



Арифметический сопроцессор



История сопроцессоров

- Intel 8087, Intel 80287, Intel 80387 – отдельные, независимые устройства
- Начиная с Intel 80486 – составная часть процессора



Типы данных

- Двоичные **целые** числа (16, 32, 64 бит)
- Двоично-десятичные числа (80 бит)
- Вещественные числа (32, 64, 80 бит)

Специальные значения (константы)

- Бесконечность (+ или —)
- Нечисло, неопределенность и т.д.

Архитектура сопроцессора

13 регистров:

- 8 регистров – стек сопроцессора (80 бит)
- 1 регистр тегов
- 1 регистр состояния
- 1 регистр управления
- 1 регистр – указатель команды
- 1 регистр – указатель данных

Стек сопроцессора

- **ST** или **ST(0)** – вершина стека
- **ST(1)**
- ...
- **ST(6)**
- **ST(7)** – дно стека

Аналоги регистров EAX, EBX, ECX, EDX, ...



Регистр тегов


Описывает состояние каждого регистра стека:

- Регистр свободен
- Регистр занят ненулевым числом
- Регистр занят нулевым числом
- Регистр занят специальным значением

Регистр состояния

**Описывает состояние сопроцессора
после последней команды:**

- Указатель на вершину стека сопроцессора
- Биты флагов (*аналог SF, ZF и т.д.*)
- Биты исключительных состояний



Исключительные состояния сопроцессора

- Недействительный операнд
- Деление на 0
- Переполнение
- Потеря значимости
- Потеря точности

Регистр управления

Описывает особенности обработки данных:

- **Поле управления точностью**
 - Обычная
 - Двойная
 - Расширенная
- **Поле управления округлением**
 - К ближайшему
 - С избытком
 - С недостатком
 - С отсечением

Регистр управления

Описывает особенности обработки данных:

- Поле управления бесконечностью
 - *Аффинная*
 - *Проективная*
- Маски исключительных ситуаций
 - Генерация ошибки
 - Возврат специального значения



Регистр – указатель команды

Содержит адрес команды, вызвавшей
исключительную ситуацию



Регистр – указатель данных

Содержит адрес операнда в команде,
вызвавшей исключительную ситуацию



Система команд сопроцессора

- Команды пересылки данных
- Арифметические команды
- Трансцендентные команды
- Команды сравнения
- Команды управления

Мнемоника команд сопроцессора

- Первая буква команды – **F**
- Вторая буква – тип данных
 - **I** – двоичные целые числа
 - **B** – двоично-десятичные числа



Команды управления

- **FINIT**

- Инициализация сопроцессора

Пример

```
include masm32.inc  
includelib masm32.lib
```

```
...
```

```
.data
```

```
    number dq 12.345
```

```
.data?
```

```
    string db 20 dup (?)
```

```
.code
```

```
...
```

```
    PUSH offset string
```

```
    PUSH dword ptr number+4
```

```
    PUSH dword ptr number
```

```
    CALL FloatToStr
```


Команды пересылки данных

- **FLD** переменная
- **FILD** переменная
- **FBLD** переменная

аналог **PUSH**

Пример

.data

integer dd 2

float dd 3.0

double dq 4.0

.code

...

FILD integer

FLD float

FLD double

Команды пересылки данных

- **FST** переменная
- **FIST** переменная
- **FBST** переменная

аналог **MOV** переменная, [ESP]

Команды пересылки данных

- **FSTP** переменная
- **FISTP** переменная
- **FBSTP** переменная

аналог **POP**

Пример

.data

dPI dq 3.14159265358979323846

.data?

fPI dd ?

.code

...

FLD dPI

FSTP fPI

Команды пересылки данных

- **FLDZ**

- *FILD 0*

- **FLD1**

- *FILD 1*

- **FLDPI**

- *FLD 3.14159265358979323846*

...

Арифметические команды

- **FADD переменная**
- **FIADD переменная**
 - $ST = ST + \text{переменная}$
- **FADD ST(i), ST(j)**
 - $ST(i) = ST(i) + ST(j)$

Пример

.data

x dq 3.1

y dq 1.7

.data?

z dq ?

.code

...

FLD x

FADD y

FSTP z

.data

x dq 3.1

y dq 1.7

.data?

z dq ?

.code

...

FLD x

FLD y

FADD ST, ST(1)

FSTP z

Арифметические команды

- **FMUL переменная**

- **FIMUL переменная**

 - $ST = ST * \text{переменная}$

- **FMUL ST(i), ST(j)**

 - $ST(i) = ST(i) * ST(j)$

Арифметические команды

- **FSUB переменная**
- **FISUB переменная**
 - $ST = ST - \text{переменная}$
- **FSUB ST(i), ST(j)**
 - $ST(i) = ST(i) - ST(j)$

Арифметические команды

- **FDIV переменная**
- **FIDIV переменная**
 - $ST = ST / \text{переменная}$
- **FDIV ST(i), ST(j)**
 - $ST(i) = ST(i) / ST(j)$

Арифметические команды

- **FSUBR переменная**

- $ST = \text{переменная} - ST$

- **FDIVR переменная**

- $ST = \text{переменная} / ST$

Арифметические команды

- **FABS**

- $ST = |ST|$

- **FCHS**

- $ST = -ST$

- **FSQRT**

- $ST = \text{sqrt}(ST)$

- **FRNDINT**

Трансцендентные команды

- **FSIN**

- $ST = \sin(ST)$

- **FCOS**

- $ST = \cos(ST)$

Пример $x + y = z$

```
include masm32.inc  
includelib masm32.lib
```

```
...
```

```
.data
```

```
    template db "Выражение: %s + %s = %s",  
            0
```

```
.data?
```

```
    xString db 20 dup (?)  
    yString db 20 dup (?)  
    zString db 20 dup (?)  
    result db 80 dup (?)
```

Пример $x + y = z$

.code

...

PUSH offset zString

PUSH offset yString

PUSH offset xString

PUSH offset template

PUSH offset result

CALL wsprintf

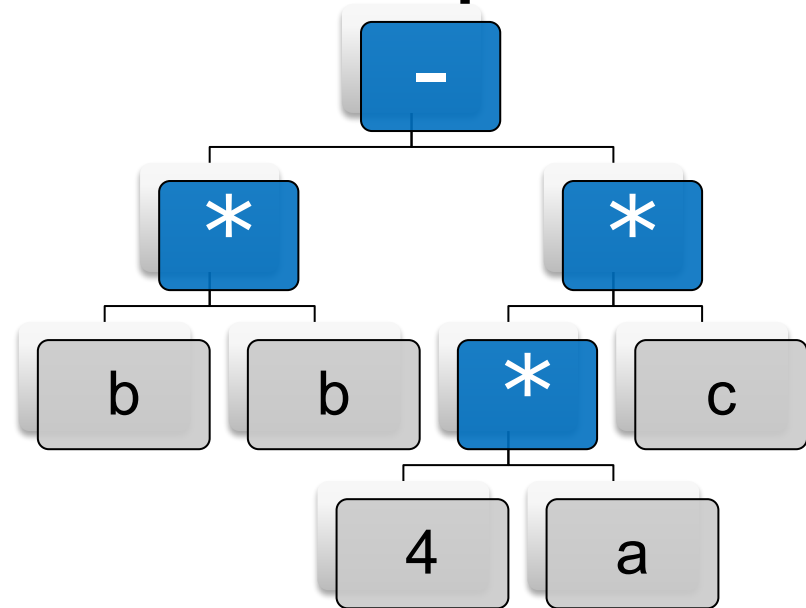
ADD ESP, 20

Вычисление сложных выражений

$$D = b^2 - 4ac$$

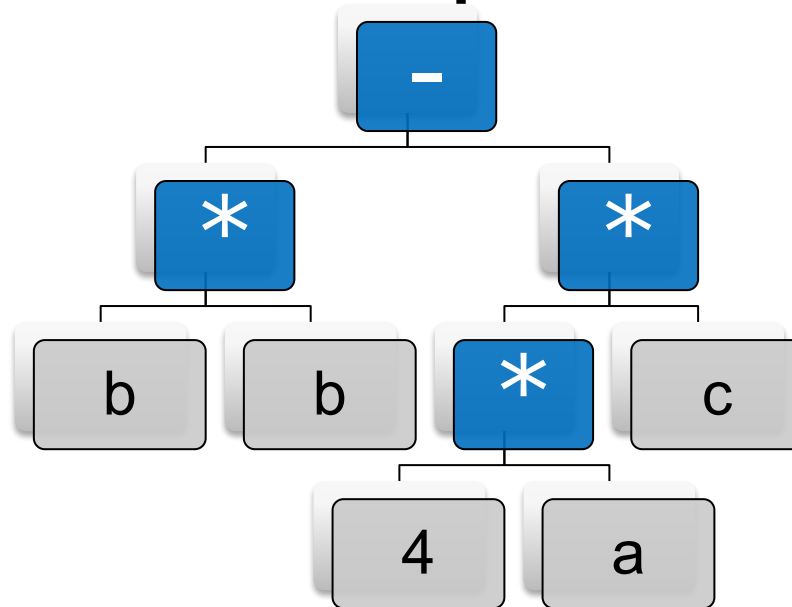
Вычисление сложных выражений

$$D = b^2 - 4ac$$



Вычисление сложных выражений

$$D = b^2 - 4ac$$

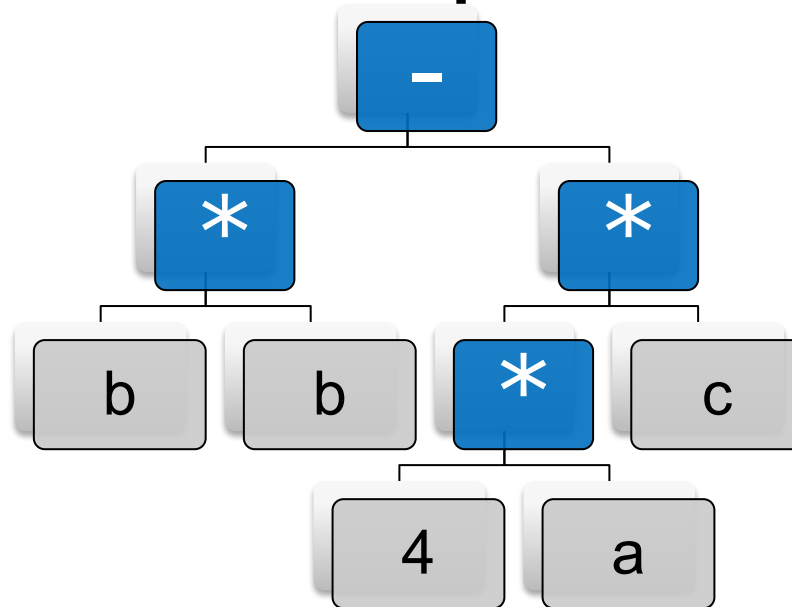


ST(0)	
ST(1)	
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**

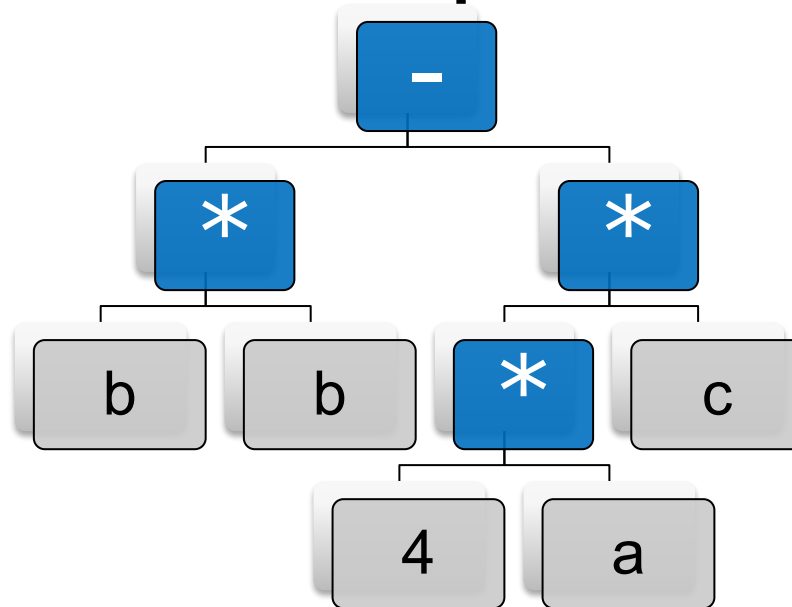


ST(0)	
ST(1)	
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**

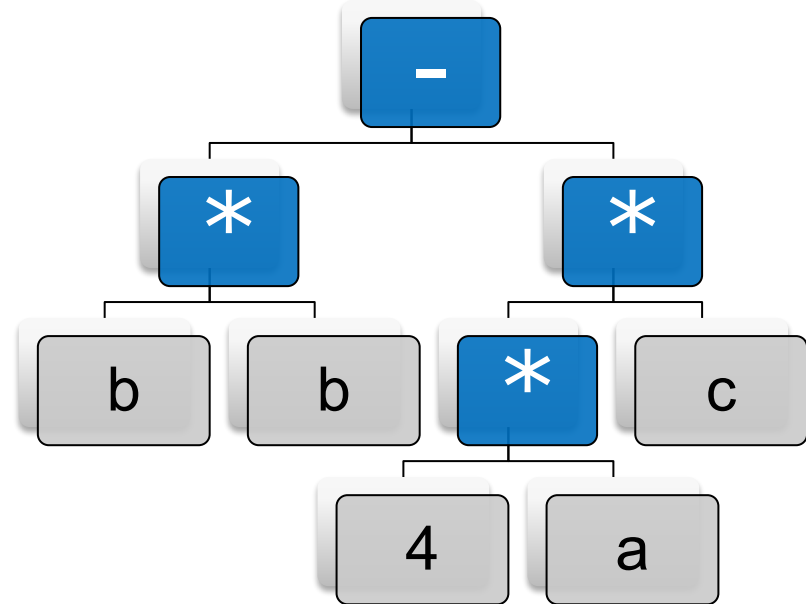


ST(0)	b
ST(1)	
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**

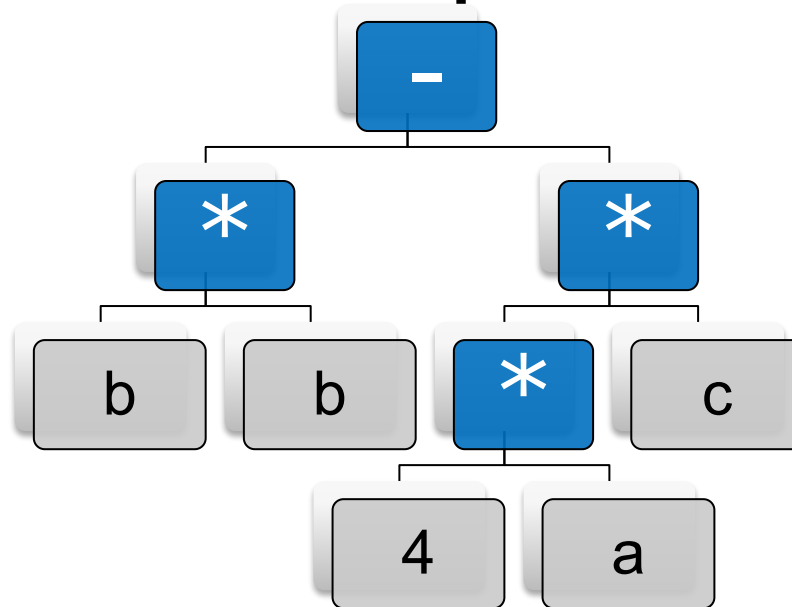


ST(0)	b
ST(1)	
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**



ST(0)	b^2
ST(1)	
ST(2)	
ST(3)	

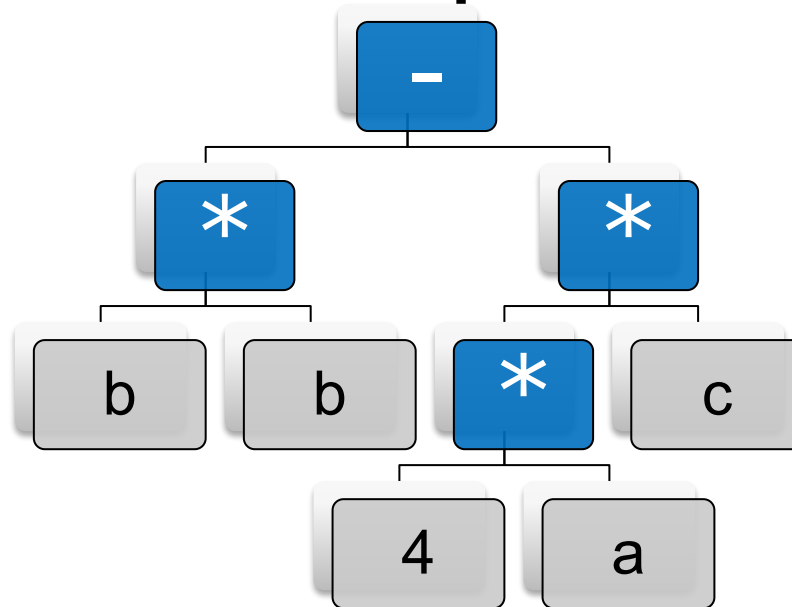
Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**

2. **FMUL b**

3. **FLD 4 ;!!!**

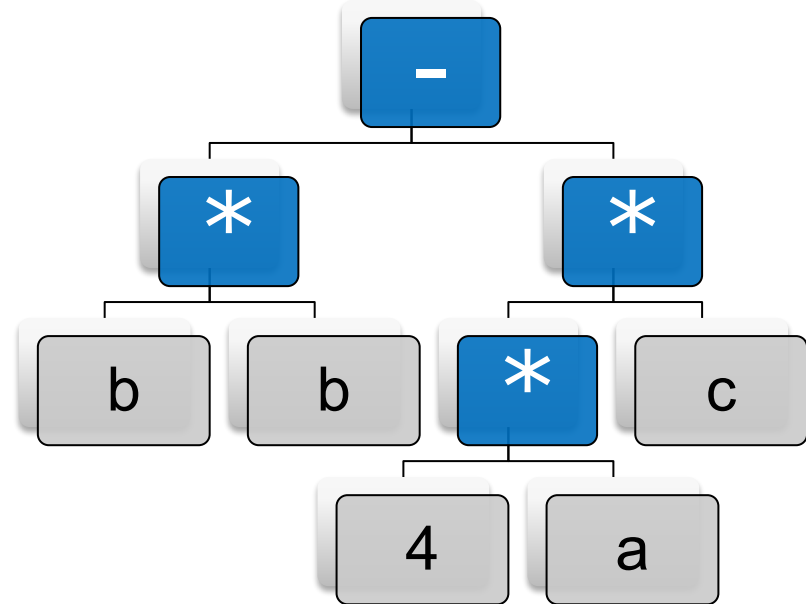


ST(0)	b^2
ST(1)	
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**
3. **FLD 4 ;!!!**

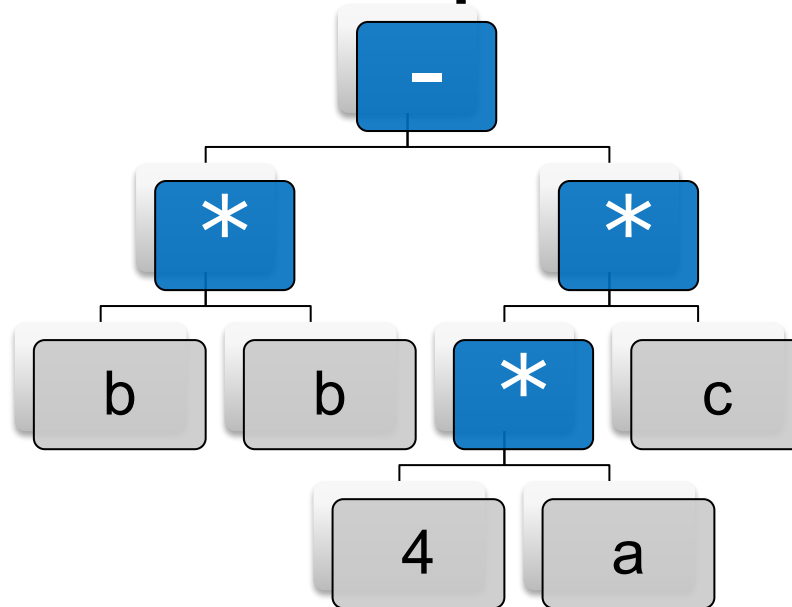


ST(0)	4
ST(1)	b^2
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**
3. **FLD 4 ;!!!**
4. **FMUL a**

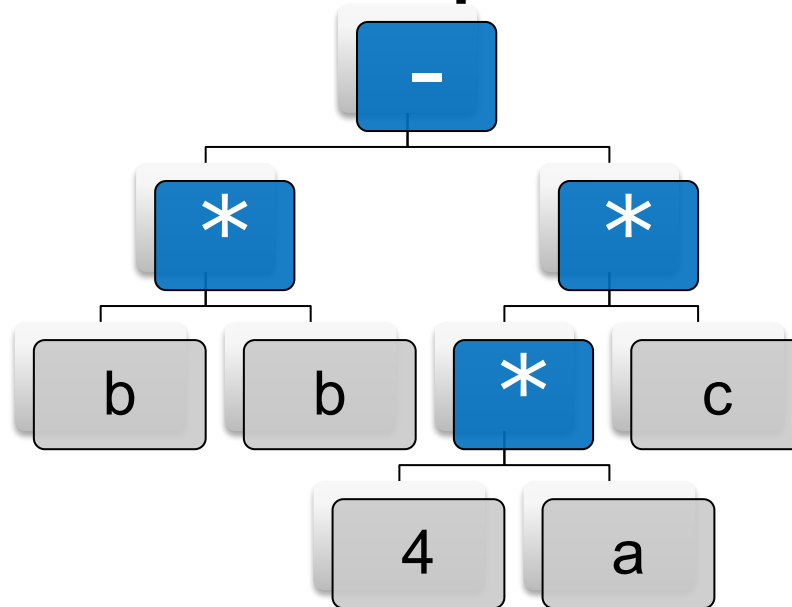


ST(0)	4
ST(1)	b^2
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**
3. **FLD 4 ;!!!**
4. **FMUL a**

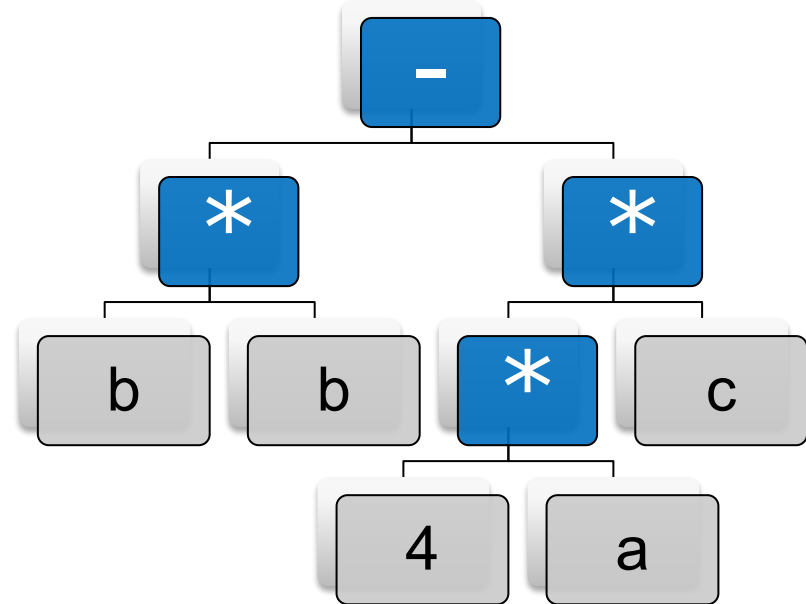


ST(0)	4a
ST(1)	b^2
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**
3. **FLD 4 ;!!!**
4. **FMUL a**
5. **FMUL c**

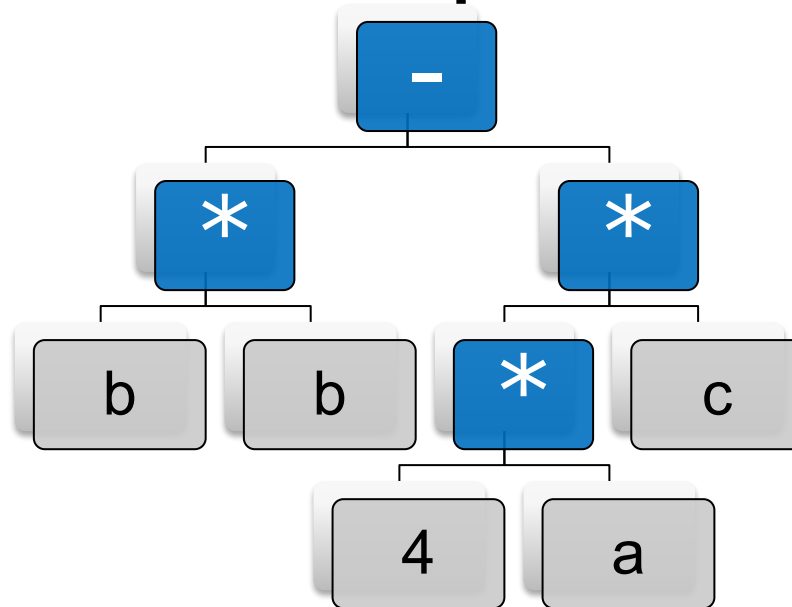


ST(0)	4a
ST(1)	b^2
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**
3. **FLD 4 ;!!!**
4. **FMUL a**
5. **FMUL c**

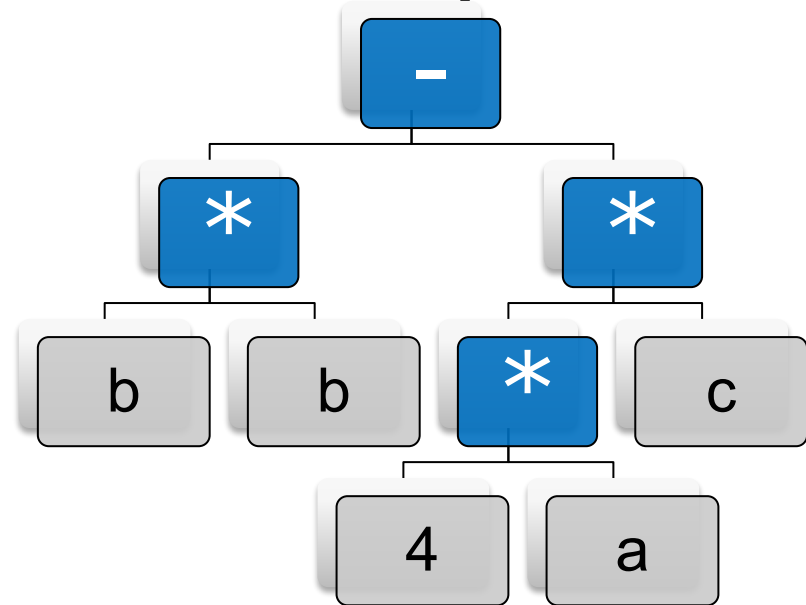


ST(0)	4ac
ST(1)	b^2
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**
3. **FLD 4 ;!!!**
4. **FMUL a**
5. **FMUL c**
6. **FSUBR ST, ST(1)**

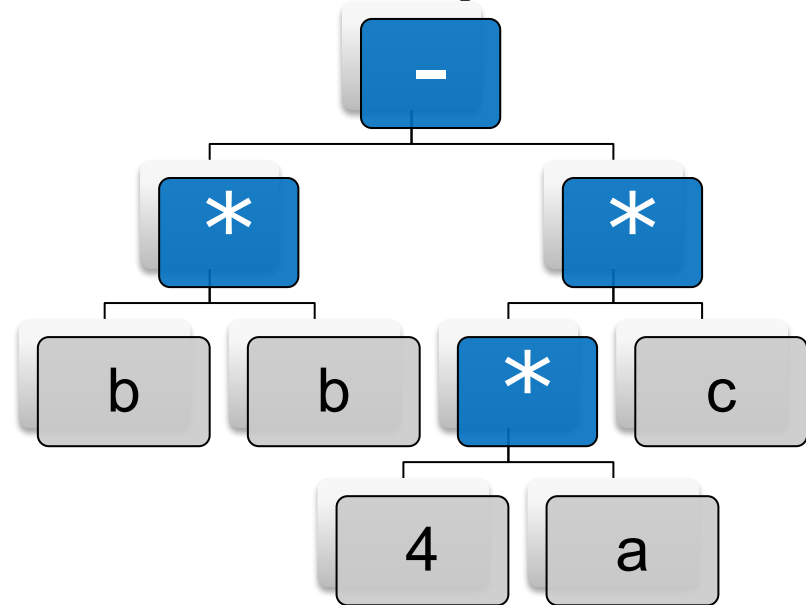


ST(0)	4ac
ST(1)	b^2
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**
3. **FLD 4 ;!!!**
4. **FMUL a**
5. **FMUL c**
6. **FSUBR ST, ST(1)**

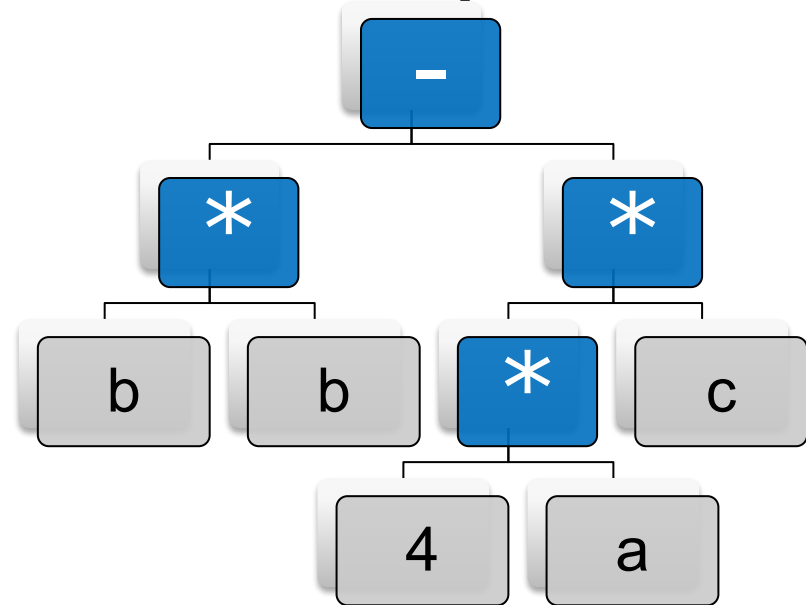


ST(0)	$b^2 - 4ac$
ST(1)	
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**
3. **FLD 4 ;!!!**
4. **FMUL a**
5. **FMUL c**
6. **FSUBR ST, ST(1)**
7. **FSTP D**

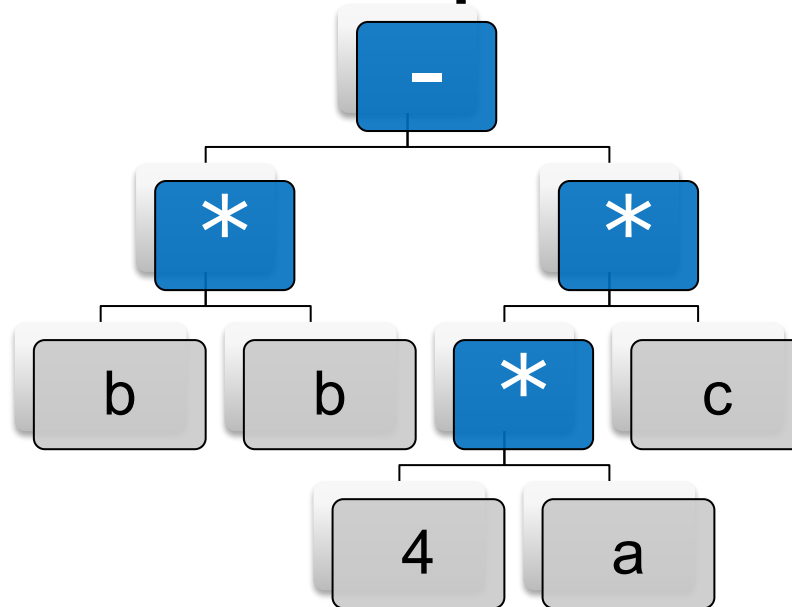


ST(0)	$b^2 - 4ac$
ST(1)	
ST(2)	
ST(3)	

Вычисление сложных выражений

$$D = b^2 - 4ac$$

1. **FLD b**
2. **FMUL b**
3. **FLD 4 ;!!!**
4. **FMUL a**
5. **FMUL c**
6. **FSUBR ST, ST(1)**
7. **FSTP D**



ST(0)	
ST(1)	
ST(2)	
ST(3)	

Команды сравнения

- **FCOM**

- *Сравнивает $ST(0)$ и $ST(1)$*

- **FCOM переменная**

- *Сравнивает $ST(0)$ и переменную*

- **FCOMP переменная**

- *Сравнивает $ST(0)$ и переменную*
 - *Выталкивает $ST(0)$ из стека*

Команды сравнения

- **FICOM переменная**

- *Сравнивает $ST(0)$ и переменную*

- **FICOMP переменная**

- *Сравнивает $ST(0)$ и переменную*
 - *Выталкивает $ST(0)$ из стека*

- **FTST**

- *Сравнивает $ST(0)$ и 0*

Результат работы команд сравнения

условие	C3	C2	C0
$ST(0) > \text{операнда}$	0	0	0
$ST(0) < \text{операнда}$	0	0	1
$ST(0) = \text{операнду}$	1	0	0

Вспомогательные команды

■ **FSTSW**

- *Записывает регистр состояния **сопроцессора** в регистр **AX***

■ **SAHF**

- *Записывает регистр **AH** в младший байт регистра флагов **процессора***

Пример

.data

number dq 13.5

.code

...

FLDZ

FCOM number ; аналог CMP

FSTSW

SAHF

JE label