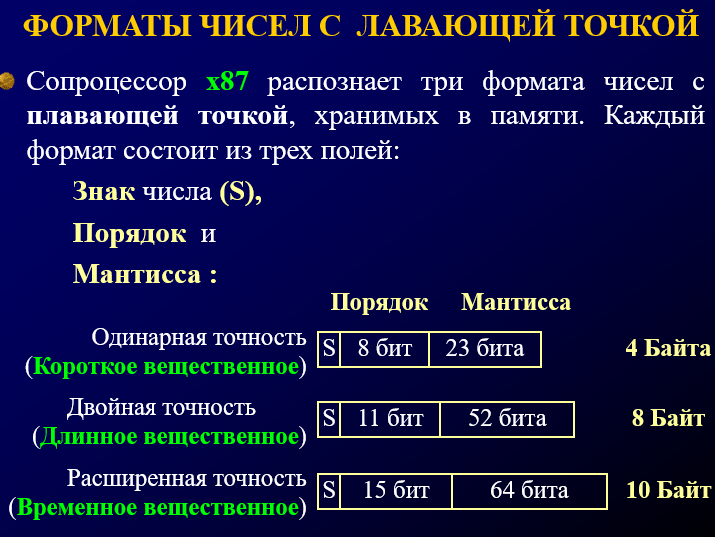
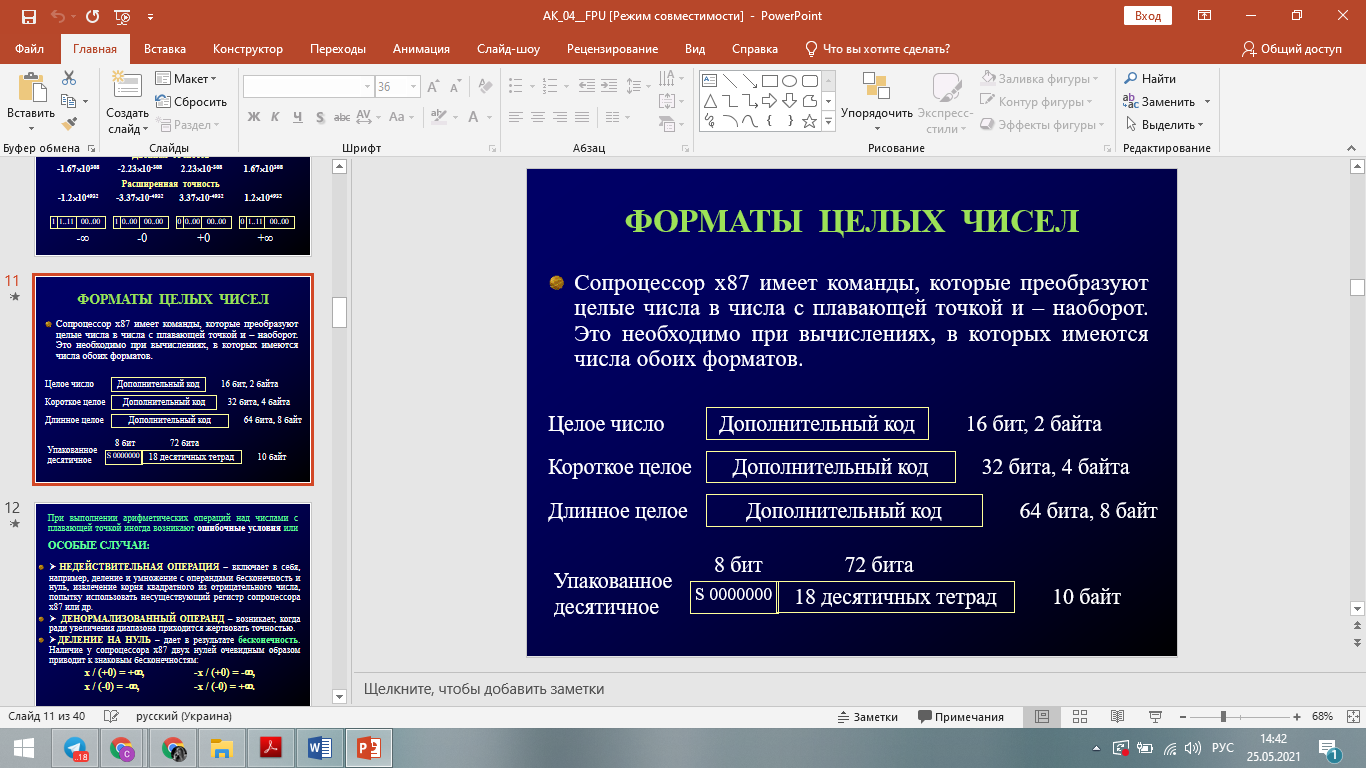
1. Форматы данных сопроцессора.

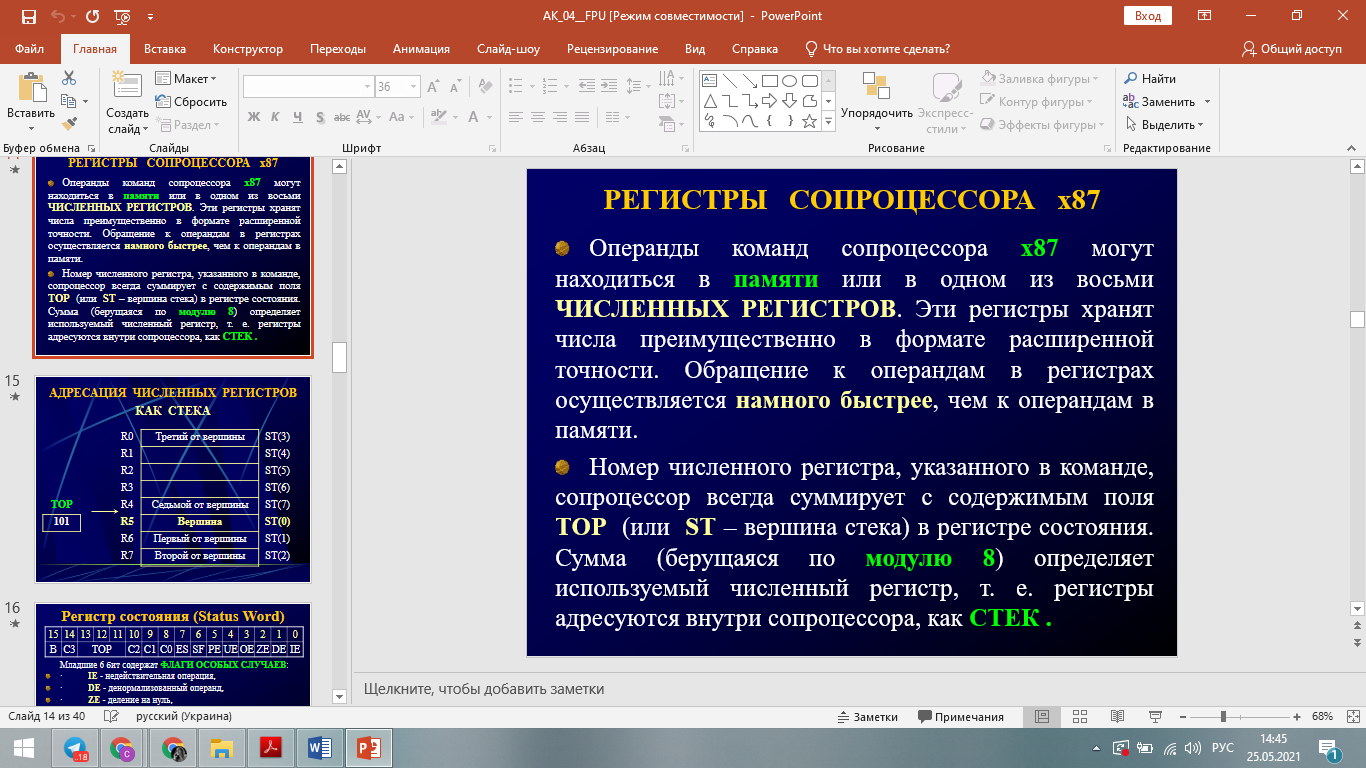


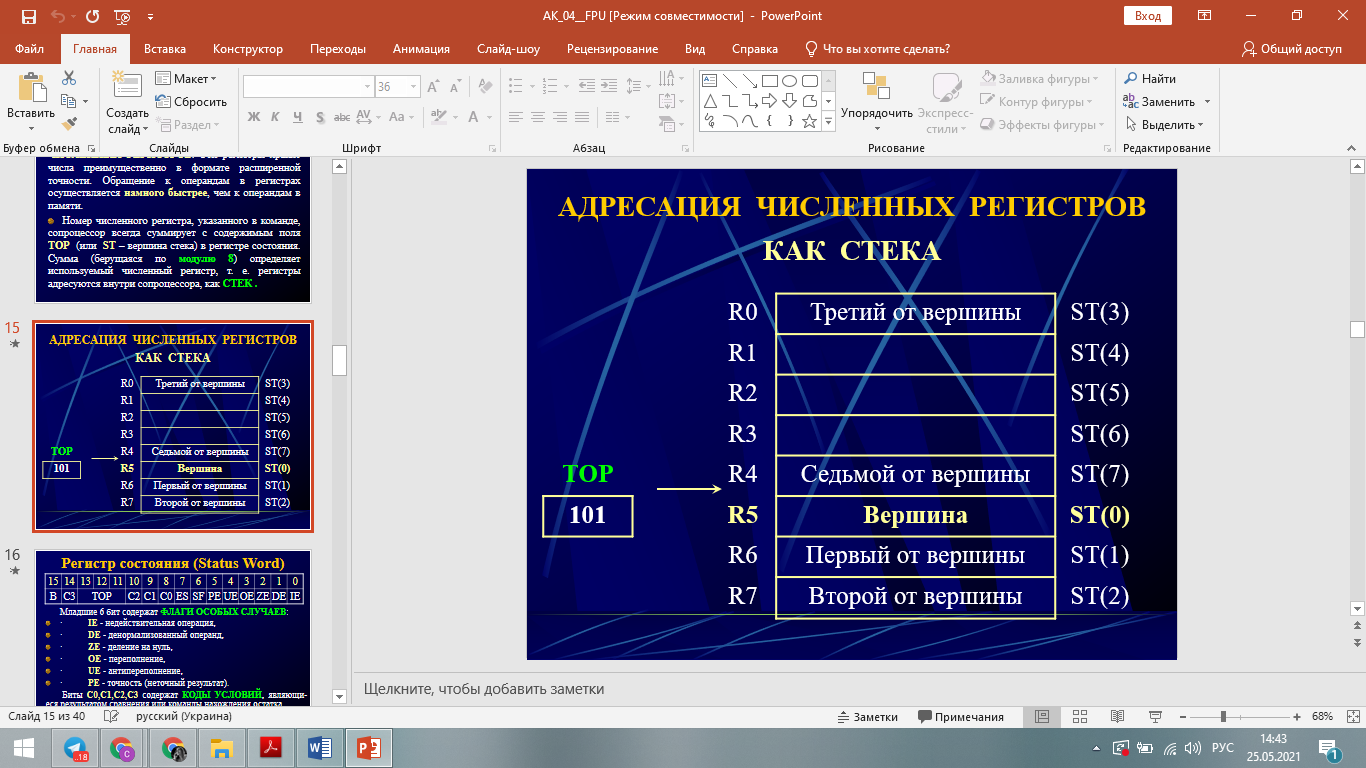


1. Программная модель сопроцессора. Адресация регистров.

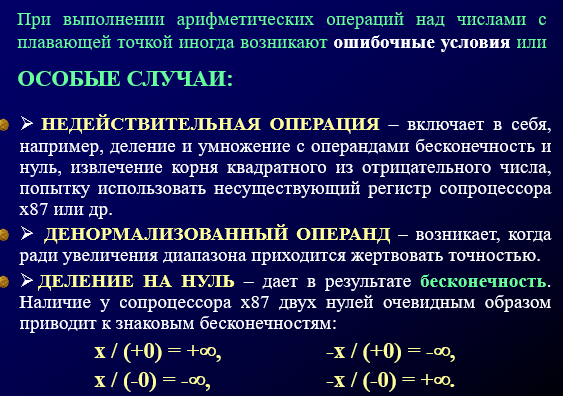
**Программная модель сопроцессора состоит из:**

* Стек сопроцессора (8 регистров по 10 бит) r0, r1, …, r7
* 3 служебных регистра по 16 бит (регистр состояния SWR, управляющий регистр CWR, регистр тегов TWR)
* 2 регистра указатели по 48 бит (DPR регистр указатель данных, IPR регистр указатель адреса)



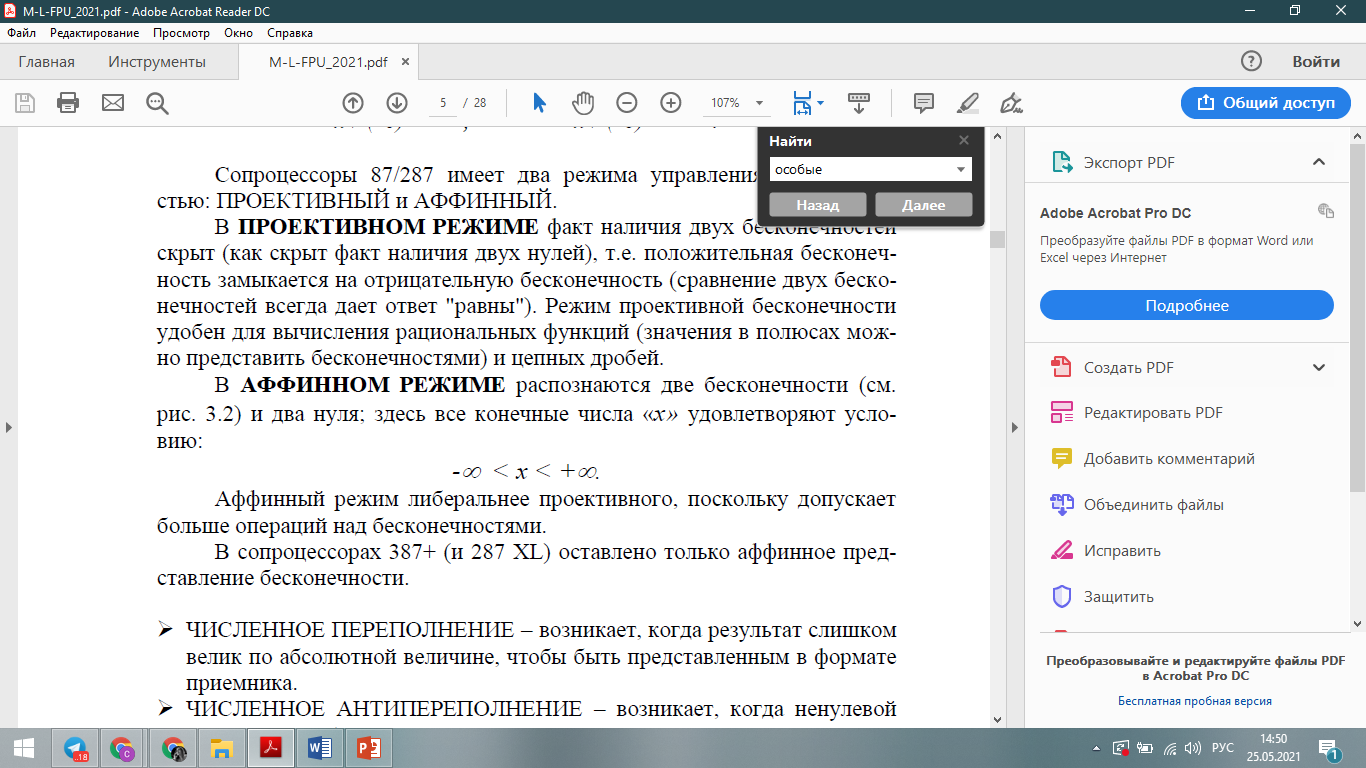


1. Особые случаи сопроцессора и их маскирование.

 Сопроцессоры 87/287 имеет два режима управления бесконечностью: ПРОЕКТИВНЫЙ и АФФИННЫЙ.

В **ПРОЕКТИВНОМ РЕЖИМЕ** факт наличия двух бесконечностей скрыт (как скрыт факт наличия двух нулей), т.е. положительная бесконеч-ность замыкается на отрицательную бесконечность (сравнение двух беско-нечностей всегда дает ответ "равны"). Режим проективной бесконечности удобен для вычисления рациональных функций (значения в полюсах мож-но представить бесконечностями) и цепных дробей.

В **АФФИННОМ РЕЖИМЕ** распознаются две бесконечности (см. рис. 3.2) и два нуля; здесь все конечные числа «*х»* удовлетворяют усло-вию:



Аффинный режим либеральнее проективного, поскольку допускает больше операций над бесконечностями.

В сопроцессорах 387+ (и 287 XL) оставлено только аффинное пред-ставление бесконечности.

ЧИСЛЕННОЕ ПЕРЕПОЛНЕНИЕ – возникает, когда результат слишком велик по абсолютной величине, чтобы быть представленным в формате приемника.

ЧИСЛЕННОЕ АНТИПЕРЕПОЛНЕНИЕ – возникает, когда ненулевой результат по абсолютной величине слишком мал для представления, т.е. когда он слишком близок к нулю.

НЕТОЧНЫЙ РЕЗУЛЬТАТ – возникает, когда результат операции не-возможно точно представить в формате приемника. Например, при де-лении 1.0 на 3.0 получается бесконечная дробь, которую невозможно точно представить ни в одном формате. Если особый случай неточного результата замаскирован, сопроцессор х87 округляет результат до обычного числа с плавающей точкой.

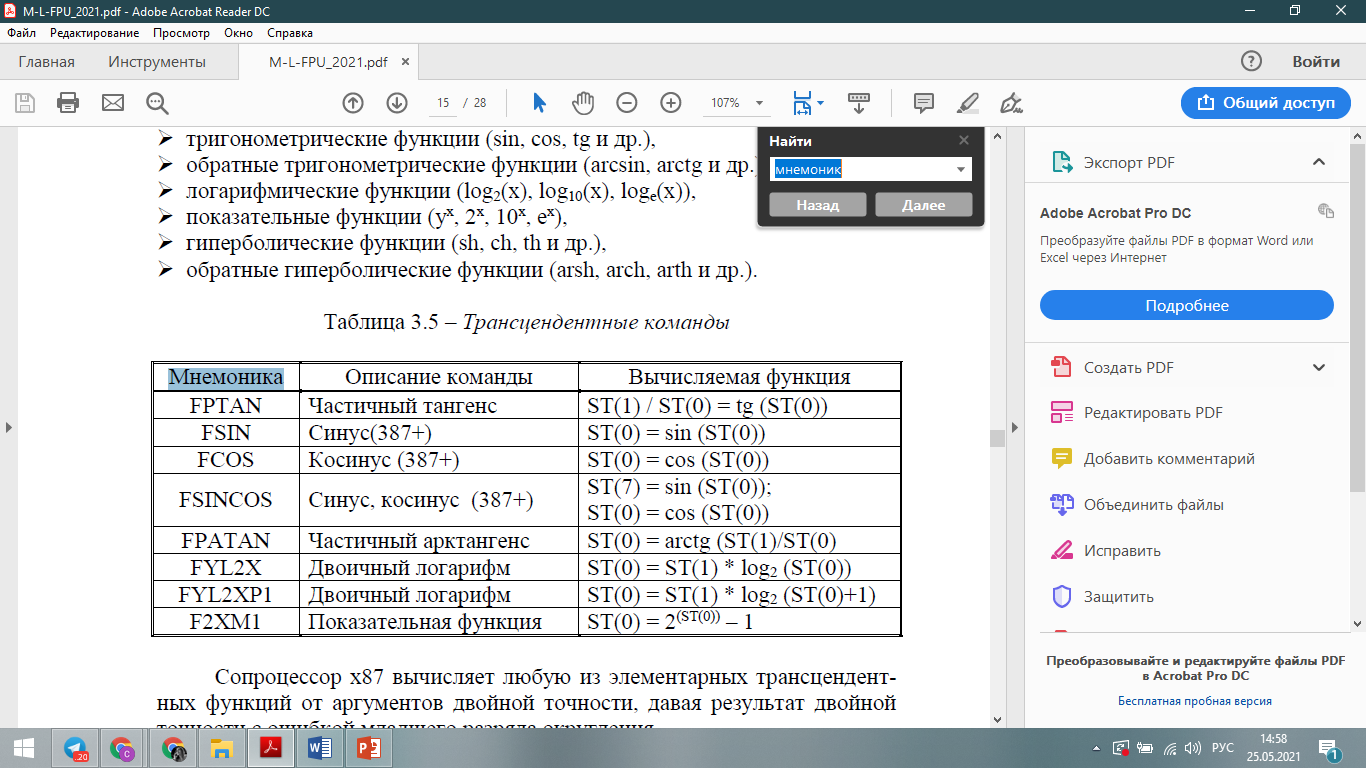
1. Кодирование мнемоник сопроцессора.

Форматы команд сопроцессора х87 аналогичны форматам команд х86. Ассемблерные мнемоники команд сопроцессора начинаются с буквы **F** (Floating) и их легко обнаружить, т.к. таких мнемоник у процессора х86 нет.

Мнемоники команд с целочисленным (двоичным) операндом начи-наются с «FI», а с десятичным операндом – с «FB».

Для некоммутативных команд вычитания и деления имеются обрат-ные варианты команд (в конце мнемоники добавляется буква R – reveres):

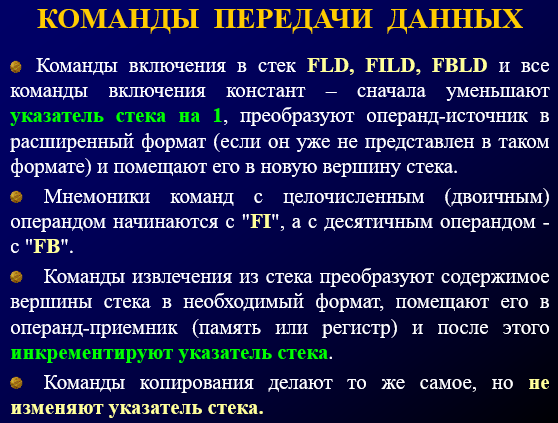
ПРИЕМНИК  ИСТОЧНИК **$** ПРИЕМНИК.



Команды управления сопроцессором х87 обеспечивают доступ к не- числовым регистрам. Мнемоники, которые начинаются с FN, соответству- ют командам «БЕЗ ОЖИДАНИЯ», т.е. процессор х86 передает их для вы- полнения в сопроцессор х87, не проверяя занятость сопроцессора и игно- рируя численные особые случаи.

Мнемоники без буквы «N» соответствуют командам «С ОЖИДАНИЕМ», т.е. заставляют процессор х86 реагировать на Внезамаски-рованные особые случаи и ожидать завершения выполнения команд в со-процессоре х87. В общем случае, программистам рекомендуется избегать форм команд «без ожидания».

1. Команды передачи данных сопроцессора.

Обозначения: (–) – декремент указателя стека до включения опе-ранда в вершину стека; (+) – инкремент указателя стека после извлечения операнда из вершины стека.

Команды извлечения из стека преобразуют содержимое вершины стека в необходимый формат, помещают его в операнд-приемник (память или регистр) и после этого инкрементируют указатель стека.

Команды копирования делают то же самое, но не изменяют указа-тель стека. Отсутствующую команду FBST можно реализовать двумя ко-мандами:

FLD ST(0) ; продублировать вершину стека с декрементом

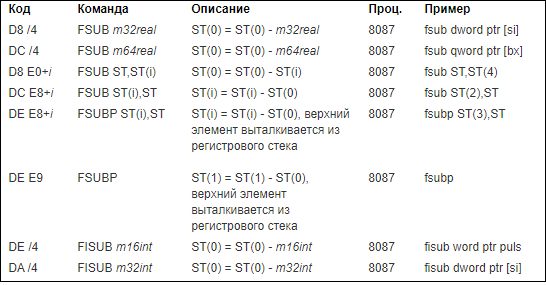
FBSTP ; извлечь из стека с инкрементом указателя.

При выполнении команд передачи данных может возникнуть необ-ходимость указать новую вершину стека. Команды FINCSTP и FDECSTP осуществляют соответственно инкремент и декремент указателя стека ST. Они не влияют на регистр тэгов и численные регистры.

1. Арифметические команды сопроцессора.



Имеется 6 форм базовых команд. Действия всех шести форм команд иллюстрируется на примере команды вычитания.



FSUB mem,

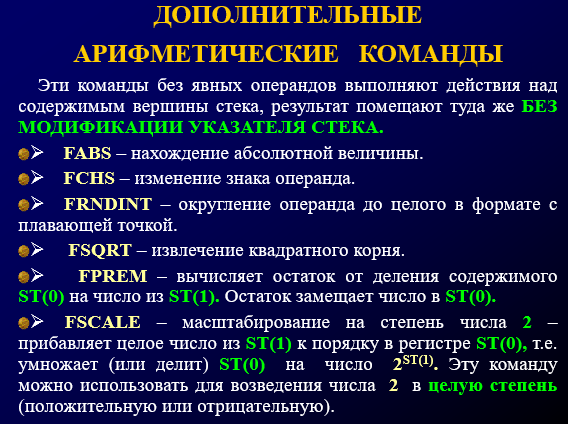
FISUB mem – адресуемый операнд в памяти является источником, а ре-гистр вершины стека ST(0) – приемником. Преобразование в расширен-ный формат с плавающей точкой осуществляется в процессе выполне-ния команды. УКАЗАТЕЛЬ СТЕКА НЕ МОДИФИЦИРУЕТСЯ.

FSUB ST, ST(i) – любой численный регистр ST(i) служит источником, а ST(0) – приемником. УКАЗАТЕЛЬ СТЕКА НЕ МОДИФИЦИРУЕТСЯ.

FSUB ST(i), ST – вершина стека является источником, а ST(i) – прием-ником. УКАЗАТЕЛЬ СТЕКА НЕ МОДИФИЦИРУЕТСЯ.

FSUBP ST(i), ST – вершина стека является источником, а ST(i) – прием-ником. По окончании операции источник ST(0) извлекается из стека с последующим ИНКРЕМЕНТОМ УКАЗАТЕЛЯ СТЕКА (рис. 3.11).

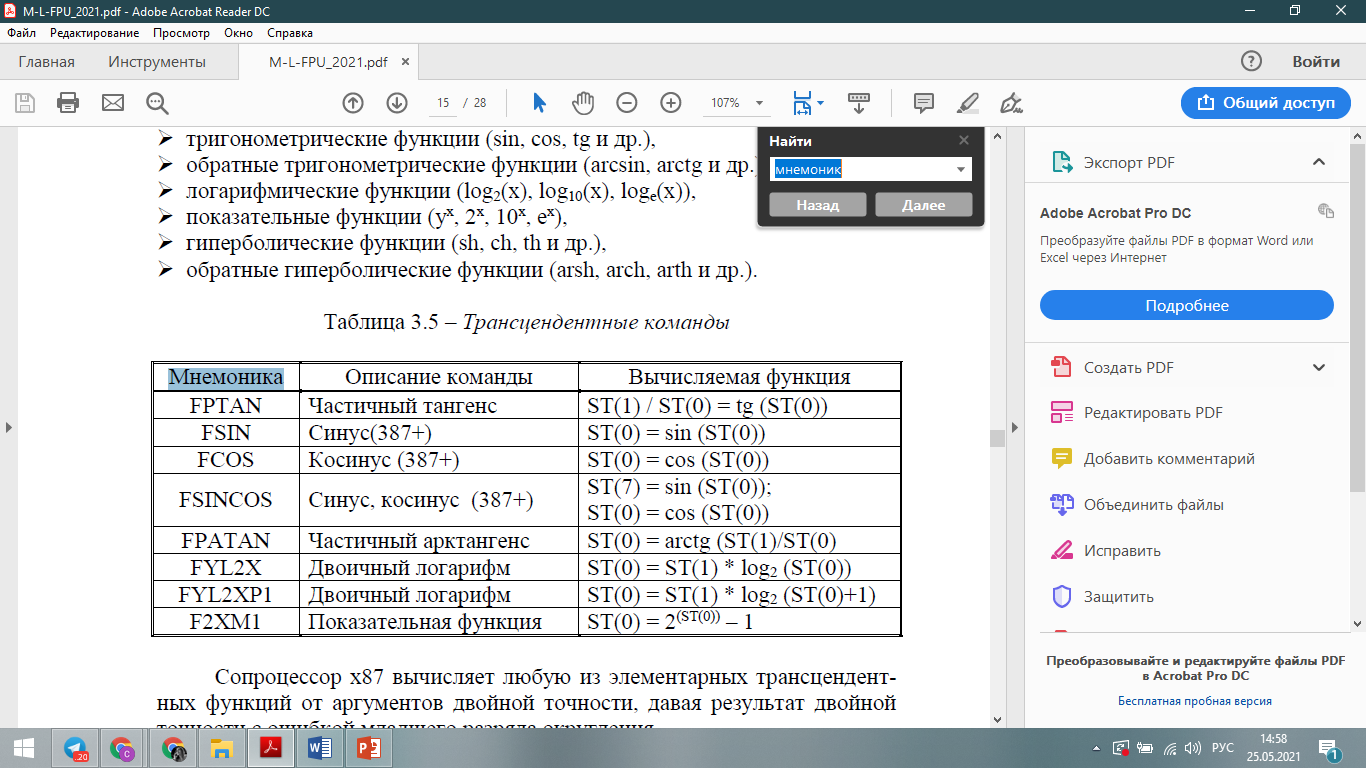
FSUB – (команда с классической стековой адресацией – использует только ST1, ST0) – извлекает из вершины стека источник (потом инкре-ментирует указатель стека), затем извлекает приемник (еще раз инкре-ментирует указатель стека), выполняет операцию и перед включением результата в стек декрементирует указатель. В итоге – ВЕРШИНА СТЕКА СДВИНУЛАСЬ В СТОРОНУ УВЕЛИЧЕНИЯ (рис. 3.12). По-следняя форма является частным случаем предыдущей.

****

1. Тригонометрические команды.

К элементарным трансцендентным функциям относятся:

* тригонометрические функции (sin, cos, tg и др.),
* обратные тригонометрические функции (arcsin, arctg и др.),
* логарифмические функции (log2(x), log10(x), loge(x)),
* показательные функции (yx, 2x, 10x, ex),
* гиперболические функции (sh, ch, th и др.),
* обратные гиперболические функции (arsh, arch, arth и др.).



Сопроцессор х87 вычисляет любую из элементарных трансцендент-ных функций от аргументов двойной точности, давая результат двойной точности с ошибкой младшего разряда округления.

Команда **FPTAN** нахождения частичного тангенса в качестве резуль-тата выдает два числа (сопроцессоры 87/287):

*y / x = tg (ST(0)).* Число «*y*» заменяет старое содержимое ST(0), а число «*x*» включает-ся сверху. Поэтому, после выполнения команды указатель стека уменьшится на 1, число «*х*» будет записано в новую вершину стека ST(0), а чис- ло «*y*» – в регистр ST(1).

Для получения значения тангенса необходимо выполнить команду FDIV. Две команды FPTAN и FDIV выбирают аргумент из вершины стека и туда же помещают значение **тангенса** (БЕЗ МОДИФИКАЦИИ УКАЗАТЕЛЯ ВЕРШИНЫ СТЕКА). Две команды FPTAN и FDIVR вычис- ляют значение **котангенса**.

Для команды FPTAN **аргумент задается в радианах** и должен нахо- дится в диапазоне (сопроцессоры 87/287):

0 <= ST(0) <= 1/4.

Для СОПРОЦЕССОРОВ 387+ аргумент команды FPTAN (**в радиа- нах**) может быть любым:

–263 <= ST(0) <= +264.

Значение тангенса исходного угла tg(ST(0)) замещает аргумент и в стек включается сверху 1,0 (для программной совместимости с предыду- щими сопроцессорами 87/287).

Значения остальных тригонометрических функций (для сопроцессо- ров 87/287) можно вычислить, используя формулы тангенса половинного угла (табл. 3.6). Поэтому перед началом вычисления тригонометрических функций с использованием команды FPTAN необходимо аргумент в ST(0) поделить на 2. Новое значение аргумента «z» должно также удовлетворять условию: 0  z  1/4.

В СОПРОЦЕССОРАХ 387+ появились новые команды:

FSIN – вычисление синуса;

FCOS – вычисление косинуса;

**FSINCOS** – вычисление синуса и косинуса.

Все они воспринимают в ST(0) исходный угол, **измеряемый в ради- анах** и находящийся в диапазоне: –263 <= ST(0) <= +263. Команды FSIN и

FCOS возвращают результат на место аргумента, а команда FSINCOS воз- вращает значение синуса на место аргумента и включает значение косину- са в стек.

Команда **FPATAN** вычисляет *arctg (ST(1)/ST(0))*. Два операнда из- влекаются из стека, а результат включается в стек. Поэтому окончательно, УКАЗАТЕЛЬ СТЕКА УВЕЛИЧИВАЕТСЯ НА 1. Операнды этой команды для сопроцессоров 8087/287 должны удовлетворять условию:

0 < ST(1) < ST(0).

В сопроцессорах 387+ ограничений на диапазон допустимых аргу- ментов команды FPATAN не существует.

Для вычисления остальных обратных тригонометрических функций по аргументу «*z*» необходимо предварительно подготовить операнды в ST(0) и ST(1) в соответствии с табл. 3.7 (***делить операнды не нужно***).

1. Логарифмические и показательные команды.

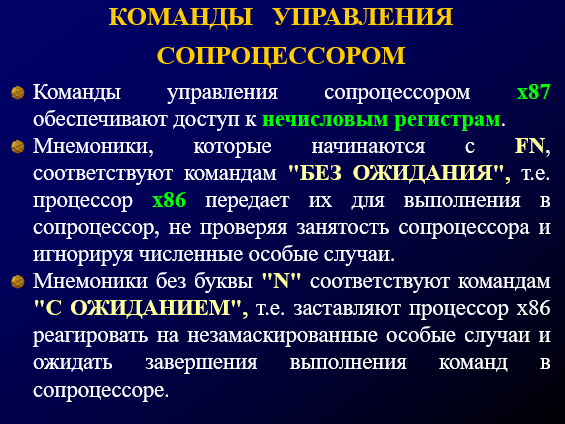
Команда **FYL2X** вычисляет функцию: *Y = ST(1) \* log2 ST(0).* Два операнда извлекаются из стека, а затем результат включается в стек. По- этому УКАЗАТЕЛЬ СТЕКА УВЕЛИЧИТСЯ НА 1. В команде требуется удовлетворение естественного для логарифмической функции условия:

ST(0) > 0 .

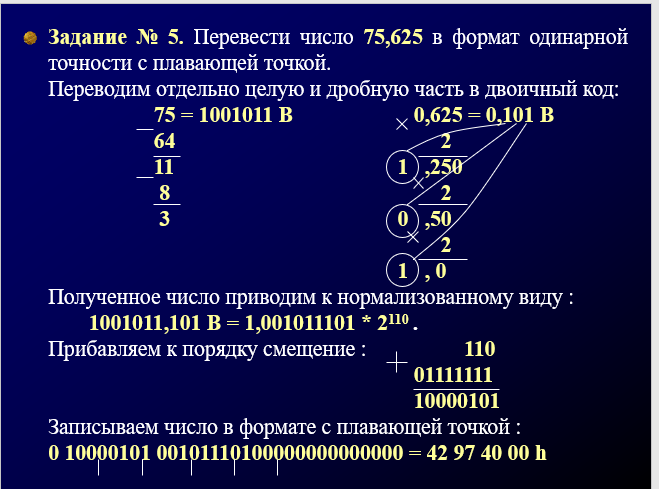
Значения других логарифмических функций вычисляются по форму- лам в табл. 3.8 с загрузкой в регистр ST(1) необходимых констант коман- дами: FLDLN2 и FLDLG2.

Еще одна логарифмическая команда **FYL2XP1** вычисляет функцию: *Y = ST(1) \* log2 (ST(0) + 1).* Причина появления этой команды заключа-ется в получении более высокой точности вычисления функции: *log(1 + x).* Эта функция часто встречается в финансовых расчетах, а также при вычислении обратных гиперболических функций.

1. Команды управления сопроцессором.

10. Перевести число 75,125 в формат двойной точности с плавающей точкой.



11. Перевести число с плавающей точкой C0 77 76 00 00 00 00 00 h в при-вычный десятичный формат.

C0 77 76 00 00 00 00 00

1 1000 0000 1110 1110 1110 1100 0000 0000

1000 0000

1110 1110 1110 1100 \* 2^1 = 1110 1110 1110 1100 \* 2^1

11,110 1110 1110 1100 = 3,28396

с0126b28

1100 0000 0001 0010 0110 1011 0010 1000

1 1000 0000 0010 0100 1101 0110 0101 000

0010 0100 1101 0110 0101 000\*2^1

10,010 0100 1101 0110 0101 000

-2,

-2.28779030