

Лекция 2

1.1.4. Регистры состояния и управления

В микропроцессор включены несколько регистров (см. рис. 1.1.4), которые постоянно содержат информацию о состоянии как самого микропроцессора, так и программы, команды которой в данный момент загружены на конвейер. К этим регистрам относятся:

1. регистр флагов eflags/flags;
2. регистр указателя команды eip/ip.

Используя эти регистры, можно получать информацию о результатах выполнения команд и влиять на состояние самого микропроцессора. Рассмотрим подробнее назначение и содержимое этих регистров:

eflags/flags (flag register) — регистр флагов. Разрядность eflags/flags — 32/16 бит. Отдельные биты данного регистра имеют определенное функциональное назначение и называются флагами. Младшая часть этого регистра полностью аналогична регистру flags для i8086.

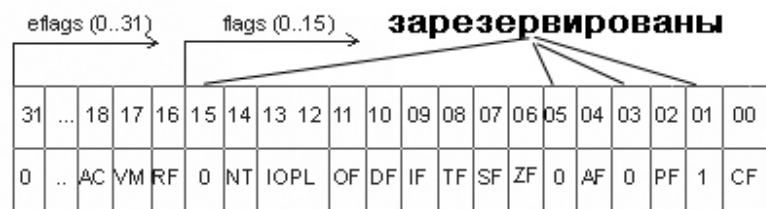


Рис. 1.1.4. Регистр флагов

Таблица

Назначение бит регистра флагов

| Мнемоника флага | Флаг | № бита в eflags | Содержание и назначение |
|-----------------|---|-----------------|--|
| CF | Флаг переноса (Carry Flag); Переполнение | 0 | 1 — указывает на переполнение старшего бита при арифметических командах. Старшим является 7, 15, 31-й бит в зависимости от размерности операнда |
| PF | Флаг чётности (Parity Flag) | 2 | 1 — 8 младших разрядов (только для 8 младших разрядов) результата содержат четное число единиц |
| AF | Флаг коррекции (Adjust Flag) | 4 | 1 - если арифметическая операция производит перенос или заём в/из 3-й бит результата, иначе - сбрасывается. Этот флаг используется для двоично-кодированной десятичной (BCD - Binary-Coded Decimal) арифметики |
| ZF | Флаг нуля (Zero Flag) | 6 | 1 — результат нулевой; 0 — результат ненулевой |
| SF | Флаг знака (Sign Flag) | 7 | Отражает состояние старшего бита результата (биты 7, 15, 31 для 8, 16, 32-разрядных операндов соот-но): 1 — старший бит результата равен 1; 0 — старший бит результата равен 0 |
| TF | Флаг ловушки (Trap Flag) | 8 | 1 - то процессор использует покомандную отладку текущей программы; 0 - программа выполняется обычным образом |
| IF | Флаг | 9 | 1 - то в ответ на IRQ процессор генерирует |

| | | | |
|-------------|--|--------|---|
| | разрешения прерываний (Interrupt enable Flag) | | прерывания; 0 - процессор не отвечает на них (но не игнорирует) |
| DF | Флаг направления (Direction Flag) | 10 | 1 - строковые команды обрабатывают строки данных, переходя от младших адресов к старшим; 0 - то в обратном направлении |
| OF | Флаг переполнения (Overflow Flag) | 11 | 1 — в результате операции происходит перенос (заем) в(из) старшего, знакового бита результата (биты 7, 15 или 31 для 8, 16 или 32-разрядных операндов соответственно); <u>Флаги состояния</u> используются командами целочисленной арифметики трёх типов. При переполнении, индикатором является: 1. для знаковой арифметики - флаг OF , 2. для беззнаковой арифметики - флаг CF , 3. для BCD-арифметики - флаг AF |
| IOPL | Уровень Привилегий ввода-вывода (Input/Output Privilege Level) | 12, 13 | Используется в защищенном режиме работы микропроцессора для контроля доступа к командам ввода-вывода в зависимости от привилегированности задачи |
| NT | Флаг вложенной задачи (Nested Task flag) | 14 | 1 - текущая задача является вызванной из предыдущей; 0 - текущая задача либо НЕ является вызванной из предыдущей |
| RF | Флаг возобновления (Resume Flag) | 16 | Управляет ответом процессора на исключение отладки. |
| VM | Флаг режима виртуального 8086 (Virtual-8086 Mode flag) | 17 | 1 - процессор переходит в режим виртуального 8086; 0 - возвращается в защищённый режим |
| AC | Флаг проверки выравнивания (Alignment Check flag) | 18 | 1- заставляет процессор проверять выравнивание при доступе к памяти и в случае невыравненного доступа генерировать исключение |
| VIF | Флаг виртуальных прерываний (Virtual Interrupt) | 19 | Это виртуальный образ флага IF, используется совместно с флагом VIP при включённом расширении режима виртуального 8086 |
| VIP | Флаг ожидания виртуального прерывания (Virtual Interrupt Pending flag) | 20 | Устанавливается, когда возникает прерывание. Процессором только считывается и используется совместно с флагом VIF; изменяется только программно |
| ID | Флаг идентификации (IDentification flag) | 21 | Если программа смогла установить и сбросить этот флаг, то это значит, что процессор может выполнить команду CPUID |

Десятичное значение 42936

Преобразование десятичного формата в шестнадцатеричный

| Частное | Остаток | Шест. | |
|------------|---------|-------|-------------------|
| 42936 / 16 | 2683 | 8 | 8 (младшая цифра) |
| 2683 / 16 | 167 | 11 | B |
| 167 / 16 | 10 | 7 | 7 |
| 10 / 16 | 0 | 10 | A (старшая цифра) |

Преобразование шестнадцатеричного формата в десятичный

| | |
|-----------------------------------|------------|
| Первая цифра: A (10) | 10 |
| Умножить на 16 | <u>*16</u> |
| | 160 |
| Прибавить следующую цифру, 7 | <u>+7</u> |
| | 167 |
| Умножить на 16 | <u>*16</u> |
| | 2672 |
| Прибавить следующую цифру, B (11) | <u>+11</u> |
| | 2683 |
| Умножить на 16 | <u>*16</u> |
| | 42928 |
| Прибавить следующую цифру, 8 | <u>+8</u> |
| Десятичное значение | 42936 |

1.2 Особенности выполнения команд

Шина МП 8086 состоит из трех шин: информационной (16 бит), адресной (20 бит) и управляющей, по которой передаются сигналы управления. Выполнение программы заключается в повторении 5-ти этапов. Эти этапы выполняются последовательно.

1. выбор служебной машинной команды из памяти;
2. расшифровка команды;
3. чтение операндов из памяти (если необходимо);
4. выполнение команды;
5. запись операндов в память (если необходимо).

Для ускорения выполнения команд они реализуются двумя устройствами: интерфейсом шины (Bus Interface Unit) и операционным блоком (Execution Unit) (рис. 1.2.1.). Первое устройство считывает команду и осуществляет передачу данных. Второй - лишь выполняет команды. Интерфейс шины может выбирать следующую команду в то время, как операционное устройство выполняет раньше выбранную.

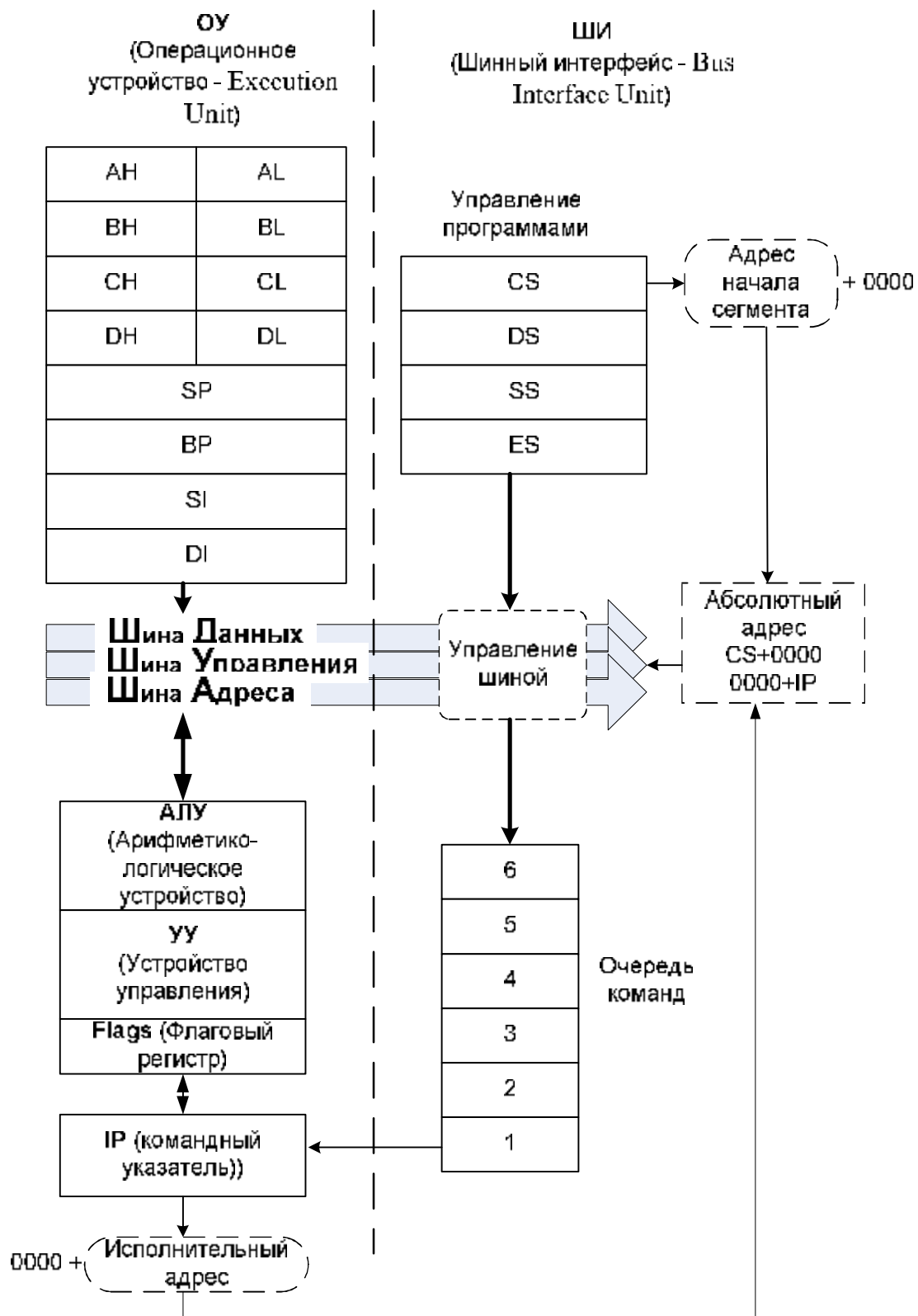


Рис. 1.2.1. Операционное устройство и шинный интерфейс.

Процессор имеет внутреннюю память, которая называется *очередью команд*. Здесь сохраняется до 4 (в 8086 - до 6) предварительно выбранных из потока команд-байтов. То есть, реализуется своеобразный конвейер команд. Как же формируется абсолютный адрес команды?

В регистре указателя команд IP записан исполнительный адрес, а в регистре сегмента кода CS - адрес начала сегмента. Абсолютный адрес равняется сумме CS+IP и записывается в 20 бит, в то время как и CS и IP имеют по 16.

Принято, что сегмент должен начинаться не с любого адреса, а кратного 16 бит. Область памяти 16 бит называется *параграфом*. Иначе говоря, сегмент выравнивается по границе параграфа. Следовательно, сегмент может начинаться лишь с адресов 16, 32, 48 . В двоичной системе это будет:

| |
|---------|
| 00000 |
| 1000000 |
| 1100000 |

Как видим, при этом последние 4 биты будут нулевыми. Для хранения они лишние и их отбрасывают. То есть, в 16-ти разрядных регистрах сегментов фактически сохраняется 20-ти разрядный адрес, но без 4 нулей справа.

При вычислении абсолютного адреса к содержанию регистра CS дописывают 4 нуля справа, а к содержанию IP - 4 нуля налево и полученные коды добавляют.

1.2.1. Реализация прерываний

Считывание символа и запись к памяти длится несколько микросекунд. Если ЦП будет лишь принимать эти символы, то большинство времени он будет простаивать. Потому, закончив обработку символа, ЦП переходит к выполнению другой программы. Каждый раз, как нажимается клавиша, устройство подает запрос на прерывание. ЦП прерывает выполнение программы и переходит к выполнению процедуры обработки для прерываний для клавиатуры. Каждая такая процедура является определенной программой, записанной в памяти.

Для того, чтобы к ней перейти, нужно знать ее начальный адрес. Этот начальный адрес и записан в так называемом *векторе прерывания*. Каждое из возможных 256 прерываний имеет свой вектор прерывания в памяти. Вектор прерываний состоит из двух слов: **CS:IP**. Записанные в начальных адресах от 0 к 03FFH (1024 байта).

Особенности 32-разрядных процессоров

В процессоре i80286 адресная шина состояла из 24 битов, что дало возможность адресовать до 16М памяти. Но это возможно сделать в т. зв. *защищенном режиме*.

В процессоре i486 32-разрядное слово, которое дает возможность адресовать $2^{32} = 4 \cdot 2^{30} = 4$ Гбайт. При сегментной организации памяти размер сегмента и будет таким. Кроме того, есть еще и страничная организация памяти. Размер страницы - 4 Кб. Такой способ позволяет использовать виртуальную память, объем которой больше физической, около 4 Тбайт.

Эти процессоры должны возможность реализовать многозначительные вычисления. Потому их структура сложнее.

Для программиста процессор состоит из 32 регистров, 16 из которых является системными, а остальные - пользователя.

Рассмотрим особенности регистров пользователя.

Регистры общего назначения: 32-разрядные. Имя 32-разрядного начинается буквой E (extended): **EAX, EBX, ECX, EDX**. Младшая половина просто **AX**. Младшая половина доступна и может делиться пополам. Старшая половина отдельно недоступна.

Регистры-указатели - такие же имена: **ESP, EBP, ESI, EDI**. Младшая половина - **SP**.

Сегментные регистры - 16-разрядны: **CS, SS, DS, ES** и есть еще 2 дополнительных -- **GS, FS**. То есть, 6 регистров.

Регистры управления: 32-разрядные -- **EFLAGS** и **EIP**, а младшая часть - **FLAGS** и **IP**.

