор сопроцессора – это программа-обработчик прерывания от сопроцессора или исключения #NM, которая должна «выловить» код операции сопроцессора, определить местонахождение данных и выполнить требуемые вычисления, используя целочисленную арифметику CPU.
- Понятно, что эмуляция будет выполняться медленней, чем те же действия выполняемые настоящим FPU. Для этого в регистре CR0 должны быть установлены сочетания флагов EM=1, MP=0.



Форматы данных FPU позволяет представлять следующие категории чисел

- Нули (положительный и отрицательный) – оба значения эквивалентны;
- Денормализованные конечные числа (положительные и отрицательные);
- Нормализованные конечные числа (положительные и отрицательные);
- Бесконечность (положительная и отрицательная)



Sign определяет знак: 0 – «+»; 1 – «-».

Exponent хранит смещенное значение двоичного порядка

Significant для нормализованного числа $1 < M < 2$;

для денормализованного числа $0 < M < 1$.





Не-числа

- Кроме вещественных чисел (конечных нормализованных и денормализованных, нулей и бесконечности) регистры сопроцессора могут содержать не-числа NaN (Not a Number) четырех видов:
 1. **-SNaN** и **+SNaN** – порядок 111...111, мантисса **1,0xxx...xxx** (ненулевая). Эти сигнализирующие не-числа (signaling NaN) вызывают исключения сопроцессора, если с ними пытаются выполнить арифметические действия.
 2. **-QNaN** и **+QNaN** – порядок 111...111, мантисса **1,1xxx...xxx** (ненулевая). Эти «тихие» не-числа (quiet NaN) не вызывают исключений при арифметических операциях.





Не-числа могут определять такие случаи

- неопределённость (IND), возникает при комплексном результате (например, при вычислении квадратного корня из отрицательного числа) и в некоторых других случаях,
- недействительное значение (qNaN, sNaN) - может использоваться компилятором (для предотвращения использования неинициализированных переменных) или отладчиком,
- нуль - в формате с плавающей запятой, нуль также считается специальным значением.





Устройство FPU

Модуль операций с плавающей запятой представляет собой стековый калькулятор, работающий по принципу обратной польской записи. Перед операцией аргументы помещаются в LIFO-стек, при выполнении операции необходимое количество аргументов снимается со стека. Результат операции помещается в стек, где может быть использован в дальнейших вычислениях или может быть снят со стека для записи в память. Также поддерживается и прямая адресация аргументов в стеке относительно вершины.

Поддерживаемые математические операции: арифметические операции, сравнение, деление по модулю, округление, смена знака, модуль, квадратный корень, синус, косинус, частичный тангенс, частичный арктангенс, загрузка константы (0, 1, число пи, $\log_2(10)$, $\log_2(e)$, $\lg(2)$, $\ln(2)$) и некоторые другие специфические операции.



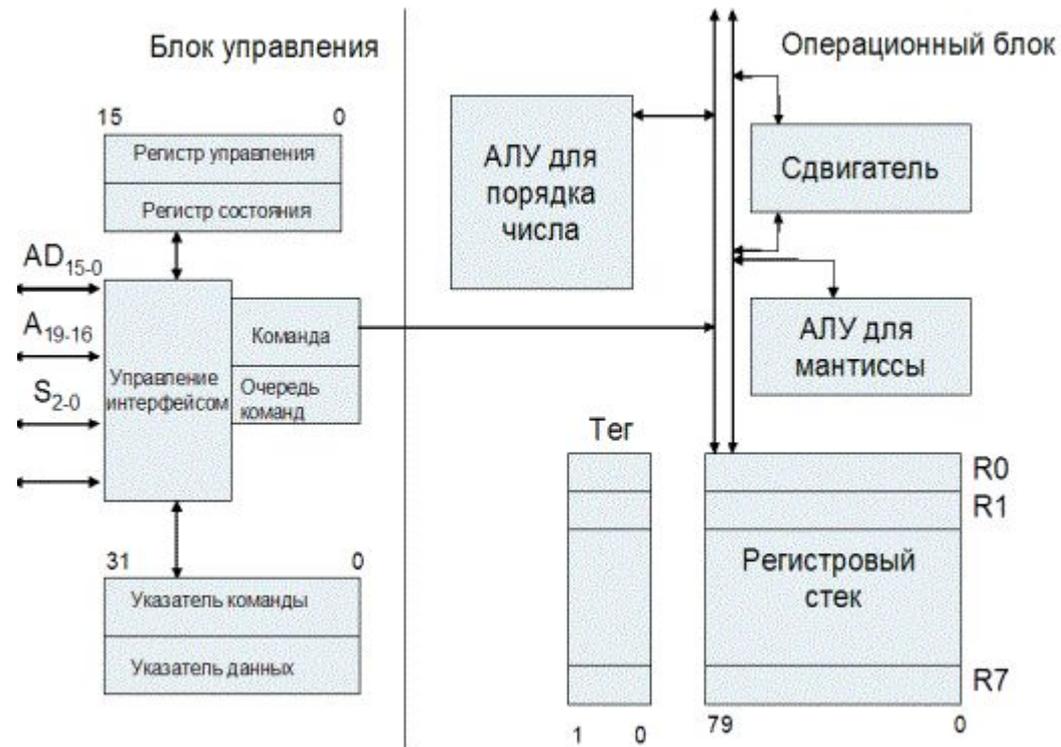
Форматы чисел, поддерживаемых сопроцессором

Тип	Длина	Точность		Диапазон нормализованных значений	
		двоичная	десятичная	двоичный	десятичный
<i>Вещественные числа</i>					
Одиночные (single)	32	24	7	$2^{-126} \leq x \leq 2^{127}$	$1,18 \cdot 10^{-38} \leq x \leq 3,4 \cdot 10^{38}$
Двойные (double)	64	53	15-16	$2^{-1022} \leq x \leq 2^{1023}$	$2,23 \cdot 10^{-308} \leq x \leq 1,79 \cdot 10^{308}$
Расширенные (extended)	80	64	19	$2^{-16382} \leq x \leq 2^{16383}$	$3,37 \cdot 10^{-4932} \leq x \leq 1,18 \cdot 10^{4932}$



Программная модель FPU

С программной точки зрения сопроцессор содержит блок регистров данных, регистр управления и группу регистров состояния и указателей.





Программная модель сопроцессоров





Для хранения данных в сопроцессоре предназначены регистры R0-R7. Эти регистры организованы в стек, и доступ к ним производится относительно вершины стека - ST. Номер регистра, соответствующего вершине стека, хранится в регистре состояния (поле TOS). Как и у ЦП, стек сопроцессора растет к регистрам с меньшими адресами. Команды, которые производят запоминание и извлечение из стека, передают данные из текущего регистра ST, а затем производят инкремент поля TOS в регистре состояния. Многие команды сопроцессора допускают неявное обращение к вершине стека, обозначаемой ST или ST(0). Для указания i-го регистра относительно вершины используется обозначение ST(i), где $i = 0, \dots, 7$. Например, если поле TOS регистра состояния содержит значение 0112 (вершиной стека является регистр R3), то команда FADD ST,ST(2) суммирует содержимое регистров R3 и R5. Стековая организация упрощает программирование подпрограмм, допуская передачу параметров в регистровом стеке сопроцессора.



Старший байт регистра состояния содержит:

- Бит занятости V (бит 15), сигнализирующий, свободен ли сопроцессор ($V=0$) или занят выполнением численной команды ($V=1$). Команды сопроцессора, которые используют регистровый стек, требуют, чтобы перед началом их выполнения сопроцессор не был занят. В FPU этот бит дублирует значение флага сигнализации особого случая.
- Поле TOS "Top-Of-the-Stack" (биты 11-13), которое содержит номер регистра, являющегося логической вершиной стека. При помещении в регистровый стек нового числа это значение уменьшается. Если это поле достигает значения 0, возможны две ситуации: произойдет заворачивание стека на регистры с большими номерами (т.е. R7) или, если заворачивание приведет к затиранию несохраненного значения, возникнет особый случай сопроцессора "переполнение стека".
- 4 бита кода условия (биты 14, 10, 9, 8), аналогичные флажкам состояния FLAGS у IA-32, отражающие результат арифметических операций. Эти флажки могут быть использованы для условных переходов.





- Младший байт регистра состояния содержит флажки особых случаев сопроцессора: переполнение стека, потеря точности, потеря значащих разрядов, численное переполнение, деление на ноль, денормализация, недействительная операция. Если соответствующий особый случай возник и не был замаскирован (в слове управления), это приведет к генерации центральным процессором особого случая сопроцессора (#16).
- Младшее слово регистра управления содержит биты масок особых случаев. Сопроцессор допускает индивидуальное маскирование особых случаев. Если тот или иной особый случай замаскирован, при возникновении соответствующей ситуации сопроцессор выполняет некоторые заранее определенные внутренние действия, которые зачастую приемлемы для большинства применений. Например, если замаскирован особый случай деления на ноль, то выполнение операции $85,32/0$ даст результат $+\infty$.
- Старшее слово регистра управления содержит два поля: управление точностью PC (биты 8 и 9) и управление округлением PC (биты 10 и 11). Биты управления точностью можно использовать для понижения точности вычислений. По умолчанию используется расширенная точность ($PC = 112$), можно также использовать двойную точность ($PC = 102$) и одинарную точность ($PC = 002$).



Регистр тегов содержит 8 тегов - признаков, характеризующих содержимое соответствующего численного регистра сопроцессора. Тег может принимать следующие значения:

- 00 - в регистре находится действительное число;
- 01 - нулевое число в регистре;
- 10 - недействительное число (бесконечность, денормализованное число, не-число);
- 11 - пустой регистр.



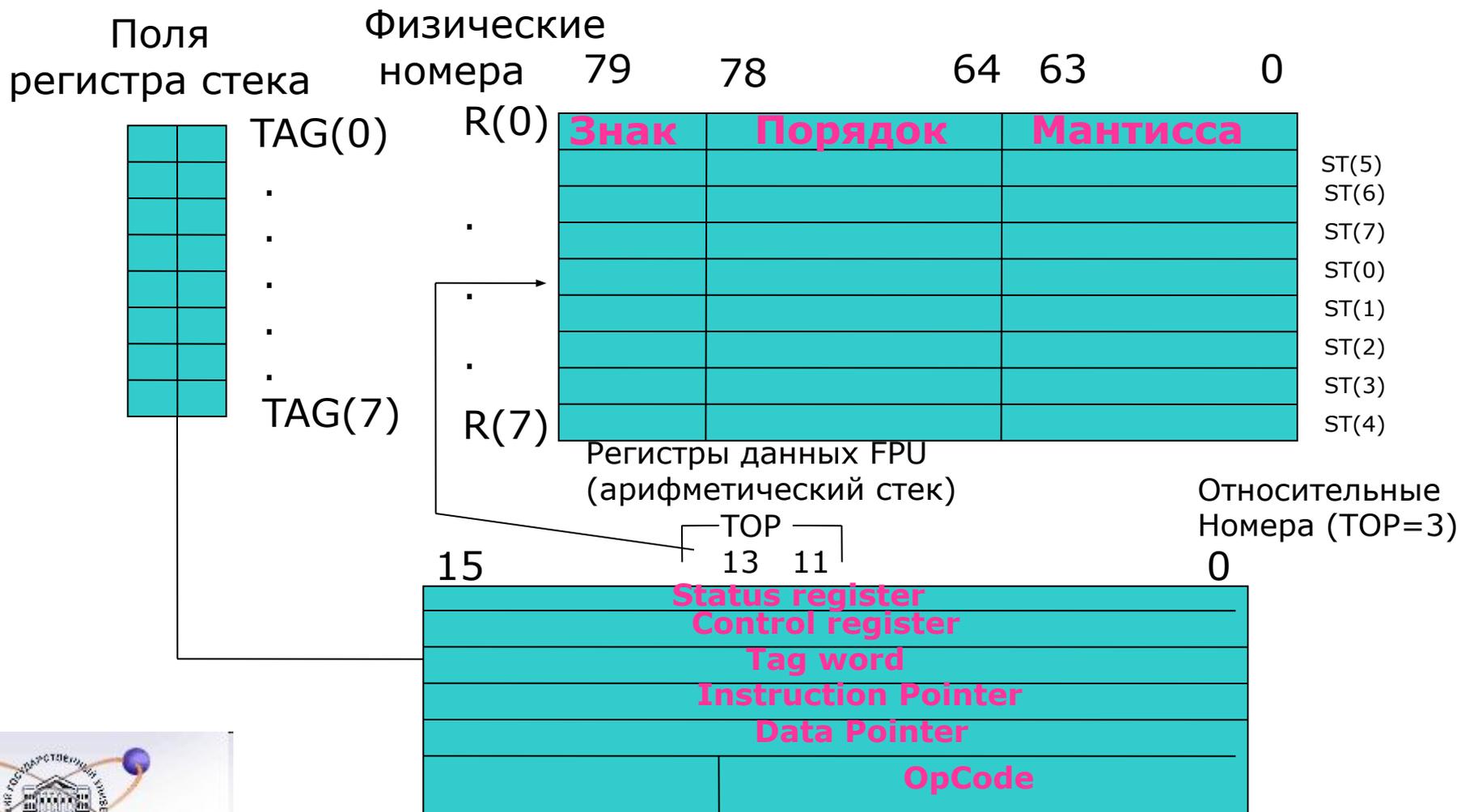


Сопроцессор использует теги, чтобы определить переполнение или опустошение стека. Если при помещении в стек очередного числа окажется, что декрементированный TOS указывает на непустой регистр (соответствующий тег не равен 112), сопроцессор сигнализирует о переполнении стека. Если при извлечении из стека числа окажется, что инкрементированный TOS указывает на пустой регистр (соответствующий тег равен 112), сопроцессор сигнализирует об опустошении стека.

Для команд сопроцессора выделена группа кодов, начинающаяся с 11011xxx. В ассемблерах мнемоники команд сопроцессора начинаются с "F": FADD (сложение), FCOM (сравнение), FCOS (косинус), FDIV (деление) и т.п.



Регистры математического сопроцессора



Используемая литература:

- Книга «Процессоры «Pentium 4, Athlon и Duron»», авторы Михаил Гук, Виктор Юров
- Книга «Ассемблер. Учебник для ВУЗов», авторы Михаил Гук, Виктор Юров
- Книга «Архитектура ЭВМ», автор Мюллер
- <http://www.intuit.ru/department/hardware/csorg/8/2.htm>
- <http://ar-pc.ru/pc/200636/>
- soft-tlt.ru/programmsi1.html
- <http://cs.njit.edu/~sohn/cs650/>

