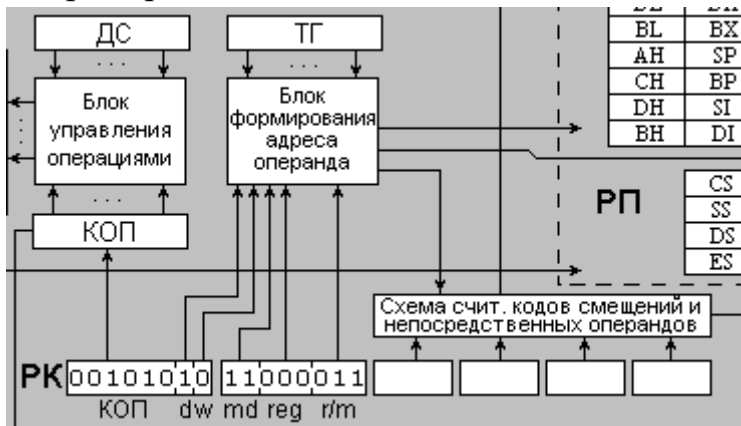


Сначала опишем регистр команд для разных случаев для общего понимания работы. Например, есть команда **sub al, bl**:



Этот формат называется **R/M, R** (регистр / память, регистр). Используется при сложении/вычитании двух регистров либо регистра и памяти. Общий формат:

R/M, R    КОП   d   w    md   reg   r/m

В регистре команд первые 6 разрядов относятся к КОП – это код операции. Определяет, какая операция совершается.

7 разряд d – определяет источник и приёмник. d=0 – источник в reg, приёмник в md, r/m; d=1 – источник в md, r/m, приёмник в reg.

8 разряд w – указывает на длину операнда.

md, r/m – задают режим адресации (см. таблицу 1).

reg – вместе с w определяют регистр.

disp – смещение адреса операнда.

**Таблица 1. Формирование 16-разрядного EA**

r/m	md			
	00	01	10	11
000	ds:[bx+si]	ds:[bx+si+disp8]	ds:[bx+si+disp16]	Смотри таблицу регистров ниже.
001	ds:[bx+di]	ds:[bx+di+disp8]	ds:[bx+di+disp16]	
010	ss:[bp+si]	ss:[bp+si+disp8]	ss:[bp+si+disp16]	
011	ss:[bp+di]	ss:[bp+di+disp8]	ss:[bp+di+disp16]	
100	ds:[si]	ds:[si+disp8]	ds:[si+disp16]	
101	ds:[di]	ds:[di+disp8]	ds:[di+disp16]	
110	ds:[disp16]	ss:[bp+disp8]	ss:[bp+disp16]	
111	ds:[bx]	ss:[bx+disp8]	ss:[bx+disp16]	

При md≠11 операндами являются регистр и память. При md=11 используются два регистра. У нас md=11, поэтому смотрим **таблицу регистров**:

reg,r/m при md=11	w	
	0	1
000	al	ax
001	cl	cx
010	dl	dx
011	bl	bx
100	ah	sp
101	ch	bp
110	dh	si
111	bh	di

Поскольку у нас d = 1, то значит, что источник находится в r/m, приёмник результата в reg.

Reg = 000, w = 0, регистр al.

r/m = 011, w = 0, регистр bl.

У нас в команде эти два регистра есть, и мы их таким образом определили.

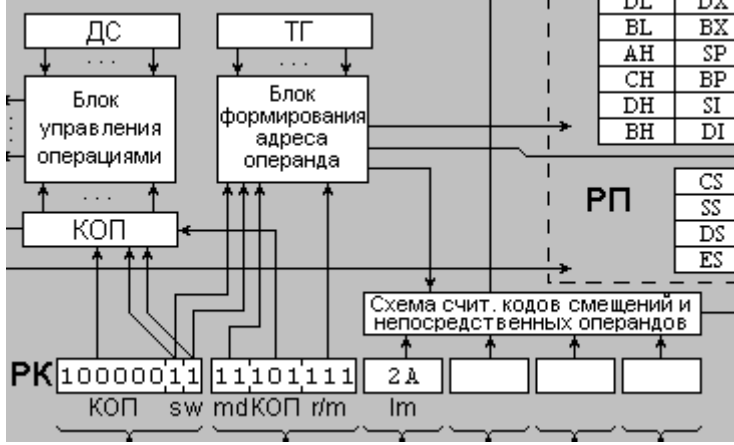
Если бы d было равно нулю, то приёмник находился бы в r/m, а источник – в reg.

**Примеры команд R/M, R:**

- add AX,BX
- add AL, CH
- add [BX+0A3h], CX
- add CX, [BX+0A3h]
- add [si], si
- sub bh, [bp+0DFF7h]

**Примечание:** тип операции (sub или add) не влияет на формат команды.

**Рассмотрим другую команду: sub di, 2Ah**



Этот формат называется **R/M, I** (регистр / память, непосредственный операнд). Используется при сложении/вычитании регистра либо памяти с числом (непосредственным операндом). Общий формат:

R/M, I	КОП	s	w	md	КОП	r/m	Im8 (ImL)	Im16 (ImH)
--------	-----	---	---	----	-----	-----	-----------	------------

В регистре команд первые 6 разрядов относятся к КОП – это код операции. Определяет, какая операция совершается.

7 разряд s – вместе с битом w определяют разрядность операнда и операции (таблица 2).

8 разряд w – разрядность операнда.

md – определяет первый операнд (см. таблицу 1). При md=11 первым операндом является регистр. Его разрядность определяется полем w. Разрядность непосредственного операнда определяется полями s, w (табл. 2 ниже).

При md≠11 первым операндом является память. Разрядность обоих операндов определяется в этом случае битами s, w.

**Таблица 2. Разрядность операции и операндов**

s	w	разрядность	
		операция	операнд
0	0	8	8
0	1	16	16
1	0	---	---
1	1	16	8

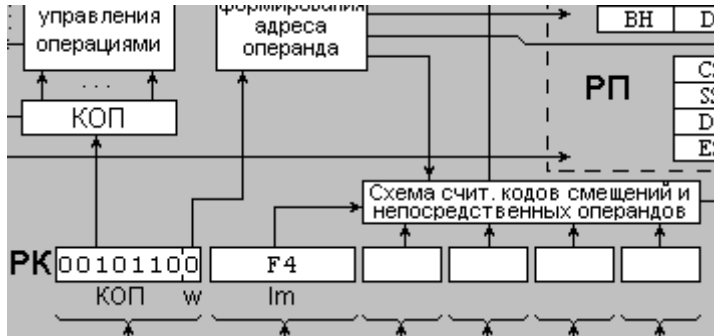
r/m вместе с md определяет формирование 16-разрядного ЕА (таблица 1).

Im – непосредственный операнд. Если операнд – не байт, а слово, то будет две ячейки: ImL и ImH (верхний и нижний индекс).

### Примеры команд R/M, I:

- add BX, 05h
- add CX, 50h
- add [BX], 50h
- add [BX+0A3h], 0AABh

Рассмотрим третью команду: **sub ax, 0F4h**



Этот формат называется **A, I** (аккумулятор AX, непосредственный операнд). Используется при сложении/вычитании регистра AX с числом (непосредственным операндом). Общий формат:

A, I 

КОП		w
-----	--	---

Im8 (ImL)	Im16 (ImH)
-----------	------------

В регистре команд первые 6 разрядов относятся к КОП – это код операции. Определяет, какая операция совершается.

7 разряд пустой.

8 разряд w – разрядность числа. При w = 1 операция проводится над словами, а при w = 0 – над байтами.

Im – непосредственный операнд. Если операнд – не байт, а слово, то будет две ячейки: ImL и ImH (верхний и нижний индекс).

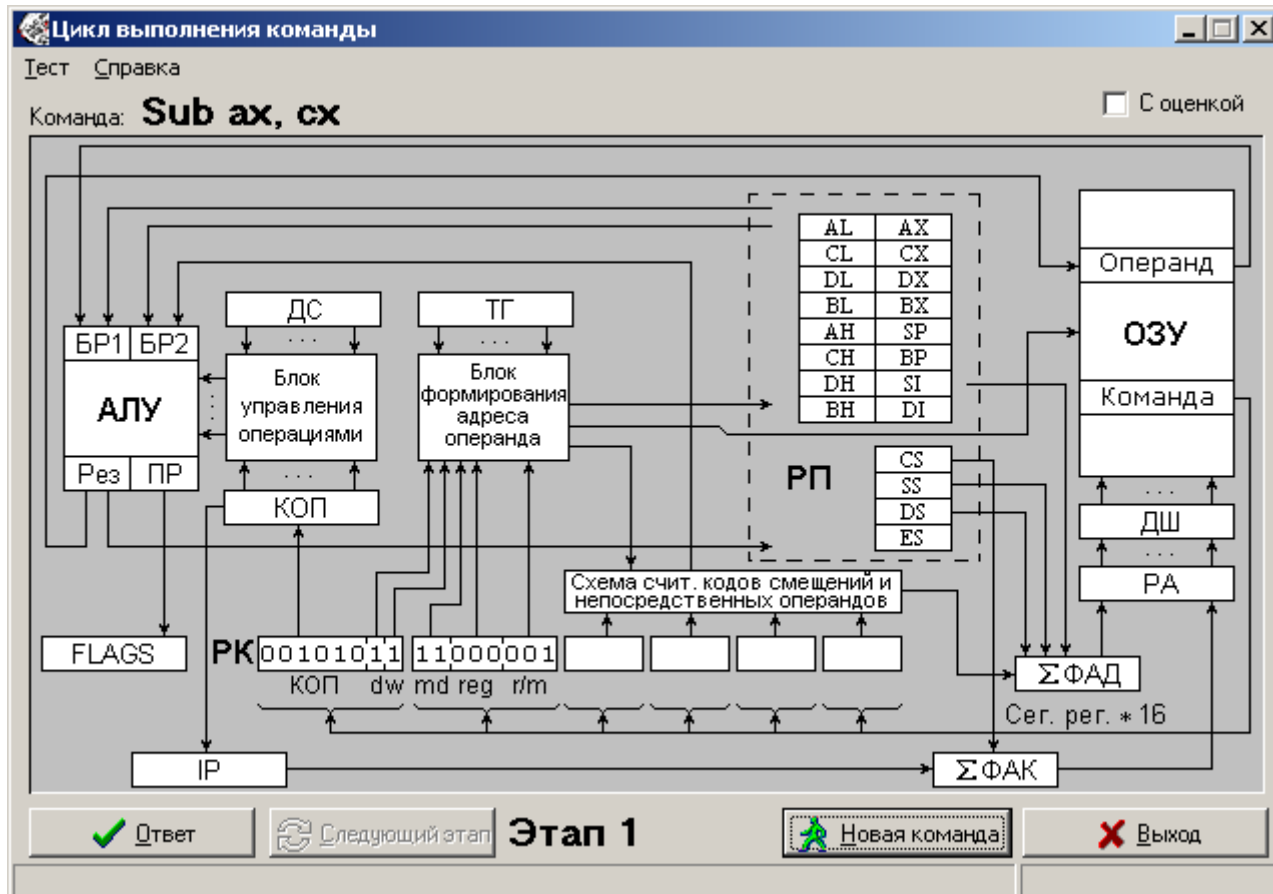
### Примеры команд A, I:

- add AX, 0AAh
- add AX, 5654h

Теперь рассмотрим этапы выполнения команды.

Допустим дана команда: **sub ax,cx**

**Первый этап.** На нём делается выборка команды из ОЗУ (выбор кода команды).



- Для реализации этого этапа необходимо код со счетчика команд (**ИП**) передать в сумматор физического адреса команды (**ΣФАК**).
- Также на **ΣФАК** необходимо передать значение регистра **CS**, умноженного на 16.

Немного теории. **СФАКИ** используются для получения адреса обращения к ОП с учётом её сегментной организации. Одним из слагаемых является начальный адрес сегмента, значение которого берётся путём умножения на 16 значения соответствующего сегментного регистра. А второе слагаемое – это смещение относительно начала сегмента. В качестве него тут выступает значение указателя команд **ИП**.

- На выходе **ΣФАК** формируется код физического адреса **ОЗУ**, по которому находится первый байт команды. Далее код с выхода **ΣФАК** поступает на регистр адреса (**РА**) **ОЗУ**. Регистр адреса служит для хранения адреса, по которому происходит обращение к **ОЗУ**, на время этого обращения.
- С **РА** код передаётся в дешифратор (**ДШ**). Дешифратор преобразует поступающий на него адрес в унитарный код, который непосредственно воспринимается физическими элементами схем памяти. На его выходах всегда имеется одна и только одна возбужденная шина, соответствующая адресу выбираемой ячейки.
- Дешифратор коммутирует схемы **ОЗУ** таким образом, чтобы на выходе получить код команды.
- С **ОЗУ** код команды записывается в регистр команд (**РК**). Регистр команд предназначен для хранения в процессоре считанной из **ОЗУ** команды на время ее выполнения. В

данной работе предполагается, что команда считывается за одно обращение к памяти (а вообще она считывается по одному байту).

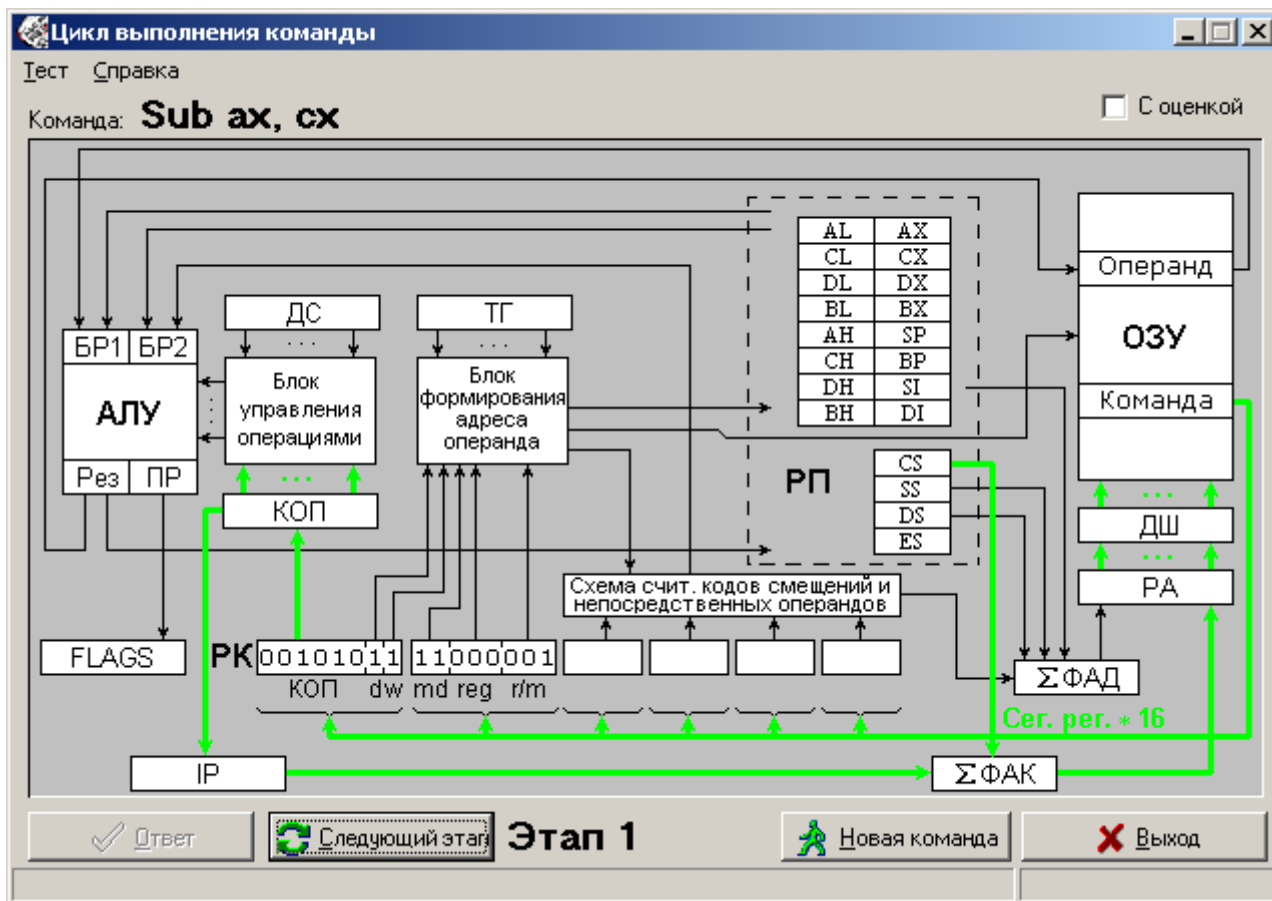
- Далее поле кода операции (**КОП**) передается на коммутатор операции (тоже **КОП** – не путать с кодом операции!). Коммутатор операции определяет тип команды.
- При командах типа «**R/M, I**» (команды вида <команда> <операнд-регистр или операнд-память>, <непосредственный операнд>) на коммутатор операции передаются также значения бит **s** и **w**, а также кода операции **КОП** из второго блока регистра команд. В нашей команде данный пункт не требуется.

**Немного теории.** Сигнал с коммутатора операции настраивает блок управления операциями таким образом, что на его выходах формируются управляющие сигналы ( $УС_i$ ), которые необходимы для автоматического выполнения всего цикла команды вплоть до занесения в **РК** новой команды. Формирование  $УС_i$  проходит на основе сигналов с датчика сигналов (**ДС**), который вырабатывает импульсы, равномерно распределенные по своим выходам. Регистр команд, дешифратор кода операции, блок управления операциями, датчик сигналов, счетчик команд составляют устройство управления.

- На этом же этапе счетчик команд **IP** увеличивается на длину считанной команды, формируя адрес следующей команды.

**Вкратце на первом этапе** должны быть отмечены следующие стрелки (голубой цвет – то, что не всегда используется, остальное – всегда):

- $IP \rightarrow \Sigma\text{ФАК}$
- $CS \rightarrow \Sigma\text{ФАК}$
- Сег. рег. \* 16
- $\Sigma\text{ФАК} \rightarrow \text{РА}$
- $\text{РА} \rightarrow \text{ДШ}$
- $\text{ДШ} \rightarrow \text{ОЗУ}$
- $\text{ОЗУ (команда)} \rightarrow \text{РК}$
- $\text{РК (КОП – код операции)} \rightarrow \text{КОП (коммутатор операции)}$
- **РК (если есть оба бита  $s$  и  $w$ , а также КОП во втором блоке РК)  $\rightarrow$  КОП (коммутатор операции). Данные стрелки отмечаются, если они вообще есть (грубо говоря ВСЕ стрелки, идущие от РК к КОП).**
- $\text{КОП} \rightarrow \text{Блок управления операциями}$
- $\text{КОП} \rightarrow \text{IP}$



## Второй этап – выборка первого операнда.

Будем всегда считать, что для команд типа "R/M, R" и "R/M, I" первый операнд определяется полями **md**, **r/m**; для команд типа "A, I" первым операндом является регистр **AX**. Для каждого типа команд действия будут различными. Рассмотрим возможные варианты:

### 1. Команды типа «R/M, R» (наша команда).

- Из **РК** на **блок формирования адреса операнда** передаются значения полей **w**, **md** и **r/m**.
- Далее, синхронизируя сигналы с помощью тактового генератора (**ТГ**), **блок формирования адреса операнда** производит необходимую коммутацию оборудования.

А вот дальше мы имеем два случая.

<b>md=11 (регистр и регистр)</b>	<b>md≠11 (память и регистр)</b>
<ul style="list-style-type: none"> <li><b>Блок формирования адреса операнда</b> коммутирует (передает сигнал) регистровую память (<b>РП</b>) так, чтобы на её выходе сформировалось значение запрашиваемого в <b>r/m</b> регистра.</li> </ul> <p>Определим, какой же из двух регистров надо считать первым. У нас <b>md=11</b>, поэтому надо смотреть таблицу регистров (выше). <b>Где же – в reg или r/m находится код регистра? Это определяется по биту d. Если d=0, то в reg; если d=1, то в r/m.</b> Смотрим значение <b>r/m</b> и ищем в таблице. У нас</p>	<p>Тут всё ясно: первый операнд – это память. Но есть два случая:</p> <p>а) В операнде поля <b>Disp</b> нет (нет чисел в []).</p> <ul style="list-style-type: none"> <li><b>Блок формирования адреса операнда</b> коммутирует (передает сигнал) регистровую память (<b>РП</b>), отмечаем используемые регистры при формировании адреса и сегментный регистр (если в операнде присутствует регистр <b>BP</b>, то отмечается <b>SS</b>, если отсутствует, то <b>DS</b>).</li> <li>Значения регистров, используемых при формировании адреса в <b>ОЗУ</b>,</li> </ul>

**r/m=001**, ищем в таблице совпадение:  
**СХ** – это наш первый операнд.

- С выхода регистровой памяти (**РП**) передаётся значение регистра на первый буферный регистр (**БР1**) АЛУ.

поступают на сумматор физического адреса данных ( $\Sigma\Phi\text{АД}$ ).

- б) В операнде есть поле **Disp** (есть числа в []). В этом случае выполняется пункт а) и плюс это:

- **Блок формирования адреса операнда** коммутирует (передает сигнал) **схемы считывания кодов смещений и непосредственных операндов**.
- Значения полей **Disp** считываются на **схемы считывания кодов смещений и непосредственных операндов**.
- С выхода **схем считывания кодов смещений и непосредственных операндов** значения идут на  $\Sigma\Phi\text{АД}$ .

Далее выполняются общие действия для каждого случая:

- На  $\Sigma\Phi\text{АД}$  передаётся значение одного из сегментных регистров. Если в операнде присутствует регистр **ВР**, то передаётся **SS**, если отсутствует, то **DS**.
- Умножение сегментного регистра на 16.
- На выходе  $\Sigma\Phi\text{АД}$  формирует физический адрес данных:  
 $(\Phi\text{АД} = \text{сегментный регистр} * 16 [+ \text{регистры смещения}] [+ \text{Disp}])$ .  
 $\Phi\text{АД}$  передаётся на регистр адреса (**РА**).
- С **РА** передаётся на дешифратор (**ДШ**).
- В соответствии с управляющим сигналом из блока формирования адреса операнда происходит коммутация **ОЗУ**.
- Благодаря коммутации на выходе из **ОЗУ** значение операнда передаётся на первый буферный регистр (**БР1**) АЛУ.

## 2. Команды типа «**R/M, I**» (регистр/память, непосредственный операнд).

Один из операндов – <b>память (md≠11)</b>	Один из операндов – <b>регистр</b>
<ul style="list-style-type: none"> <li>• На <b>блок формирования адреса операнда</b> считывается значение бита <b>s</b>.</li> <li>• Остальные действия аналогичны пункту 1.</li> </ul>	<ul style="list-style-type: none"> <li>• Все действия аналогичны пункту 1 при <b>md=11</b>.</li> </ul>

## 3. Команды типа «**A, I**» (регистр **АХ**, непосредственный операнд).

- **Тактовый генератор (ТГ)** синхронизирует сигналы **блока формирования адреса операнда**.

- С помощью **ТГ блок формирования адреса операнда** коммутирует (передает сигнал) регистровую память (**РП**), отмечаем регистр **АХ**.
- С выхода регистровой памяти (**РП**) передается значение регистра **АХ** на первый буферный регистр (**БР1**) АЛУ.

**Вкратце на втором этапе** должны быть отмечены следующие стрелки (по вариантам):

1. Команды типа «**R/M, R**» (наша команда).

- w (РК) → Блок формирования адреса операнда.
- md (РК) → Блок формирования адреса операнда.
- r/m (РК) → Блок формирования адреса операнда.
- ТГ → Блок формирования адреса операнда.

md=11 (регистр и регистр)	md≠11 (память и регистр)
<ul style="list-style-type: none"> <li>• Блок формирования адреса операнда → РП (смотрим значение r/m и ищем в таблице регистров, выбираем этот регистр).</li> <li>• РП → БР1.</li> </ul>	<p>а) В операнде поля Disp нет (чисел в []).</p> <ul style="list-style-type: none"> <li>• Блок формирования адреса операнда → РП (отмечаем указанные в квадратных скобках регистры + регистр SS (если BP есть в []) / DS (если BP нет в [])).</li> <li>• РП → ΣФАД.</li> </ul> <p>б) В операнде есть поле Disp (есть числа в []). В этом случае выполняется пункт а) и плюс это:</p> <ul style="list-style-type: none"> <li>• Блок формирования адреса операнда → схемы считывания кодов смещений и непосредственных операндов.</li> <li>• Disp → схемы считывания кодов смещений и непосредственных операндов.</li> <li>• Схемы считывания кодов смещений и непосредственных операндов → ΣФАД.</li> </ul> <p>Далее выполняются общие действия для каждого случая:</p> <ul style="list-style-type: none"> <li>• SS (если BP есть в []) / DS (если BP нет в []) → ΣФАД</li> <li>• Сег. рег. * 16.</li> <li>• ΣФАД → РА</li> <li>• РА → ДШ</li> <li>• Блок формирования адреса операнда → ОЗУ</li> <li>• ОЗУ (операнд) → БР1.</li> </ul>

2. Команды типа «**R/M, I**» (регистр/память, непосредственный операнд).

Один из операндов – память (md≠11)	Один из операндов – регистр
<ul style="list-style-type: none"> <li>• s (РК) → блок формирования адреса операнда.</li> <li>• Остальные действия аналогичны</li> </ul>	<ul style="list-style-type: none"> <li>• Все действия аналогичны пункту 1 при md=11.</li> </ul>

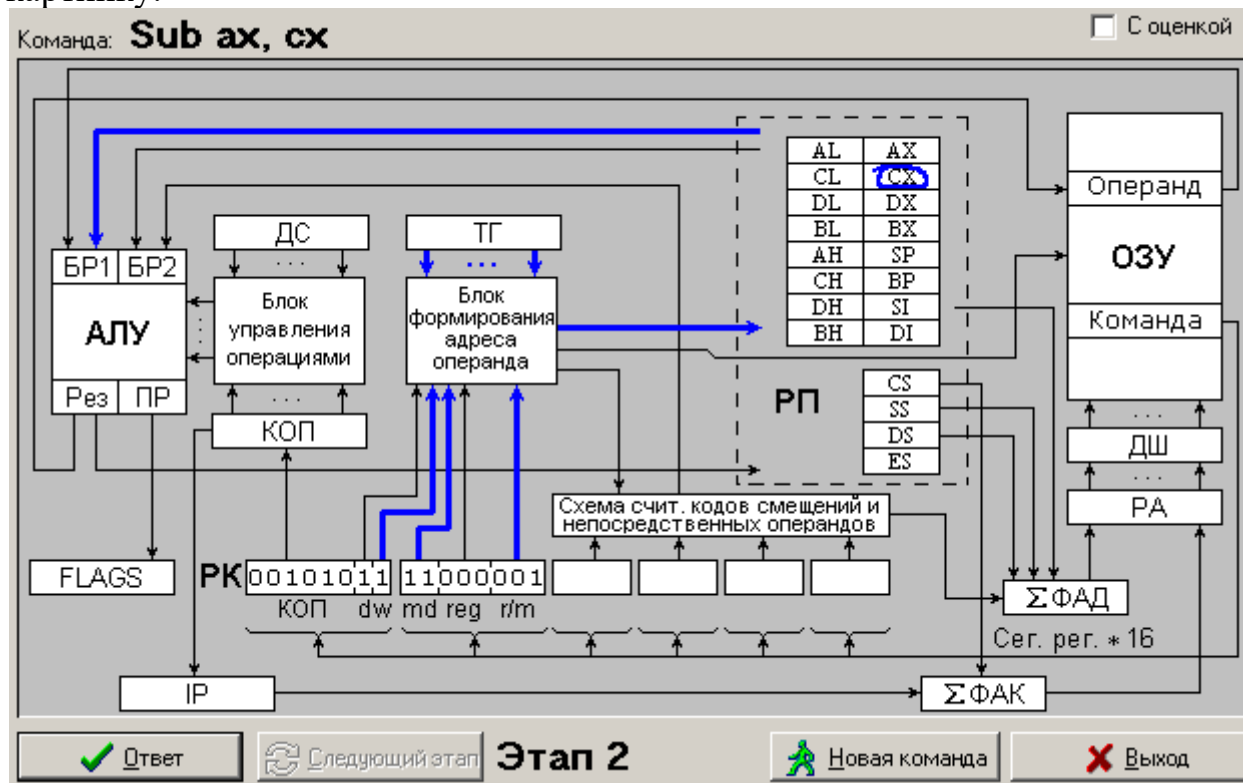


пункту 1.

3. Команды типа «**A, I**» (регистр AX, непосредственный операнд).

- ТГ → блок формирования адреса операнда.
- Блок формирования адреса операнда → РП (отмечаем AX).
- РП → БР1.

Наш случай относится к пункту 1, когда **md**=11. Пройдясь по пунктам, имеем такую картинку:



**Этап 3. Считывание второго операнда из регистра команд (РК).**

Для каждого типа команд действия будут различными. Рассмотрим возможные варианты:

1. Команды типа «**R/M, R**».

- Из **РК** на **блок формирования адреса операнда** передаются значения полей **w** и **reg**.
- Синхронизация **блока формирования адреса операнда** с тактовым генератором (**ТГ**).
- **Блок формирования адреса операнда** посылает сигналы в регистровую память (**РП**). Смотрим значение **reg** и ищем в таблице регистров. У нас **reg**=000, ищем в таблице совпадение: **AX** – это наш второй операнд.
- С выхода регистровой памяти передаётся значение регистра на второй буферный регистр (**БР2**) АЛУ.

2. Команды типа «**R/M, I**» и «**A, I**». Второй операнд содержится в полях **Im**.

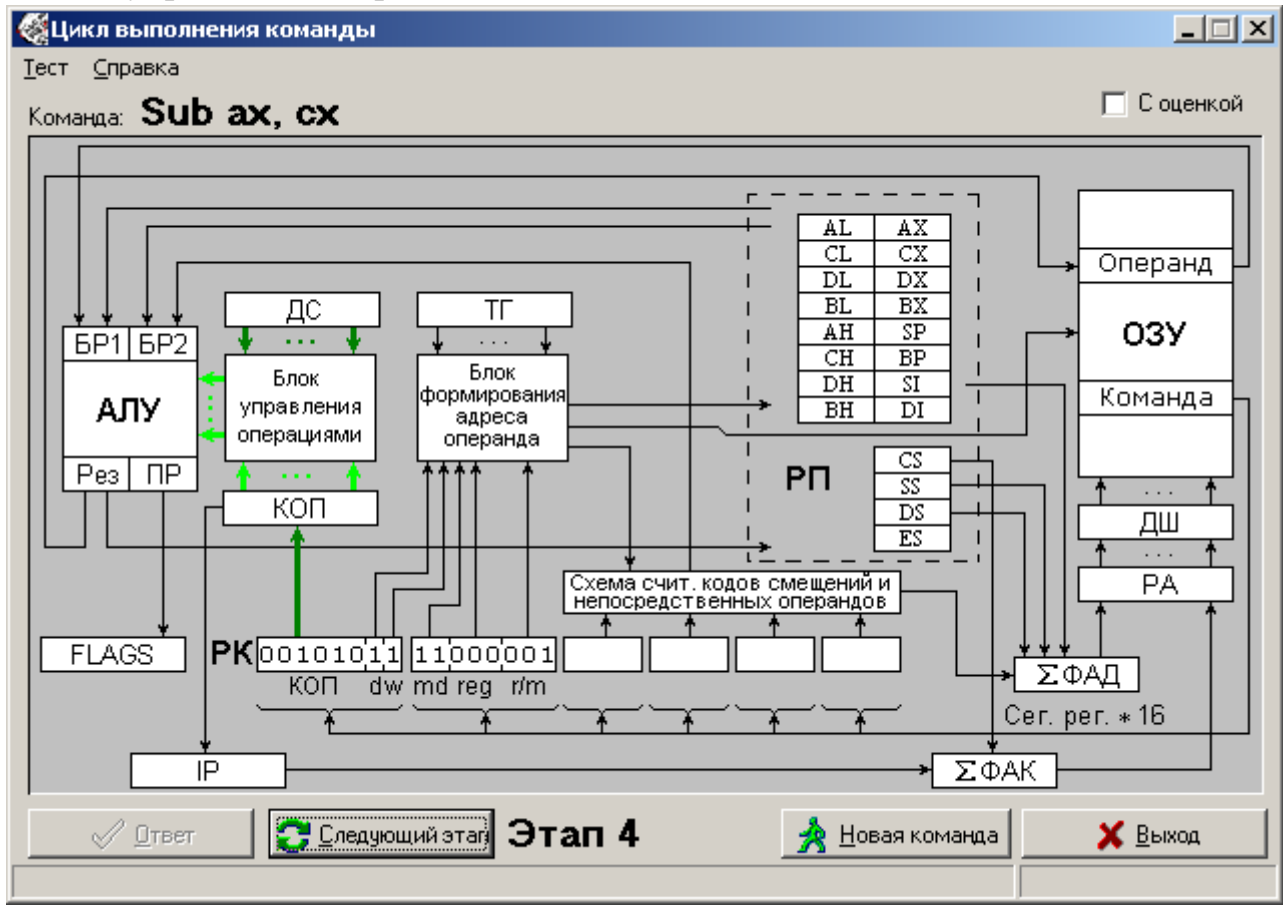
- Из **РК** на **блок формирования адреса операнда** передаются значения полей **s** (для «**R/M, I**») и **w** (для обоих типов), которые определяют разрядность операнда.
- Синхронизация блока формирования адреса операнда с тактовым генератором (**ТГ**).
- **Блок формирования адреса операнда** коммутирует (передаёт сигнал) **схемы считывания кодов смещений и непосредственных операндов**.
- Значения полей **Im** считываются на **схемы считывания кодов смещений и непосредственных операндов**.
- С выхода **схем считывания кодов смещений и непосредственных операндов** значение непосредственного операнда идёт на второй буферный регистр (и) АЛУ.

- Поле кода операции команды (**КОП** в **РП**) передаётся в коммутатор операции (**КОП**).
- При командах типа «**R/M, I**» (команды вида <команда> <операнд-регистр или операнд-память>, <непосредственный операнд>) на коммутатор операции передаются также значения бит **s** и **w**, а также кода операции из второго блока регистра команд. В нашей команде данный пункт не требуется.

- **КОП** определяет тип команды и коммутирует оборудование блока управления операциями таким образом, чтобы на его выходе были сигналы, необходимые **АЛУ** для выполнения операции.
- Подача сигналов на **АЛУ** регулируется датчиком сигналов (**ДС**).
- Сигналы передаются **АЛУ** для выполнения арифметической операции в **АЛУ**.

Вкратце на четвёртом этапе должны быть отмечены следующие стрелки:

- КОП (РК) → КОП.
- РК (если есть оба бита s и w, а также КОП во втором блоке РК) → КОП (коммутатор операции). Данные стрелки отмечаются, если они вообще есть (грубо говоря ВСЕ стрелки, идущие от РК к КОП).
- КОП → Блок управления операциями.
- ДС → Блок управления операциями.
- Блок управления операциями → АЛУ.



**Этап 5. Запись результата.**

Для каждого типа команд действия будут различными. Рассмотрим возможные варианты:

1. Команды типа «**R/M, R**».

- Приёмник определяется полем **d** в регистре команд (**РК**). Значение этого бита идёт на блок формирования адреса операнда.

Далее два различных случая:

<b>d=1</b> (приёмник определяется полем reg)	<b>d=0</b> (приёмник определяется полями md, r/m)	
	<b>md=11</b> (приёмник – регистр)	<b>md≠11</b> (приёмник – память)
<ul style="list-style-type: none"><li>• Синхронизация тактового генератора (<b>ТГ</b>) с <b>блоком формирования адреса операнда</b>.</li><li>• <b>Блок формирования адреса</b></li></ul>	<ul style="list-style-type: none"><li>• Действия аналогичны действиям при <b>d=1</b> (только вместо <b>reg</b></li></ul>	<ul style="list-style-type: none"><li>• Синхронизация тактового генератора (<b>ТГ</b>) с <b>блоком</b></li></ul>

<p><b>операнда</b> коммутирует (передаёт сигнал) регистровую память (<b>РП</b>), подготавливая нужный регистр для приёма значения.</p> <p><b>Приёмник – всегда первый операнд.</b> А так берём значение из <b>reg</b> и ищем в таблице регистров. У нас <b>reg=000</b>, ищем в таблице совпадение: <b>АХ</b> – туда запишется результат.</p> <ul style="list-style-type: none"> <li>• <b>АЛУ</b> посылает в <b>РП</b> результат.</li> </ul>	берётся <b>r/m</b> ).	<p><b>формирования адреса операнда.</b></p> <ul style="list-style-type: none"> <li>• В регистре адреса (<b>РА</b>) уже хранится физический адрес <b>ОЗУ</b>, вычисленный при обращении к этой ячейке за операндом.</li> <li>• Из <b>РА</b> значение адреса передаётся в дешифратор (<b>ДШ</b>).</li> <li>• <b>ДШ</b> отсылает сигнал в <b>ОЗУ</b>, подготавливая место для записи.</li> <li>• <b>Блок формирования адреса операнда</b> управляет пересылкой результата из <b>АЛУ</b> в <b>ОЗУ</b>.</li> <li>• <b>АЛУ</b> посылает в <b>ОЗУ</b> результат.</li> </ul>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2. Команды типа «**R/M, I**».

- Действия аналогичны пункту 1 таблицы при **d=0**.

## 3. Команды типа «**A, I**».

- Синхронизация тактового генератора (**ТГ**) с **блоком формирования адреса операнда**.
- **Блок формирования адреса операнда** коммутирует (передаёт сигнал) регистровую память (**РП**), подготавливая регистр **АХ** для приёма значения.
- **АЛУ** посылает в **РП (АХ)** результат.

Для всех случаев:

- Запись признаков результата (**ПР**) из **АЛУ** в регистр флагов (**FLAGS**).

**Вкратце на пятом этапе** должны быть отмечены следующие стрелки:

## 1. Команды типа «**R/M, R**».

- **d (РК) → блок формирования адреса операнда.**

Далее два различных случая:

d=1 (приёмник определяется полем reg)	d=0 (приёмник определяется полями md, r/m)	
	md=11 (приёмник – регистр)	md≠11 (приёмник – память)
<ul style="list-style-type: none"> <li>• <b>ТГ → блок формирования адреса операнда.</b></li> <li>• <b>Блок формирования адреса операнда → РП</b> (смотрим значение reg и ищем в</li> </ul>	<ul style="list-style-type: none"> <li>• Действия аналогичны действиям при d=1 (только вместо reg берётся r/m).</li> </ul>	<ul style="list-style-type: none"> <li>• <b>ТГ → блок формирования адреса операнда.</b></li> <li>• <b>РА → ДШ.</b></li> </ul>

таблице регистров, выбираем этот регистр).		<ul style="list-style-type: none"> <li>ДШ → ОЗУ.</li> <li>Блок формирования адреса операнда → ОЗУ.</li> <li>АЛУ → ОЗУ.</li> </ul>
--------------------------------------------	--	-----------------------------------------------------------------------------------------------------------------------------------

## 2. Команды типа «R/M, I».

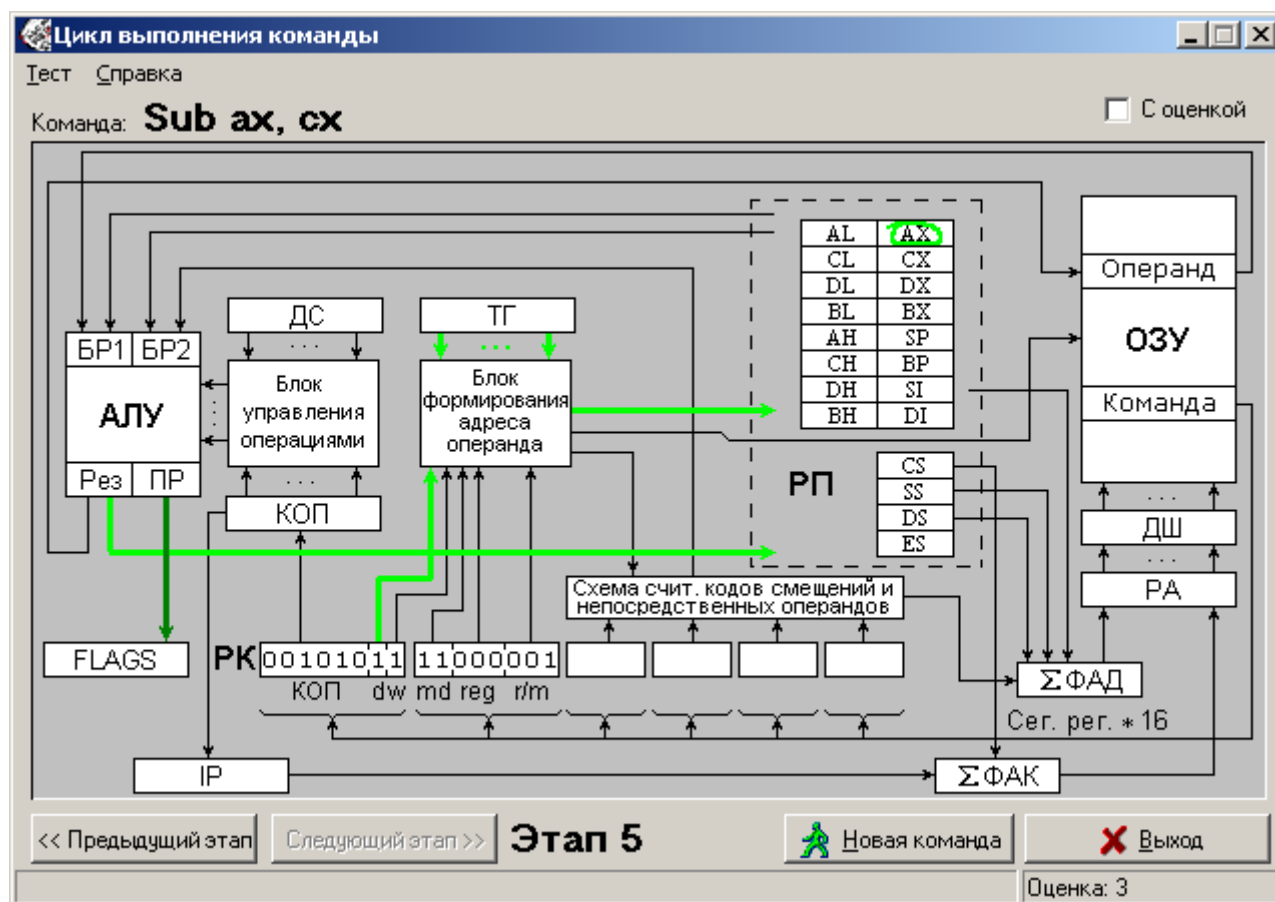
- Действия аналогичны пункту 1 таблицы при d=0.

## 3. Команды типа «A, I».

- ТГ → блок формирования адреса операнда.
- Блок формирования адреса операнда → РП (АХ).
- АЛУ → РП (АХ).

Для всех случаев:

- РП (АЛУ) → FLAGS.



**Примечание.** Если в команде встретится word ptr или byte ptr, то это в общем-то ни на что не влияет. Это пишется, чтобы определить, что брать из памяти – слово или байт. Например, есть команда `add byte ptr [bp+si+7120h], 0C6h`. Это значит, что мы обращаемся к ОЗУ по адресу `[bp+si+7120h]` и берём оттуда всего байт. Если было бы указано word ptr, то из ОЗУ брали бы слово, то есть два байта.