

# Система команд универсальных микропроцессоров



# Упрощенная структурная схема МП 8086

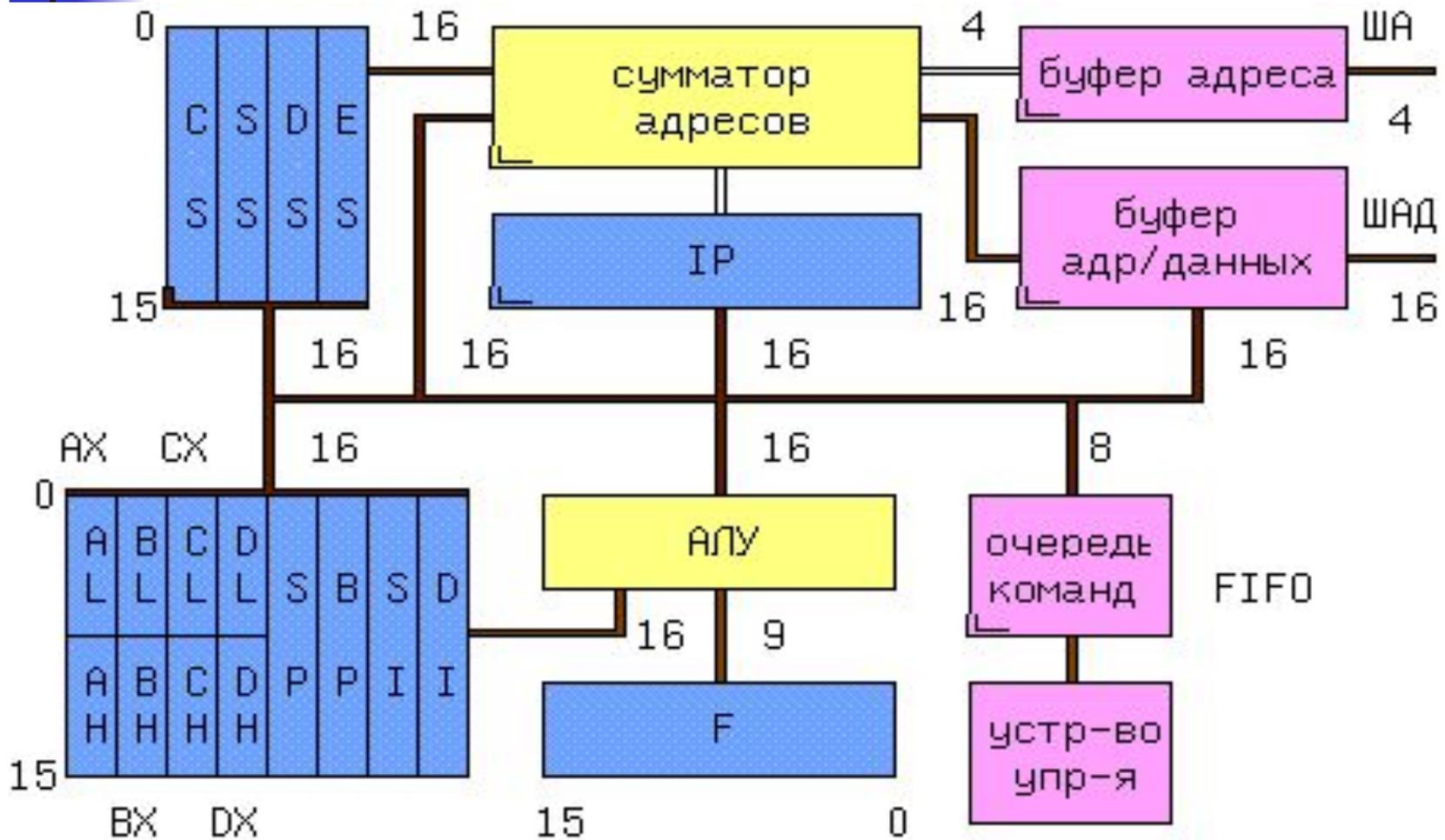
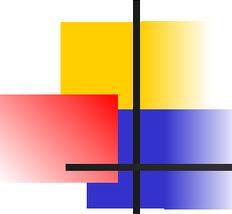


Рис.1



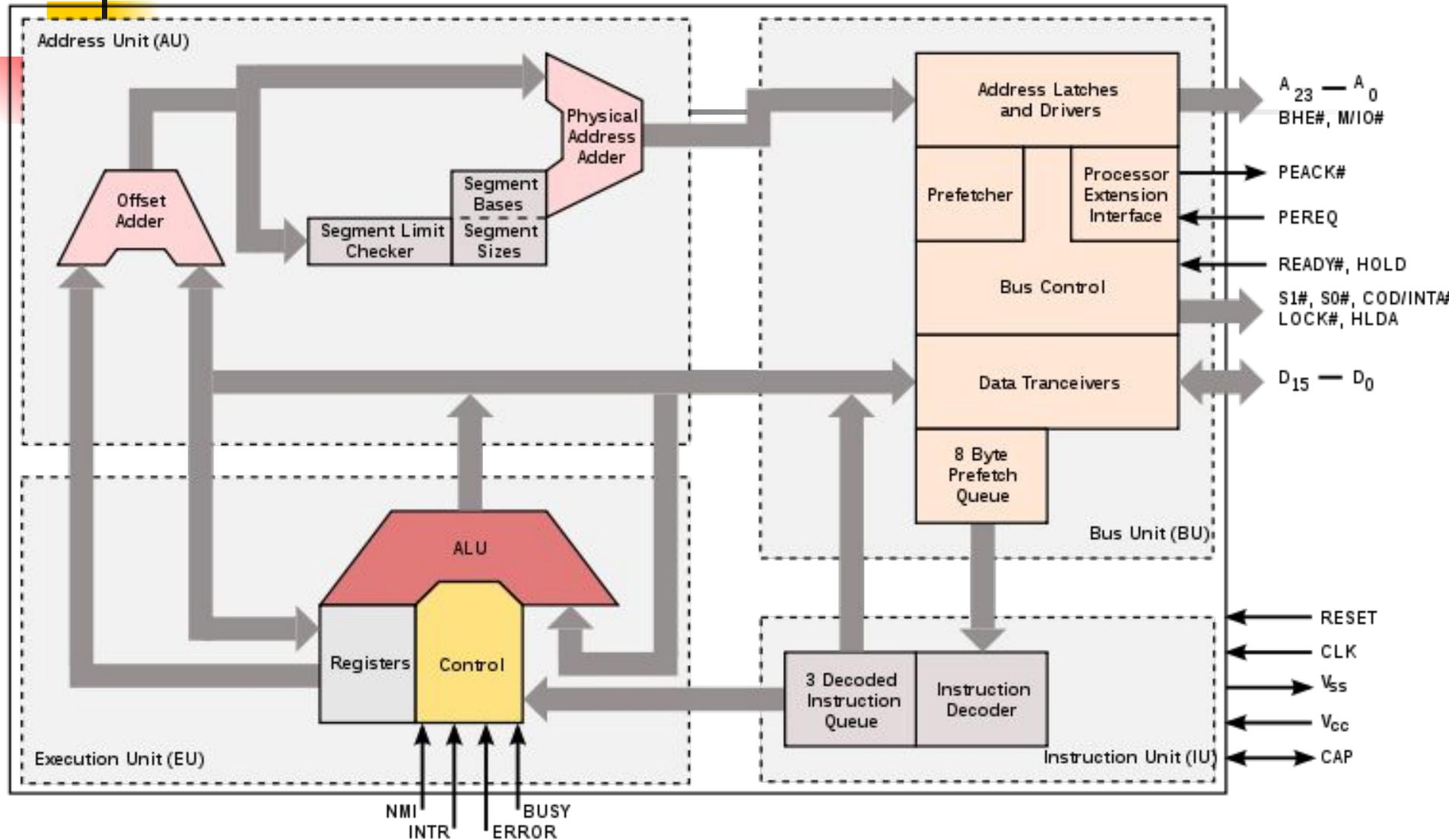
# Структурная схема 8086

---

- Устройство управления декодирует байты из очереди команд и управляет работой **операционного устройства и шинного интерфейса**. Операционное устройство МП состоит из 4-х шестнадцатиразрядных регистров **общего назначения: РОН** (АХ, ВХ, СХ, ДХ), из 4-х регистров указателей (адресных регистров SP, BP, SI, DI) и **арифметико-логического устройства (АЛУ) с регистром признаков операций (флагов F)**.
- Для сравнения ниже представлена стр схема МП 80286.

# 80286 МП

Intel 80286 architecture



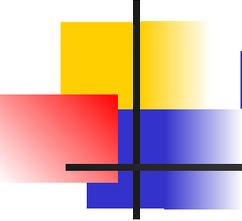
# 1.3 Система команд универсальных микропроцессоров

- Система команд любого процессора характеризуется
- форматами команд(рассмотрим кратко),
- группами команд (знать обязательно!)
- и видами адресации.
- Форматы команд зависят от архитектуры МП, вида адресации, а также от разрядности данных или адреса конкретного МП.

# Фирмы, выпускающие МП



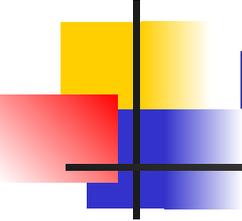
- Complex Instruction Set Computers означает - устройство вычислений с полной системой команд, которые реализуют на уровне машинного языка наборы команд разной сложности; от простых, что характерно для МП первого поколения, к значительной сложности, характерным для современных 32 -разрядных микропроцессоров типа Pentium, 680xx и др. Мы будем изучать процессоры в основном с Cisc-архитектурой.



# Форматы команд универсальных микропроцессоров

---

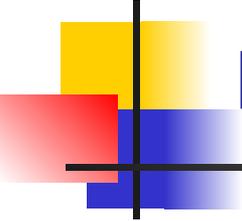
- В МП с RISC-архитектурой различных форматов команд немного. В идеале ВСЕ команды в таких МП имеют одинаковый формат. Например – 3 или 4 байта. В реальности даже в МП RISC-архитектурой используется 2-3 и более форматов.
- В МП с CISC-архитектурой используются самые разнообразные форматы команд. Знать их все не обязательно, но полезно понимать, от ЧЕГО они зависят.



# Форматы команд универсальных микропроцессоров

---

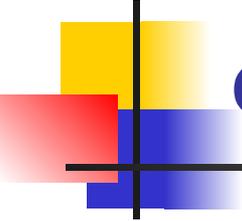
- Как пример системы команд универсальных МП, рассмотрим систему команд процессора 8086. Она содержит 113 мнемоник, и допускает использование разнообразных режимов адресации. *Формат команды зависит операндов и вида адресации.*
- **Команды имеют длину от 2 до 6 байтов.**
- Каждая команда состоит из кода операции (КОП), идентифицирующего ее, и, если нужно, одного или двух операндов. Чем больше операндов и чем они длиннее, тем больше байтов занимает команда и **тем больше времени нужно для передачи ее из памяти в процессор.**



# Форматы команд универсальных микропроцессоров

---

- Всего же в системе команд процессора 8086 насчитывается более 3800 команд вместе со всеми их вариантами, а в системе команд МП 80386 – вдвое больше.
- Чтобы минимизировать общее число бит в команде, команды процессора 8086 имеют **не более двух операндов** (mov AX, BX) причем одним обязательно является регистр . Ограничение двумя операндами, естественно, уменьшает гибкость многих операций.
- Некоторые команды (например INC и DEC) имеют **один операнд**.



# Перевод чисел из двоичной системы в шестнадцатеричную

---

При программировании на ассемблере мы будем в основном пользоваться 16-ричной системой. Немного напомню принцип перевода.

- Пример (перевод 2-16 без использования 10-ной системы):
- **$10110100011_2 = 0101\ 1010\ 0011 = 5A3_{16}$**
- При переводе каждой тетрады нужно просто пользоваться приведенной ниже таблицей двоично-16-ричного соответствия.

# Виды адресации универсальных микропроцессоров

- **Мощность системы команд процессора зависит в большой мере от количества видов адресации, которые он использует.**  
Система команд МП с CISC-архитектурой насчитывает не менее чем десять видов адресации.
- **1. Неявная** – когда в коде команды никак не указанные операнды.
  - *Примеры:* DAA, STOSW
- **2. Прямая регистровая** – один или два операнда находятся в регистрах.
  - *Примеры:* INC di, DEC cx, ADD ax, bx

# Виды адресации универсальных микропроцессоров

**3. Непосредственная** – один из операндов в регистре, второй – непосредственно после кода операции.

- *Примеры: ADD di, 160; AND ax, C5H*

**4. Прямая к памяти или портам.** Один из операндов в регистре, адрес второго прямо в команде.

- *Примеры: MOV ax, alfa;*
- *IN ax, 40H; IN al, 40h.*

# Виды адресации универсальных микропроцессоров

**5. Косвенная** – существует несколько вариантов.

Например к памяти и к портам.

- Примеры: IN ax, dx.

**6. Индексная** MOV ax, [si], ADD ax, [di+2]

**7. Базовая** MOV ax, [bp] ADD ax, [bx+2]

- Базовая и индексная адресации могут быть со сдвигом, то есть адрес ячейки памяти в сегменте) вычисляется как сумма регистра адреса и сдвига.
- Формирование физического адреса (20- или 24-разрядной) из логической (32-разрядной) – функция сумматора адресов.

# Виды адресации универсальных микропроцессоров

**8. Базово-индексная** Используется для обработки двухмерных массивов.

- *Пример:* **MOV cx, [si+bx+2]**

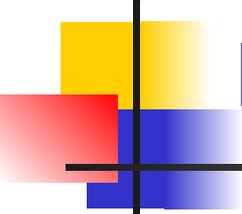
**9. Стековая - с использованием указателя стека SP** (см. слайд 15).

- *Примеры:* PUSH cx, Call delay, RET data.

**10. Короткая или длинная адресация в командах перехода** (см. слайд 30).

- Существуют к тому же сочетания разных видов адресации, например, косвенной и непосредственной

# Группы команд универсальных микропроцессоров

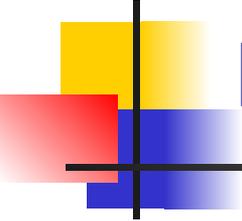


---

По функциональному назначению в МП 8086 выделяют 6 групп команд:

- 1. Команды передачи данных.
- 2. Команды арифметических операций.
- 3. Команды логических операций и сдвигов.
- 4. Команды передачи управления.
- 5. Цепочечные команды.
- 6. Команды управления микропроцессором.

# Группы команд универсальных микропроцессоров



---

- **Первая группа команд - команды передачи данных**

Команды передачи данных разделяют на 3 подгруппы:

- общие команды передачи данных;
- стековые команды;
- команды ввода-вывода;

*Команды передачи данных как правило не модифицируют состояния регистра флагов.*

# Группы команд универсальных микропроцессоров

- **Самой распространенной среди команд передачи данных является команда MOV.**
- Она имеет следующей обобщенной вид:
- MOV dst, src
- Первым указывается операнд-получатель, а вторым – операнд-источник.
- Одним из операндов обязательно должен быть регистр общего назначения.
- *Команда осуществляет передачу: регистр - регистр, регистр - память, память - регистр, непосредственные данные - регистр, непосредственные данные - память, регистр - сегментный регистр, память - сегментный регистр, сегментный регистр - регистр, сегментный регистр - память.*

# Группы команд универсальных микропроцессоров

- Являются допустимыми следующие форматы команды:
- MOV mem/reg1, mem/reg2
- MOV mem/reg, data
- MOV sreg, mem/reg
- MOV mem/reg, sreg
- ***Примеры***
- MOV AX,DX
- MOV AX,0B800H
- MOV AX,ES:[160]

# Группы команд универсальных микропроцессоров

- В отличие от ОБЩИХ команд передачи данных команды работы со стеком – парные.
- **Каждая команда занесения данных в стек PUSH имеет соответствующую ей команду извлечения из стека POP.**
- Для адресации вершины стека используется регистры SS:SP.
- *Все стековые команды манипулируют только словами и сопровождаются автоматической модификацией указателя стека: при занесении в стек производится декремент, а при извлечении из стека - инкремент указателя стека SP.*
- *До выполнения стековых команд регистры SP и SS должны быть инициализированы.*
- *Команды PUSHF и POPF предназначены для временного запоминания в стеке и последующего возобновления из стека содержания регистра флагов.*

# Группы команд универсальных микропроцессоров

---

- Формат команд:
- PUSH mem/reg    POP mem/reg
- PUSH sreg    POP sreg
- PUSHF    POPF
- ***Примеры:***
- PUSH CX
- ..... Фрагмент программы
- POP CX

# Группы команд универсальных микропроцессоров

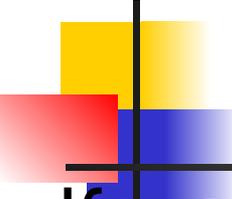
- **Команда ввода IN и команда вывода OUT.**
- Команда IN загружает данные из заданного порта в аккумулятор(Ax или AL), а команда OUT выполняет передачу из аккумулятора в порт. Для портов ввода-вывода в диапазоне 00-FF можно использовать прямую укороченную адресацию, а остальные порты в диапазоне 100-FFFF можно адресовать только косвенно через регистр DX.
- ***Формат команд:***
- IN ac,port    OUT port,ac    (прямая укороченная адресация)
- IN ac,DX    OUT DX,ac    (косвенная адресация)

# Группы команд универсальных микропроцессоров

- **Вторая группа - арифметические команды**
- *Арифметические операции выполняются над целыми числами четырех типов: беззнаковыми двоичными, знаковыми двоичными, упакованными десятичными и не упакованными десятичными. Длина чисел может быть 8 или 16 бит.*
- *Диапазоны чисел: беззнаковое 8-битовое: 0 - 255, беззнаковое 16-битовое: 0 - 65535, знаковое 8-битовое: -128 - +127, знаковое 16-битовое: -32768 - +32767.*
- Команды арифметических операций влияют на флаги OF, SF, ZF, AF, PF и CF.

# Группы команд универсальных микропроцессоров

- Команда ADD позволяет производить сложение 8- или 16-битовых двоичных чисел в режиме регистр-регистр, регистр-память и память-регистр.
- Общее представление команды имеет вид
- ADD dst, src
- При этом первый операнд складывается со вторым и результат операции замещает первый операнд. Формат команды:
- ADD mem/reg1,mem/reg2 (*Например* **ADD AX,BX**).
- ADD mem/reg,data (*Например* **ADD DI,160**)



# Арифметические команды

---

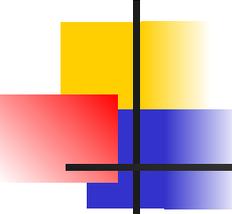
Команда SUB позволяет производить вычитание 8- или 16-битовых двоичных чисел. Общее представление команде имеет вид

- SUB dst, src

второй операнд отнимается из первого и результат операции замещает первый операнд.

## ***Формат команды:***

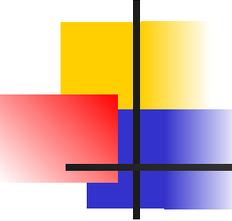
- SUB mem/reg1,mem/reg2
- SUB mem/reg,data (***Например*** SUB DI, 160)



# Арифметические команды

---

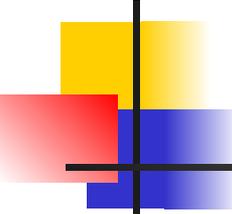
- Команда ADC выполняет сложение с переносом: в отличие от команды ADD в операции принимает участие флаг CF, значение которого добавляется к младшему биту результата. **Формат команды:**
  - ADC mem/reg1,mem/reg2
  - ADC mem/reg,data
- Команда SBB выполняет вычитание с перенесением: в операции вычитание принимает участие флаг CF, значение которого отнимается из младшего бита результата вычитания операндов. **Формат команды:**
  - SBB mem/reg1,mem/reg2
  - SBB mem/reg,data



# Арифметические команды

---

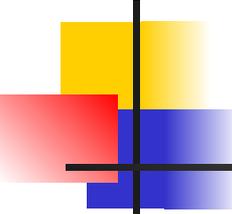
- Команда INC позволяет увеличить на 1 содержание какого-нибудь общего регистра или ячейки памяти.
- **Формат команды:**
- *INC mem/reg, (Например **INC ax**, **INC [DX]**)*
- Команда DEC позволяет уменьшить на 1 содержание какого-нибудь общего регистра или ячейки памяти. **Формат команды:**
- *DEC mem/reg (Например **DEC CX** )*



# Арифметические команды

---

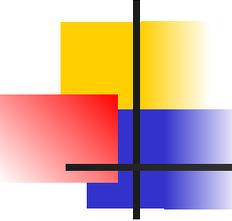
- Команда NEG изменяет знак числа, что находится в регистре или ячейке памяти, на противоположный. Формат команды:
- NEG mem/reg
- Команда сравнения CMP выполняет вычитание второго операнда из первого, но нигде не запоминает результат операции и влияет только на состояние флажков.
- Формат команды:
- CMP mem/reg1,mem/reg2
- CMP mem/reg,data
- *Команда десятичной коррекции для составления DAA выполняет коррекцию суммы, полученной в результате выполнения команд ADD и ADC и должна следовать непосредственно за ними. Команда десятичной коррекции для вычитания DAS выполняет коррекцию разницы, полученной в результате выполнения команд SUB и SBB и должна следовать непосредственно за ними. Форматы команд: DAA, DAS*



# Арифметические команды

---

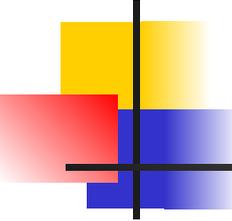
- Микропроцессор 8086 имеет две команды умножения: для беззнаковых и для знаковых двоичных чисел. Умножение десятичных чисел требует использования команд коррекции.
- Команда умножения беззнаковых целых чисел MUL выполняет умножение операнда, на содержание аккумулятора.
- При операции над байтами функции аккумулятора выполняет регистр AL, а 16-битовый результат операции помещается в регистр AX. При операции над словами функции аккумулятора выполняет регистр AX, а результат длиной 32 бита формируется в регистрах DX (старшее слово) и AX (младшее слово).
- Формат команды: MUL reg, MUL mem



# Арифметические команды

---

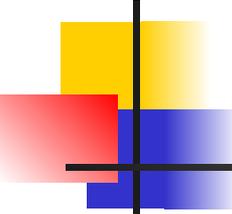
- Команда IMUL аналогичная команде MUL, но сомножители и произведение интерпретируются как знаковые двоичные числа в дополнительном коде.
- Формат команды: IMUL reg, IMUL mem
- Микропроцессор 8086 имеет две команды распределения: для беззнаковых и для знаковых двоичных чисел. Распределение десятичных чисел также требует использования специальных команд коррекции.
- Команда распределения беззнаковых чисел DIV производит распределение содержания аккумулятора и его расширения на содержание операнда, что адресуется.
- При распределении 16-битового делимого на 8-битовый делитель делимое помещают в регистр AX. В результате выполнения операции частное формируется в регистре AL, а остаток - в AH.
- При распределении 32-битового делимого на 16-битовый делитель старшая часть делимого помещается в регистр DX, а младшая - в AX. В результате выполнения операции частное формируется в регистре AX, а остаток - в DX.



# Логические команды

---

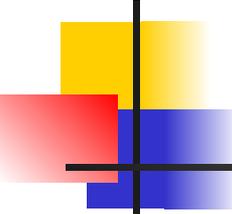
- **3 группа - команды логических операций и сдвигов**
- Логические операции представлены командами NOT (инверсия), AND (конъюнкция), OR (дизъюнкция), XOR (исключающее ИЛИ) и командой TEST, которая выполняет логическое умножение операндов, но не изменяет их значений. Все логические операции являются поразрядными, то есть выполняются независимо для каждого бита.



# Логические команды

---

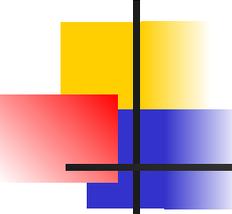
- Бинарные команды AND, OR, XOR и TEST влияют на флажки OF, SF, ZF, PF и CF. Унарная операция NOT не влияет на состояние флажков.
- Форматы команд:
- AND mem/reg1,mem/reg2
- AND mem/reg,data
- OR mem/reg1,mem/reg2
- OR mem/reg,data
- XOR mem/reg1,mem/reg2
- XOR mem/reg,data
- TEST mem/reg1,mem/reg2
- TEST mem/reg,data
- NOT mem/reg



# Команды сдвигов

---

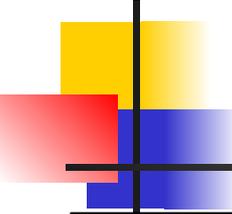
- Команды сдвигов подразделяются на команды простых сдвигов и команды циклических СДВИГОВ. Циклические сдвиги влияют только на флаги OF и CF, а простые изменяют пять флажков: OF, SF, ZF, PF и CF.
- Команды сдвигов могут работать как с байтами, так и со словами.
- Команды ROL и ROR реализуют простой циклической сдвиг влево и вправо соответственно, помещая значение из бита, что выдвигается, в бит, что освобождается.
- Команды RCL и RCR называются командами циклического сдвига влево и вправо через перенос, поскольку флажок CF расширяет сдвигаемый операнд на один бит. Таким образом, значение с CF загружается в бит, который освобождается, а выдвигаемый бит помещается в CF.



# Команды сдвигов

---

- Команды SHL(SAL) и SHR и SAR реализуют арифметические сдвиги влево и вправо соответственно на произвольное число разрядов. Для них характерно, что в бит, что освобождается, загружается ноль, а бит, что выдвигается, теряется. Появляются с МП 80286.
- Арифметическое сдвиг вправо отличается от логического сдвига тем, что знаковый бит не сдвигается, а дублируется в соседнем правом бите, сохраняя тем самым знак числа. Арифметический сдвиг влево эквивалентен логическому, поэтому мнемоника SAL и SHL помечает одну и ту же машинную команду. Команды арифметического сдвига в сущности реализуют умножение и деление чисел на степень числа 2.



# Вопросы для самоконтроля

---

1. Особенности форматов команд МП 8086
2. Перечислите группы команд МП 8086.
3. Приведите примеры использования команд из первых трех групп.
4. Перечислите виды адресации МП 8086.
5. Приведите примеры использования команд с прямой регистровой и прямой к портам адресацией.
6. Приведите примеры использования команд с непосредственной и неявной адресацией