**Процессоры и их регистры**

**Процессор состоит:**

**1. Контроллер** управляет получением инструкций из памяти и их декодированием. Контроллер не обрабатывает инструкцию: после декодирования он просто передает ее по внутренней шине управления к другим модулям, которые выполняют необходимое действие.

**2. Арифметико-логическое устройство** (АЛУ) выполняет арифметические и логические действия над данными.

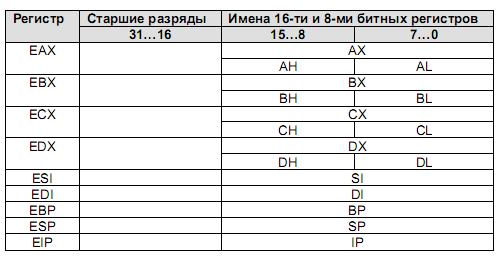
**3. Регистры** участки высокоскоростной памяти предназначенные для оперативного хранения данных и быстрого доступа к ним со стороны внутренних компонентов процессора.

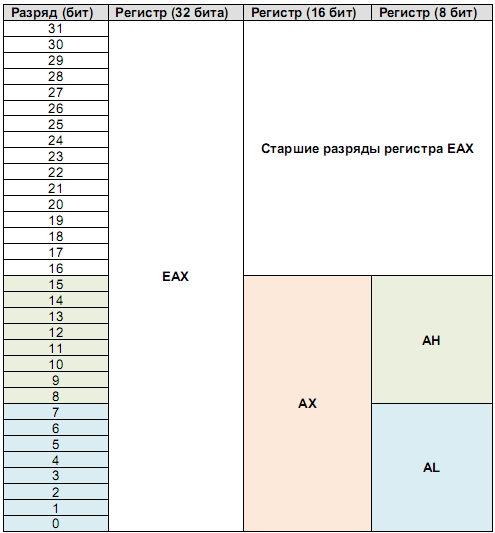
* Регистры общего назначения - содержат рабочие данные, полученные из памяти.
* Регистры состояния - содержат текущее состояние процессора (или состояние АЛУ).
* Счетчики - счетчик команд, содержащий адрес следующей инструкции.

**Основные регистры процессора**



**«Делимые» регистры**





**Регистры общего назначения**

Они называются:

1. ЕАХ (Аккумулятор),
2. ЕВХ (База),
3. ЕСХ (Счетчик),
4. EDX (Данные).

Кроме названий, они больше ничем другим не отличаются друг от друга, поэтому рассмотрим только первый регистр — ЕАХ.



Регистр ЕАХ разделен на две части — 16-разрядный регистр АХ и верхние 16 битов, которые никак не называются. В свою очередь, регистр АХ разделен на два 8-битных регистра — АН и AL.

* **EAX, EBX, ECX, EDX** - 32-разрядные регистры, которые могут хранить числа от 0 до 4 294 967 295 (0FFFFFFFFh).
* **AX, BX, CX, DX** - 16-разрядные регистры, которые могут хранить числа от 0 до 65 535 или от 0h до FFFFh в шестнадцатеричной системе.
* **AH, AL, BH, BL, CH, CL, DH, DL** - 8-разрядные регистры, которые могут хранить числа от 0 до 255 (FFh).

Если мы заносим в регистр **ЕАХ** значение **0x12345678**, то регистр **АХ** будет содержать значение **0x5678** (**0x56** в **АН** и **0x78** в **AL**), а значение **0x1234** будет помещено в **верхнюю часть** регистра ЕАХ.

**Регистр флагов**

**Регистр флагов** (регистр признаков) — EFLAGS. Он состоит из одноразрядных флагов, отображающих в основном текущее состояние арифметико-логического устройства.



В наших программах мы будем использовать все 32 флага, а пока рассмотрим только самые важные из них:

• **Признак** нуля ZF (Zero Flag) — 1, если результат предыдущей операции равен нулю.

**• Признак знака** SF (Sign Flag) - 1, если результат предыдущей операции отрицательный.

**• Признак переполнения** OF (Overflow Flag) — 1, если при выполнении предыдущей операции произошло переполнение (overflow), то есть результат операции больше, чем зарезервированная для него память.

**• Признак переноса** CF (Carry Flag) — 1, если бит был «перенесен» и стал битом более высокого порядка.

**• Признак прерывания** IF (Interrupt Flag) — 1, если прерывания процессора разрешены.

**• Признак направления** DF (Direction Flag) — используется для обработки строк.

**Системы счисления**

**1.** Запись шестнадцатеричного числа 123416:

* **0x**1234,
* **$0**1234,
* 01234**h**,
* Если первая символ A-F, то перед числом нужно написать 0: **0**DEh.

**2.** Запись десятичного числа 123410:

* 1234,
* 1234**d**.

**3.** Запись двоичного числа 11002:

* 1100**b**

**4.** Запись восьмеричного числа 128:

* 12**q**

**Типы данных**

|  |  |  |
| --- | --- | --- |
| **Тип числа** | **Диапазон значений** | **Степени двойки** |
| 1 Байт = 8 бит | 0…255 | 0…(28 – 1) |
| 1 Слово **(word)** = 2 байта = 16 бит | 0…65 535 | 0…(216 – 1) |
| 1 Двойное слово **(dword)** = 4 байта *=* 32 бит | 0…4 294 967 295 | 0…(232 – 1) |

**Основные команды языка ассемблера**

Данные, которые обрабатываются командами, называются **операндами.**

**1. Команда MOV**

Команда **MOV** – копирует значение из источника в приемник:

**MOV приемник, источник**

Оба операнда команды MOV должны быть одного размера:

mov ах, bl ;**НЕПРАВИЛЬНО!** - Операнды разных

;размеров.

**2. Арифметические команды**

**2.1 Инструкции сложения ADD и вычитания SUB**

ADD о1, о2 ; o1=o1+o2

Команда ADD складывает оба операнда и записывает результат в о1, предыдущее значение которого теряется.

SUB o1, о2 ; o1=o1-o2

Результат, о1-о2, будет сохранен в о1, исходное значение о1 будет потеряно.

**2.2 Команды инкрементирования INC и декрементирования DEC**

Команда **INC** добавляет, а **DEC** вычитает единицу из единственного операнда.

INC о1 ; о1 = о1 + 1

DEC о1 ;о1 = о1 - 1

**Задача №1.** Требуется вычислить выражение: Y=a+b-c. Пусть a, b, c, Y - целые числа: а=1516, b=1216, c=716.

**Решение:**

***Примечание.*** *В языке ассемблера точка с запятой является символом начала комментария.*

|  |  |  |
| --- | --- | --- |
| Mov ax, 15h | ; заносим в АХ число 1516 |  |
| Mov bx, 12h | ; заносим в BХ число 1216 |  |
| Add ax, bx | ; AX=AX+BX | 1516+1216=2716 |
| Mov cx, 7h | ; заносим в CХ число 716 |  |
| Sub ax, cx | ; AX=AX-CX | 2716-716=2016 |

**2.3 Целочисленное умножение MUL и деление DIV**

Операции умножения и деления имеют свою специфику. В результате умножения двух чисел мы можем получить число, диапазон которого будет в два раза превышать диапазон операндов. Деление целых чисел — это операция целочисленная, поэтому в результате образуются два значения: частное и остаток.

Эти команды спроектированы так, что один из операндов и результат находятся в фиксированном регистре, а второй операнд указывается программистом.

**1. Команда MUL** может быть записана в трех различных форматах — в зависимости от операнда:

В **8-разрядной форме** операнд может быть любым 8-битным регистром или адресом памяти. Второй операнд всегда хранится в AL. Результат (произведение) будет записан в регистр АХ.

В **16-разрядной форме** операнд может быть любым 16-битным регистром или адресом памяти. Второй операнд всегда хранится в АХ. Результат сохраняется в паре DX:AX.

В **32-разрядной форме** второй операнд находится в регистре ЕАХ, а результат записывается в пару EDX:EAX.

**2. Команда DIV** может быть представлена в трех различных форматах в зависимости от типа операнда.

Операнд служит делителем, а делимое находится в фиксированном месте.

В **8-битной форме** переменный операнд (делитель) может быть любым 8-битным регистром или адресом памяти. Делимое содержится в АХ. Результат сохраняется так: частное — в AL, остаток — в АН.

В **16-битной форме** операнд может быть любым 16-битным регистром или адресом памяти. Второй операнд всегда находится в паре DX:AX. Результат сохраняется в паре DX:AX (DX — остаток, АХ — частное).

В **32-разрядной форме** делимое находится в паре EDX:EAX, а результат записывается в пару EDX:ЕАХ (частное в ЕАХ, остаток в EDX).

**Задача №2.** Требуется вычислить 4862 результат сохранить в DX:AX.

**Решение:**

***Примечание:*** В контексте арифметических операций очень часто используются так называемые пары регистров. Пара — это два регистра, использующихся для хранения одного числа. Часто используется пара EDX:EAX (или DХ:АХ) — обычно при умножении. Регистр АХ хранит младшие 16 битов числа, а DX — старшие 16 битов.

|  |  |  |
| --- | --- | --- |
| Mov ах, 486 | ; АХ = 486 | 48610=01E616 |
| Mul ах | ; DX:AX=АХ\*АХ | 48610\*48610=23619610=39AA416  Результат помещается в пару регистров **DX**:**AX** – **0003** **9AA4** |

**Задача №3.** Разделить 13 на 2, частное сохранить в BL, а остаток в — ВН.

**Решение:**

|  |  |  |
| --- | --- | --- |
| Mov ах, 13 | ; АХ = 13 | 1310=D16 |
| Mov сl, 2 | ; CL = 2 | 210=216 |
| Div сl | ; AX/CL  ; Результат сохраняется так: частное — в AL, остаток — в АН. | 13/2=6,5  Ищем остаток:  6\*2=12  13-12=1  **AL**=610=**06**16  **AH**=110=**01**16 |
| Mov bx, ах | ; ожидаемый результат находится в АХ,  ; копируем в ВХ |  |

**Задача №4.** Требуется вычислить выражение: . Пусть a, b, c, Y - целые числа: а=01012, b=00112, c=00102.

**Решение:**

|  |  |  |
| --- | --- | --- |
| Mov ah, 0011b | ; заносим в АH число 00112 (b) | 310=0316 |
| Mov al, 0010b | ; заносим в AL число 00102 (c) | 210=0216 |
| Mul ah | ; AX=AH\*AL | 610=000616 |
| Mov bx, 0101b | ; заносим в BX число 01012 (a) | 510=000516 |
| Add ax, bx | ; AX=AX+BX | 610+510=1110=000B16 |
| Mul ax | ; **DX**:**AX**=AX\*AX | 1110\*1110=12110=**0000** **0079**16 |
| Add ax, 1010b | ; AX = AX +10102 (10102=1010) | 12110+1010=13110=008316 |
| Mov bx, 0011b | ; заносим в BX число 00112 (00112=310) |  |
| Div bx | ; DX:AX=AX/BX  ; DX — остаток, АХ — частное | 13110/310=43,710  Ищем остаток:  131/3=**43**,7  43\*3=129  131-129=**2**  **AX**=4310=002B16  **DX**=210=000216 |

**Для самостоятельного выполнения**

**[Пример №1.](#пр1)** Вычислить выражение: F=458 – (478 + X + Y) + Z — 368

**[Пример №2.](#пр2)** Вычислить выражение: F=(8116 – 5816) + X / (Y \* 616)

**Пример №3.** Вычислить выражение: F=778 + (248 + X - Y) - Z + 478

**Пример №4.** Вычислить выражение: F=1216 - (5616 - X + Y) + Z – 2316

**Пример №5.** Вычислить выражение: F=y2-4y+1

**Пример №6.** Вычислить выражение:

**Пример №7.** Вычислить выражение: