**Лабораторная работа № 6.**

***Сопроцессор и его программирование***

Сопроцессор (**FPU, Floating Point Unit**) как модуль для выполнения операций над числами с плавающей запятой представляет из себя специальное устройство, имеющее общий с основным процессором (**CPU**) доступ к потоку команд и памяти данных. Начиная с модели 80486DX сопроцессор исполняется на одной микросхеме с процессором, но остаётся логически отдельным устройством. С точки зрения программиста сопроцессор представляет из себя множество регистров и набор команд, предназначенных для обработки собственных типов данных: три целых двоичных, один целый десятичный и три – с плавающей запятой.

Независимо от типа обрабатываемых данных, при загрузке в регистры сопроцессора все они преобразуются к внутреннему формату данных сопроцессора – 10-байтному вещественному формату. При сохранении результатов в память выполняется обратное преобразование в соответствии с сохраняемым типом. В следующей таблице даны сведения о типах данных сопроцессора.

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип данных** | **Число бит** | **К-во знач.**  **цифр** | **Диапазон** |
| Целое слово | 16 | 4 | –32768…32767 |
| Короткое целое | 32 | 9 | –2х109…2х109 |
| Длинное целое | 64 | 18 | –9х1018…9х1018 |
| Упаков. десятичное | 80 | 18 | –99..99…+99..99 (18 цифр) |
| Короткое веществ. | 32 | 7 | 1.18х10-38…3.40х1038 |
| Длинное веществ. | 64 | 15-16 | 2.23х10-308…1.79х10308 |
| Расшир. веществ. | 80 | 19 | 3.37х10-4932…1.18х104932 |

Кроме обычных чисел, формат **IEEE** сопроцессора предусматривает несколько специальных случаев, которые могут получаться в результате математических операций, и над которыми модно выполнять отдельные операции:

* Положительный и отрицательный ноль;
* Положительная и отрицательная бесконечность;
* Денормализованные числа;
* Неопределённость;
* Сигнальные не-числа (SNAN);
* Спокойные (тихие) не-числа (QNAN);
* Неподдерживаемые числа.

**Регистры FPU.** FPU предоставляет восемь регистров для хранения данных и 5 вспомогательных регистров.

**Регистры данных (R0-R7)** не адресуются по физическим именам, а рассматриваются в качестве стека, вершина которого называется **ST** (или **ST(0)**), и которая может находиться в любом физическом регистре. Её положение указывается полем TOP регистра SWR. Остальные регистры доступны программисту под именами **ST(1) – ST(7).** Операции могут выполняться либо между двумя операндами в вершине стека (в этом случае используется команда без операндов), либо между вершиной стека и операндом в памяти. Загрузка и выгрузка операндов всегда идет через вершину стека, при этом устанавливается новое соответствие физических и логических регистров (но программисту не надо об этом задумываться).

**Регистр SWR** – 16-битный регистр состояния.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ бита** | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **Название** | **B** | **C3** | **TOP** | | | **C2** | **C1** | **C0** | **ES** | **SF** | **PE** | **UE** | **OE** | **ZE** | **DE** | **IE** |

Здесь **B –** бит занятости сопроцессора, **C3–C0** – условные флаги, **ES** и **SF –** флаги ошибок, **PE– IE** – флаги состояния (неточный результат, антипереполнение, переполнение, деление на ноль, денормализация результата, недопустимая операция).

**Регистр CWR** –16-битный регистр управления.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ бита** | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **Название** |  | | | **1** | **RC** | | **PC** | |  |  | **PM** | **UM** | **OM** | **ZM** | **DM** | **IM** |

Здесь **RC –** управление округлением, **PC –** управление точностью, **PM – IM** – маски состояний, соответствующих битам регистра **SWR**.

**Регистр TWR** – регистр *тегов*, каждая пара бит описывает соответствующий физический регистр данных (11 – пуст, 00 – содержит число, 01 – 0, 10 – всё остальное).

**Регистр FIP** – 48-битный физический адрес последней выполненной команды;

**Регистр FDP** – 48-битный физический адрес операнда этой команды.

**Команды FPU.** FPU может выполнять команды загрузки-сохранения данных, команды базовой арифметики, команды сравнения и команды вычисления трансцендентных функций (расширенная арифметика). Независимо от типов исходных данных вычисления в регистрах сопроцессора всегда выполняются над величинами в расширенном вещественном формате. На выполнение команд влияют биты управляющего регистра **CWR**, а индикация состояний ведётся в регистре **SWR**.

**CPU** и **FPU** одновременно «видят» очередную команду выполняемого потока. Сопроцессор распознаёт «свои» команды по битам **11011** в начале кода, а прочие – пропускает. Если команда содержит операнд, то сопроцессор может обратиться к процессору за требуемыми данными. После этого **CPU** и **FPU** могут работать параллельно. Для синхронизации работы **CPU** и **FPU** может потребоваться команда **FWAIT**, которая сейчас встроена в большинство команд сопроцессора.

**Сравнения и условные переходы**. Нужно помнить, что условия команд сравнения для сопроцессора записываются не в регистр флагов, а помещаются в биты **C3,C2,C0** регистра **SWR**. С помощью команды **FSTSW AX** можно сохранить регистр **SWR** в регистре **AX.** Далее можно проанализировать командами логической обработки соответствующие биты в регистре **AX** непосредственно, организовав по ним условный переход, или же переписать регистр **AH** в младший байт регистра **FLAGS**, при этом установится следующее соответствие битов и условий:

|  |  |  |  |
| --- | --- | --- | --- |
| **УСЛОВИЕ** | **C3** | **C2** | **C0** |
| **ZF** | **PF** | **CF** |
|  | | | |
| ST(0) > источник | **0** | **0** | **0** |
| ST(0) < источник | **0** | **0** | **1** |
| ST(0) = источник | **1** | **0** | **0** |
| Несравнимы | **1** | **1** | **1** |

**Пример программы.** Ниже даётся фрагмент программы для вычисления функции **y=sin(πx2).**

Команда комментарий

…….………………………………………..……………………………………………

**finit**  ; инициализация сопроцессора.

**fld x** ; x→st(0).

**fmul x** ; st(0):=st(0)\*x, st(0)=x2.

**fldpi**  ;загрузка в st(0) константы **π,**

**;**x2 проталкивается в st(1).

**fmul**  ; перемножение данных в st(0)и st(1);

; после этого в вершине стека – **πx2**.

**fsin**  ; вычисление синуса от операнда,

;находящегося в вершине стека (st(0)).

**fstp y**  ; запись результата в память.

…………………………………………………………………………………………..

Вероятно, к этому моменту Вы достаточно уже успели привыкнуть к правилам написания программ, чтобы по данному фрагменту написать целую программу. Если же нет – то рекомендуется заново выполнить лабораторные работы 1–3.

Для выполнения работы нужно воспользоваться вариантами заданий из приведённых ниже двух таблиц. В таблицах приведены уравнения в виде . Необходимо преобразовать их к виду , где , и составить программу вычисления значения f(x).

Затем необходимо модифицировать программу таким образом, чтобы она вычисляла таблицу значений для данной функции на отрезке с постоянным шагом. Найти два таких значения аргумента, между которыми функция меняет знак.

В данной работе требуется предоставить для проверки письменный отчёт, в котором привести составленную программу и результаты её работы.

