**Система команд сопроцессора (дополнительный материал по теме Асссемблер)**

**Команды передачи данных**

Группа команд передачи данных предназначена для организации обмена между регистрами стека, вершиной стека сопроцессора и ячейками оперативной памяти. С помощью этих команд осуществляются все перемещения значений операндов в сопроцессор и из него. В сопроцессоре имеются следующие команды передачи данных:

fld/fild источник — загрузка вещественного/целого числа из памяти в вершину стека сопроцессора;

fst/fist приемник — сохранение вещественного/целого числа из вершины стека сопроцессора в память. Сохранение числа в памяти не сопровождается выталкиванием его из стека, то есть текущая вершина стека сопроцессора не изменяется (поле top не изменяется);

fstp/fistp приемник — сохранение вещественного/целого числа из вершины стека сопроцессора в память. В отличие от предыдущей команды происходит выталкивание вещественного числа из стека после его сохранения в памяти. Команда изменяет поле top, увеличивая его на единицу. Вследствие этого вершиной стека становится следующий, больший по своему физическому номеру, регистр стека сопроцессора;

fxch st(i) — обмен значений между текущей вершиной стека и регистром стека сопроцессора ST(i).

**Команды загрузки констант**

В математических вычислениях достаточно часто встречаются предопределенные константы. Сопроцессор хранит значения некоторых из них. Для каждой предопределенной константы существует своя специальная команда:

fldz — загрузка нуля в вершину стека сопроцессора;

fld1 — загрузка единицы в вершину стека сопроцессора;

fldpi — загрузка числа π в вершину стека сопроцессора;

fldl2t — загрузка двоичного логарифма десяти в вершину стека сопроцессора;

fldl2e — загрузка двоичного логарифма е в вершину стека сопроцессора;

fldlg2 — загрузка десятичного логарифма двух в вершину стека сопроцессора;

fldln2 — загрузка натурального логарифма двух в вершину стека сопроцессора.

**Команды сравнения данных**

Команды данной группы выполняют сравнение значений числа, находящегося в вершине стека и операнда, указанного в команде:

fcom/ficom операнд — команда без операндов сравнивает два значения: одно находится в регистре ST(0), другое — в регистре ST(1). Если указан операнд, то сравнивается значение в регистре ST(0) стека сопроцессора со значением в памяти;

fcomp/ficomp операнд — команда сравнивает значение в вершине стека сопроцессора ST(0) со значением операнда, который находится в регистре или в памяти. После сравнения и установки бит с3–с0 команда выталкивает значение из ST(0). Длина целого операнда 16 или 32 бита, то есть целое слово и короткое целое;

ftst — команда не имеет операндов и сравнивает значения в ST(0) со значением 0. В результате работы команд сравнения в регистре состояния SWR устанавливаются следующие флаги c3, c2, c0: если операнды несравнимы, то с3=1, с2=1, с0=1; если операнд\_2 > операнд\_1 (ST(1) > ST(0)), то с3=0, с2=0, с0=1; если операнд\_2 < операнд\_1 (ST(1) < ST(0)), то с3=0, с2=0, с0=0; если операнд\_2 = операнд\_1 (ST(1) = ST(0)), то с3=1, с2=0, с0=0.

Для программирования реакции на результаты сравнения необходимо анализировать состояние этих бит. Сопроцессор не имеет для этого специальных команд, за исключением команды fstsw, которая позволяет записать содержимое регистра SWR в регистр АХ. Затем командой sahf содержимое регистра AH, в котором находятся биты с3, с2 и с0, записывается в младший байт регистра EFLAGS/FLAGS. Местоположение бит с3, с2 и с0 соответствует местоположению флагов ZF, PF и CF. Таким образом, после применения команды sahf становится возможным реагировать на результаты сравнения значений в сопроцессоре командами условного перехода основного процессора.

**Арифметические команды**

Команды сопроцессора, входящие в данную группу, реализуют четыре основные арифметические операции — сложение, вычитание, умножение и деление. Имеется также несколько дополнительных команд, предназначенных для повышения эффективности использования основных арифметических команд. Целочисленные арифметические команды:

fiadd источник — команда складывает значения ST(0) и целочисленного источника, в качестве которого выступает 16- или 32- разрядный операнд в памяти. Результат сложения запоминается в регистре стека сопроцессора ST(0);

fisub источник — команда вычитает значение целочисленного иточника из ST(0). Результат вычитания запоминается в регистре стека сопроцессора ST(0). В качестве источника выступает 16- или 32-разрядный целочисленный операнд в памяти;

fimul источник — команда умножает значение целочисленного источника на содержимое ST(0). Результат умножения запоминается в регистре стека сопроцессора ST(0). В качестве источника выступает 16- или 32- разрядный целочисленный операнд в памяти;

fidiv источник — команда делит содержимое ST(0) на значение целочисленного источника. Результат деления запоминается в регистре стека сопроцессора ST(0). В качестве источника выступает 16- или 32-разрядный целочисленный операнд в памяти. Вещественные арифметические команды:

fadd — команда складывает значения из регистров ST(0) и ST(1). Результат сложения запоминается в регистре стека сопроцессора ST(0);

fadd слагаемое — команда складывает значения ST(0) и слагаемого, представляющего собой адрес ячейки памяти. Результат сложения запоминается в регистре стека сопроцессора ST(0);

fadd st(i), st — команда складывает значение в регистре стека сопроцессора ST(i) со значением в вершине стека ST(0). Результат сложения запоминается в регистре ST(i);

faddp st(i), st — команда производит сложение вещественных операндов аналогично команде fadd st(i), st. Последним действием команды является выталкивание значения из вершины стека сопроцессора ST(0).

Результат сложения остается в регистре ST(i – 1);

fsub — команда вычитает из значения в регистре ST(1) значение регистра ST(0). Результат вычитания запоминается в регистре стека сопроцессора ST(1); fsub вычитаемое — команда вычитает значение вычитаемого из значения в ST(0). Вычитаемое представляет собой адрес ячейки памяти, содержащей допустимое вещественное число. Результат сложения запоминается в регистре стека сопроцессора ST(0);

fsub st(i), st — команда вычитает значение в вершине стека ST(0) из значения в регистре стека сопроцессора ST(i). Результат вычитания запоминается в регистре стека сопроцессора ST(i);

fsubp st(i), st — команда вычитает вещественные операнды аналогично команде fsubp st(i), st. Последним действием команды является выталкивание значения из вершины стека сопроцессора ST(0). Результат вычитания остается в регистре ST(i-1);

fmul — команда (без операндов) умножает значение в ST(1) на содержимое регистра в ST(0). Результат умножения запоминается в регистре стека сопроцессора ST(1);

fmul множитель — команда умножает множитель на содержимое регистра стека ST(0). Результат умножения запоминается в регистре стека сопроцессора ST(0);

fmul st(i), st — команда умножает значение в произвольном регистре ST(i) на содержимое регистра стека ST(0). Результат умножения запоминается в регистре стека сопроцессора ST(i);

fmulp st(i), st — команда производит умножение подобно команде fmul st(i), st. Последним действием команды является выталкивание значения из вершины стека сопроцессора ST(0). Результат умножения остается в регистре ST(i – 1);

fdiv — команда (без операндов) делит содержимое регистра ST(1) на значение регистра сопроцессора ST(0). Результат деления запоминается в регистре стека сопроцессора ST(0); fdiv делитель — команда делит содержимое регистра ST(0) на делитель. Результат деления запоминается в регистре стека сопроцессора ST(0);

fdiv st(i), st — команда производит деление содержимого регистра ST(i) на значение регистра сопроцессора ST(0). Результат деления запоминается в регистре стека сопроцессора ST(i);

fdivp st(i), st — команда производит деление аналогично команде fdiv st(i), st. Последним действием команды является выталкивание значения из вершины стека сопроцессора ST(0). Результат деления остается в регистре ST(i – 1). Дополнительные арифметические команды:

fsqrt — вычисление квадратного корня из значения, находящегося в вершине стека сопроцессора ST(0). Команда не имеет операндов. Результат вычисления помещается в регистр ST(0). Следует отметить, что данная команда обладает определенными достоинствами. Во-первых, результат извлечения достаточно точен, во-вторых, скорость исполнения чуть больше скорости выполнения команды деления вещественных чисел fdiv; fabs — вычисление модуля значения, находящегося в вершине стека сопроцессора — регистре ST(0). Команда не имеет операндов. Результат вычисления помещается в регистр ST(0);

fchs — изменение знака значения, находящегося в вершине стека сопроцессора — регистре ST(0). Команда не имеет операндов. Результат вычисления помещается обратно в регистр ST(0). Отличие команды fchs от команды fabs в том, что команда fchs только инвертирует знаковый разряд значения в регистре ST(0), не меняя значения остальных бит. Команда вычисления модуля fabs при наличии отрицательного значения в регистре ST(0) наряду с инвертированием знакового ряда выполняет изменение остальных бит значения таким образом, чтобы в ST(0) получилось соответствующее положительное число;

fxtract — команда выделения порядка и мантиссы значения, находящегося в вершине стека сопроцессора — регистре ST(0). Команда не имеет операндов. Результат выделения помещается в два регистра стека: мантисса — в ST(0), а порядок — в ST(1). При этом мантисса представляется вещественным числом с тем же знаком, что и у исходного числа, и порядком, равным нулю. Порядок, помещенный в ST(1), представляется как истинный порядок, то есть без константы смещения, в виде вещественного числа со знаком и соответствует величине p формулы A=(±M)\*N±(p); fscale — команда умножает или делит значение, находящееся в вершине стека сопроцессора ST(0), на степень двух. Значение показателя степени находится в регистре стека ST(1) и представляет собой целое число со знаком. Если показатель степени не является целым числом, то он округляется до целого в меньшую сторону. Команда не имеет операндов.

Команда fscale не очищает регистр ST(1). Результат помещается в регистр ST(0);

frndint — команда округления до целого значения — округляет значение, находящееся в вершине стека сопроцессора ST(0). Команда не имеет операндов.

**Команды трансцендентных функций**

Сопроцессор имеет ряд команд, предназначенных для вычисления значений тригонометрических функций, таких, как синус, косинус, тангенс, арктангенс, а также значений логарифмических и показательных функций. Наличие этих команд значительно облегчает разработку вычислительных алгоритмов. В системе команд сопроцессора имеются следующие трансцендентные команды:

fcos — команда вычисляет косинус угла, находящегося в вершине стека сопроцессора ST(0). Команда не имеет операндов. Результат возвращается в регистр ST(0);

fsin — команда вычисляет синус угла, находящегося в вершине стека сопроцессора ST(0). Команда не имеет операндов. Результат возвращается в регистр ST(0);

fsincos — команда вычисляет синус и косинус угла, находящегося в вершине стека сопроцессора ST(0). Команда не имеет операндов. Результат возвращается в регистры ST(0) и ST(1). При этом косинус помещается в ST(0), а синус — в ST(1);

fptan — команда вычисляет частичный тангенс угла в радианах, (в диапазоне –263…263), находящегося в вершине стека сопроцессора ST(0). Команда не имеет операндов. Результат возвращается в регистры ST(0) — значение единицы, и ST(1) — тангенс угла;

fpatan — команда вычисляет частичный арктангенс частного x/y (x находится в регистре ST(0), y — в регистре ST(1)). Команда не имеет операндов. Результат возвращается в регистре ST(0); fprem — команда получения частичного остатка от деления. Исходные значения делимого и делителя размещаются в стеке: делимое — в ST(0), делитель — в ST(1). Делитель рассматривается как некоторый модуль, поэтому в результате работы команды получается остаток от деления по модулю. Но произойти это может не сразу, так как этот результат в общем случае достигается за несколько производимых подряд обращений к команде fprem. Это происходит, если значения операндов сильно различаются. Физическая работа команды заключается в реализации хорошо известного действия деления в столбик. При этом каждое промежуточное деление осуществляется отдельной командой fprem. Цикл, центральное место в котором занимает команда fprem, завершается, когда очередная полученная разность в ST(0) становится меньше значения модуля в ST(1). Судить об этом можно по состоянию флага c2 в регистре состояния SWR: если с2 = 0, то работа команды fprem полностью завершена, так как разность ST(0) меньше значения модуля в ST(1); если с2 = 1, то необходимо продолжить выполнение команды fprem, так как разность ST(0) больше значения модуля в ST(1);

f2xm1 — команда вычисления значения функции y = 2x–1. Исходное значение x размещается в вершине стека сопроцессора в регистре ST(0) и должно лежать в диапазоне –1≤ x ≤ 1. Результат y замещает x в регистре ST(0). Эта команда может быть использована для вычисления различных показательных функций;

fyl2x — команда вычисления значения функции z = ylog2(x). Исходное значение х размещается в вершине стека сопроцессора, а исходное значение y — в регистре ST(1). Значение х должно лежать в диапазоне 0 ≤ x ≤ +∞, а значение у — в диапазоне –∞ ≤ у ≤ +∞. Перед тем, как осуществить запись результата z в вершину стека, команда fyl2x выталкивает значения х и у из стека и только после этого производит запись z в регистр ST(0); fyl2xp1 — команда вычисления значения функции z = ylog2(x+1). Исходное значение х размещается в вершине стека сопроцессора — регистре ST(0), а исходное значение y — в регистре ST(1). ). Значение х должно лежать в диапазоне 0 ≤ x ≤ 1–1/ 2 , а значение у — в диапазоне –∞ ≤ у ≤ +∞. Перед тем, как осуществить запись результата z в вершину стека, команда fyl2xp1 выталкивает значения х и у из стека и только после этого производит запись z в регистр ST(0).

**Команды управления сопроцессором**

Последняя группа команд предназначена для общего управления работой сопроцессора: wait/fwait — команда ожидания, предназначена для синхронизации работы процессора и сопроцессора;

finit — команда инициализации сопроцессора. Данную команду используют перед первой командой сопроцессора в программе и других случаях, когда необходимо привести сопроцессор в начальное состояние; fstsw назначение — команда сохранения содержимого регистра состояния SWR в ячейке памяти размером 2 байта или регистре AX;

ffree st(i) — команда освобождения регистра стека ST(i). Команда записывает в поле регистра тегов, соответствующего регистру ST(i), значение 11b, что соответствует пустому регистру.