- Регистровая модель процессора -

Регистровую модель процессора образуют программно доступные регистры. Для процессора P6 существует две группы регистров:

- прикладные регистры (пользовательские), используемые процессором при выполнении прикладных (пользовательских) программ. В эту группу входят основные функциональные регистры, регистры обработки чисел, (FP и MMX технология) и пакетов чисел с плавающей точкой (SSE технология).

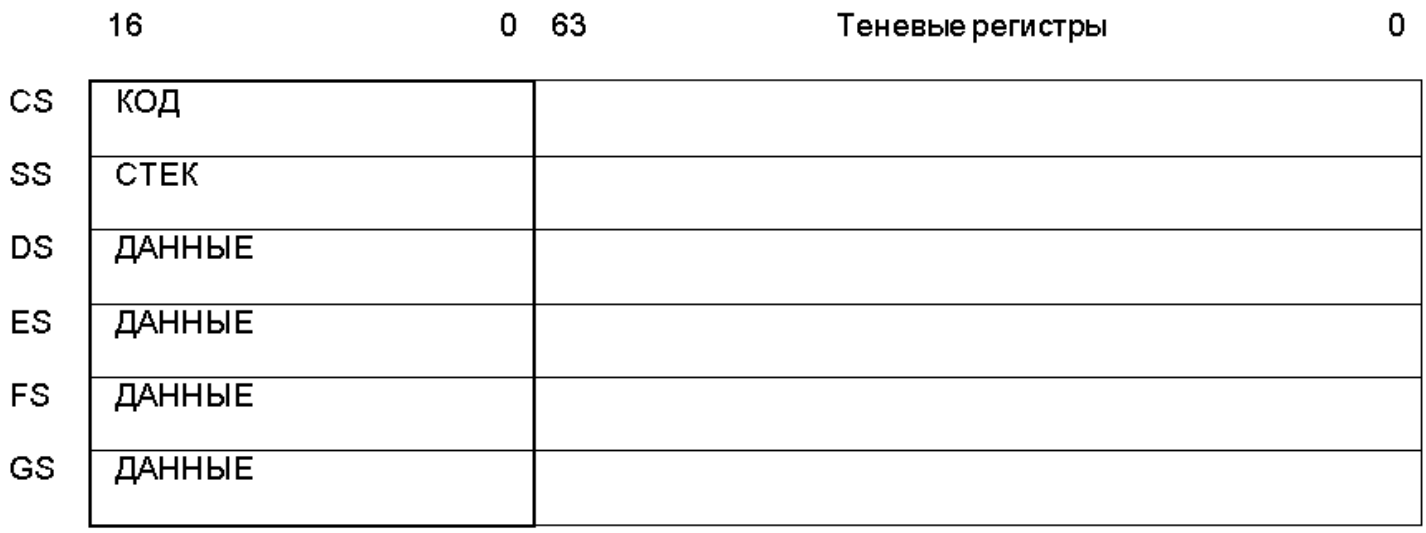
- системные и служебные регистры, используемые при выполнении системных программ, отладке, тестировании процессора и контроле эффективности выполнения программ. Регистры этой группы доступны только программам с высшим уровнем привилегий.

Основные функциональные регистры

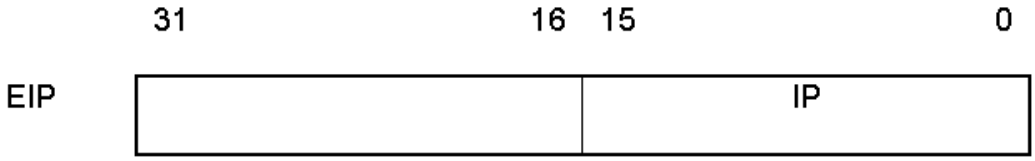
Регистры общего назначения.

	31 16	15		0
EAX		(AH)	AX	(AL)
EBX		(BH)	BX	(BL)
ECX		(CH)	CX	(CL)
EDX		(DH)	DX	(DL)
ESI			SI	
EDI			DI	
EBP			BP	
ESP			SP	

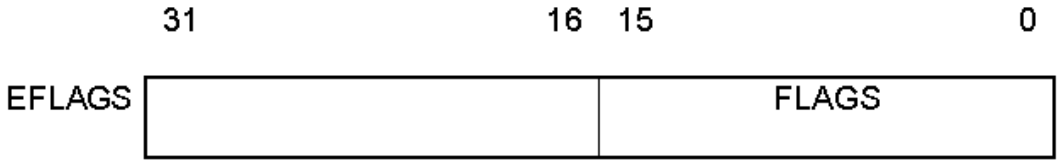
Регистры сегментов.



Указатель команд.



Указатель флагов.



Назначение регистров.

Регистры общего назначения применяются для хранения данных и\или адресов.

Арифметические регистры **EAX,EBX,ECX,EDX** и их 16 и 8 разрядные аналоги позволяют процессору выполнять операции над двойными словами, словами и байтами.

EAX/AX/AL (Accumulator) – аккумулятор. Применяется для хранения промежуточных данных. В операции умножения/деления аккумулятор содержит множимое/делимое до выполнения операции и произведение/частное после выполнения операции. В командах ввода/вывода аккумулятор хранит вводимые и

выводимые данные. Вся десятичная арифметика выполняется только с участием аккумулятора.

EBX/BX/BL (Base) – базовый регистр. Применяется для указания базового (начального) адреса объекта данных в памяти.

ECX/CX/CL (Counter) – счетчик. Участвует в некоторых командах, которые производят повторяющиеся операции, например сдвиги или манипуляции цепочками.

EDX/DX (Data) – регистр данных. Наиболее часто применяется для хранения промежуточных данных, а также в командах умножения/деления совместно с аккумулятором. Кроме того в командах ввода/вывода в регистре DX хранится адрес порта, к которому производится обращение.

ESI/SI (Source Index) – индекс источника. Выполняет функцию регистра адреса, в частности при производстве цепочечных операция адресует элемент цепочки источника.

EDI/DI (Destination Index) – индекс получателя. Выполняет функцию регистра адреса. В частности при производстве цепочечных операций адресует элемент цепочки получателя.

EBP/BP (Base Pointer) – указатель базы. Предназначен для удобного доступа к объектам данных, находящихся в стеке, например объектах подпрограмм.

ESP/SP (Stack Pointer) – указатель стека. Неявно используется в командах PUSH (включения) и POP (исключения), а также в других стековых операциях. ESP\SP адресует вершину стека в текущем сегменте стека.

EIP/IP (Instruction Pointer) – регистр указателя команды. Предназначен для адресации команд внутри текущего сегмента кода.

EFLAGS/FLAFS – регистр флагов. Содержит флаги состояния и управления.

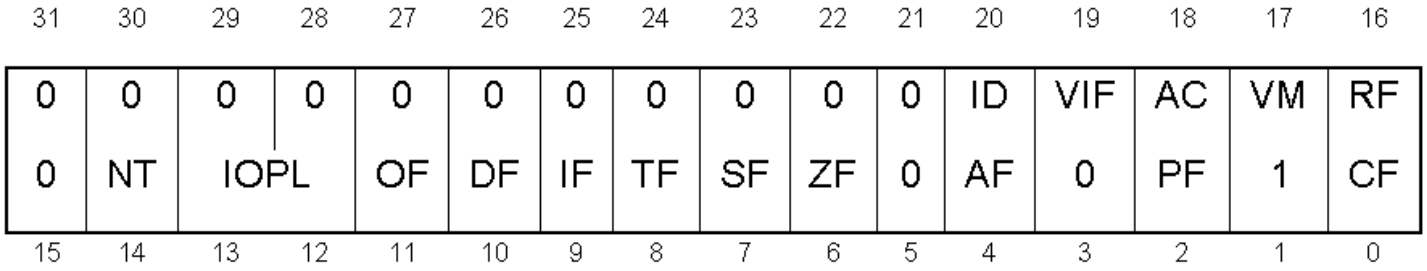
Сегментные регистры идентифицируют текущие сегменты, к которым может обращаться текущая выполняемая программа.

Регистр флагов процессора P6.

Этот регистр содержит две группы флагов:

- флаги состояния

- флаги управления



Флаги результата сообщают об особенностях результата обработки данных

CF (Carry Flag) – флаг переноса. Устанавливается, если арифметическая операция вызвала перенос (при сложении) или заем (при вычитании) из старшего бита результата.

PF (Parity Flag) – флаг паритета. Устанавливается, если младшие восемь бит результата содержит четное число единиц.

AF (Auxiliary carry Flag) – флаг вспомогательного переноса. Устанавливается, если арифметическая операция вызвала перенос (при сложении) или заем (при вычитании) из младшей тетрады результата.

ZF (Zero Flag) – флаг нуля. Устанавливается если результат операции равен нулю.

SF (Sign Flag) – флаг знака. Устанавливается, если результат отрицательный.

OF (Overflow Flag) – флаг переполнения. Устанавливается если происходит переполнение разрядной сетки.

IOPL (Input\Output Privilege Level) – уровень привилегий ввода-вывода. Показывает минимальный уровень привилегий выполняющейся задачи, на котором разрешаются выполнение операций ввода-вывода.

NT (Nested Task) – флаг вложенной задачи. Устанавливается когда текущая задача производит переключение на другую задачу с помощью команды CALL.

Флаги управления регистра EFLAGS воздействуют на те или иные аспекты работы процессора.

TF (Trap Flag) – флаг трассировки. При установленном значении процессор переводится в режим пошаговой работы.

IF (Interrupt Flag) – флаг прерывания. При установленном значении, процессор распознает и обрабатывает внешние аппаратные прерывания.

DF (Direction Flag) – флаг направления. Задает направление обработки цепочек. При DF=0 цепочки обрабатываются слева направо. При DF=1 – справа налево.

RF (Resume Flag) – флаг возобновления. Этот флаг временно запрещает особые случаи отладки, чтобы осуществить рестарт команды после особого случая отладки, без немедленного формирования еще одного особого случая отладки.

VM (Virtual Mode) – флаг виртуального режима. Установка этого флага приводит процессор в режим виртуального процессора 8086.

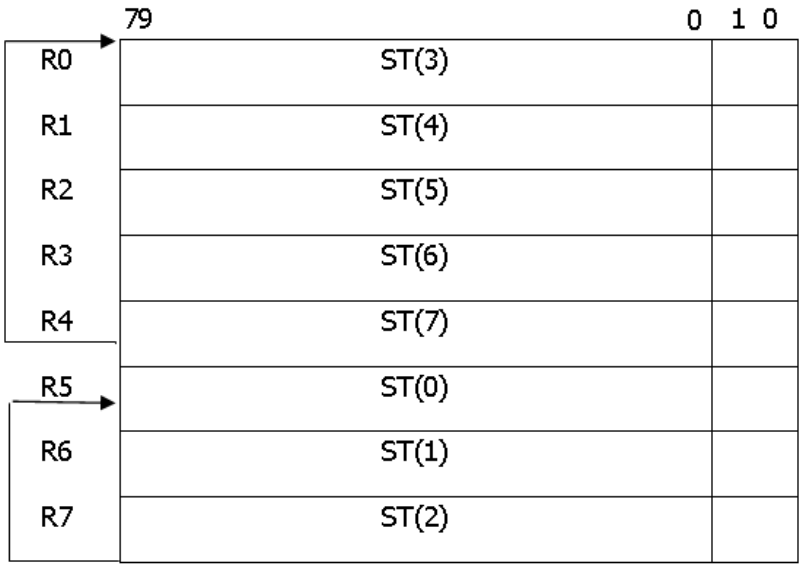
AC (Alignment Check) – флаг контроля выравнивания. Совместно с флагом AM в регистре CR0 разрешает контроль выравнивания при обращении к памяти.

VIF (Virtual Interrupt Flag) – виртуальный флаг прерывания. Виртуальный образ флага IF. Используется совместно с флагом VIP. Процессор распознает VIF, если бит VME=1 в CR4 или бит PVI=1 в CR4 (разрешено расширение виртуального режима) и IOPL<3.

VIP (Virtual Interrupt Pending Flag) – виртуальный флаг задержки прерывания. Системное ПО устанавливает этот флаг, если требуется отложить обработку прерывания. Флаг распознается, если разрешено расширение виртуального режима.

ID (Identification Flag) – флаг поддержки инструкции CPUID.

Регистры FPU.



Регистр состояния.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPSR	B	C3	T	O	P	C2	C1	C0	ES	SF	PE	UE	OE	ZE	DE	IE

Регистр управления.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPCR	x	x	x	IC	R	C	P	C	x	x	PM	UM	OM	ZM	DM	IM

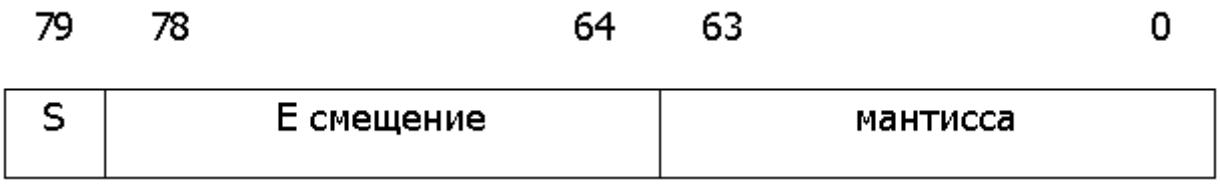
Регистр тегов.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TW	TAG 7		TAG 6		TAG 5		TAG 4		TAG 3		TAG 2		TAG 1		TAG 0	

Регистры указателей команд и данных.

	15	8	7	0
FIR|FDR	FIR		FID	

Формат регистров данных.



Формат тегов

00 – действительное число.

01 – нуль.

10 – недействительное число (например, бесконечность).

11 – незаполненный регистр.

FPSR (FP State Register) – регистр состояния FPU.

Признаки ошибок.

IE – недействительная операция.

DE – денормализованный операнд.

ZE – деление на ноль.

OE – переполнение.

UE – антипереполнение.

PE – нарушение точности.

Другие признаки.

SF – признак переполнения стека.

При этом C1 = 1 – выход за верхнюю границу.

C1 = 0 – выход за нижнюю границу.

ES – общий признак ошибки. Устанавливается, если устанавливается один из признаков ошибки.

Признаки результата выполненной операции.

C0 – перенос.

C1 – вспомогательный перенос.

C2 – ноль.

C3 – знак.

TOP – указывает на вершину стека.

B – бит занятости. Дублирует бит ES.

FPCR (FP Control Register) – регистр управления FPU.

Биты **PM, UM, OM, ZM, DM, IM** предназначены для маскирования признаков ошибок.

IC – управляет бесконечностью.

0 – проективный режим;

1 – афиктивный режим.

RC – определяет режим округления:

00 – к ближайшему числу;

01 – к -∞;

10 – к +∞;

11 – к нулю.

PC – задает точность представления результатов арифметических операций FPU.

00 – одинарная точность; (23 разряда мантисса, 8 – порядок)

01 – не используется;

10 – двойная точность; (52 разряда мантисса, 11 – порядок)

11 – расширенная точность; (64 разряда мантисса, 15 – порядок)

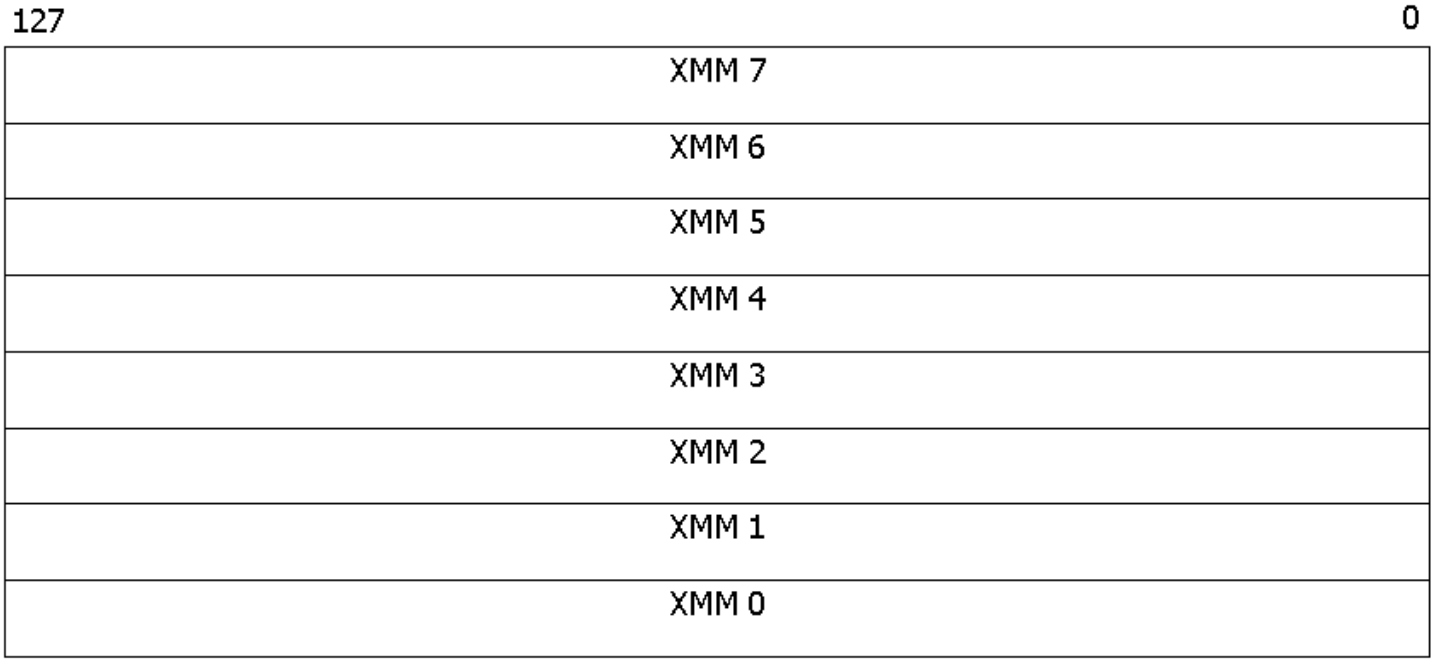
FIR – регистр-указатель команды.

FDR – регистр-указатель данных.

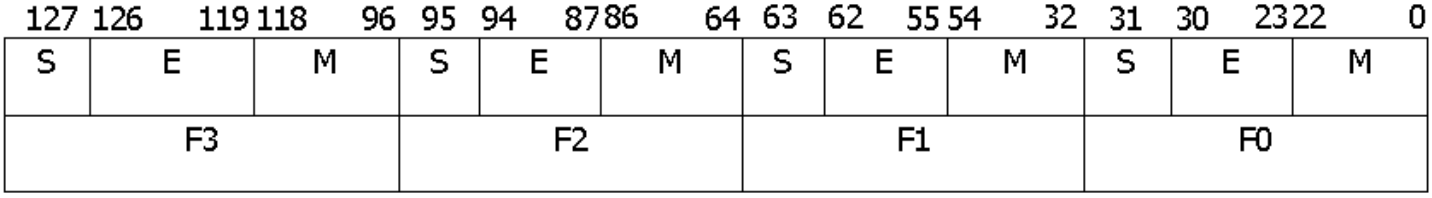
FIR и FDR предназначены для идентификации команды, вызвавшей ошибку.

Регистры для реализации SSE технологии.

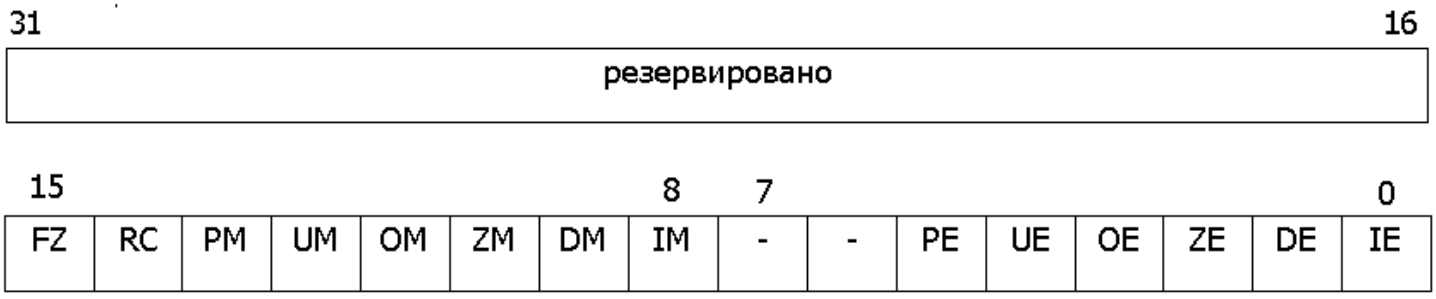
Предназначены для пакетной обработки чисел с плавающей точкой.



Формат регистра данных.



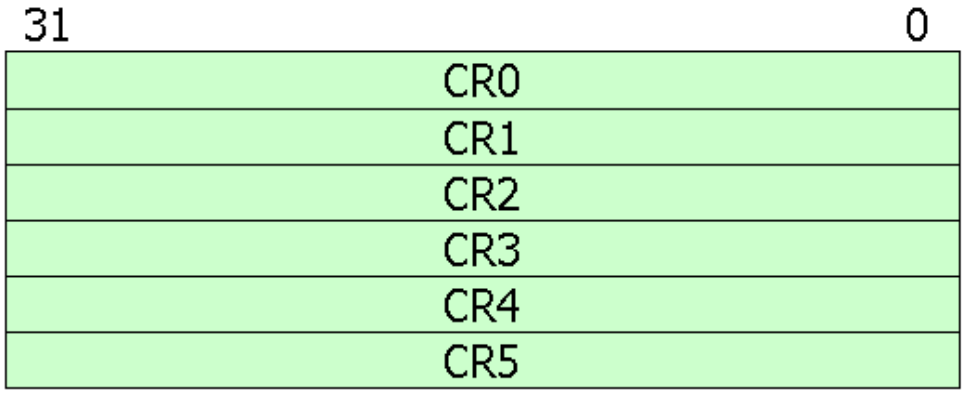
Регистр управления-состояния MXCSR.



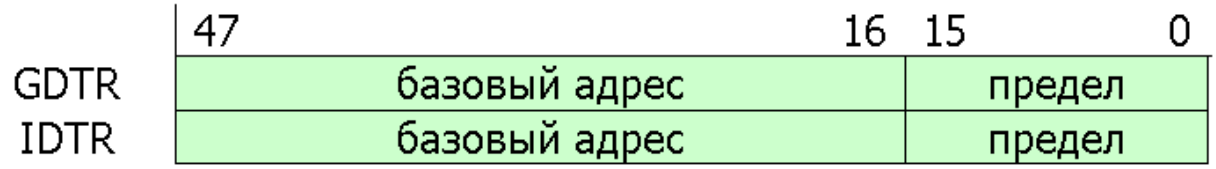
Биты регистра MXCSR имеют такое же назначение, как соответствующие биты в регистрах FPCR блока FPU.

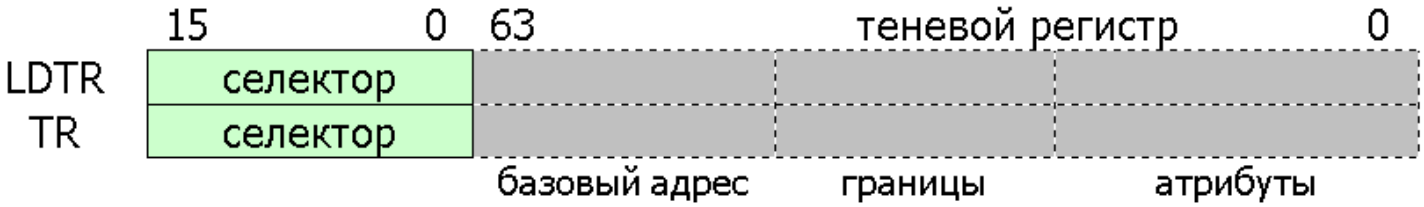
Системные регистры

Регистры управления

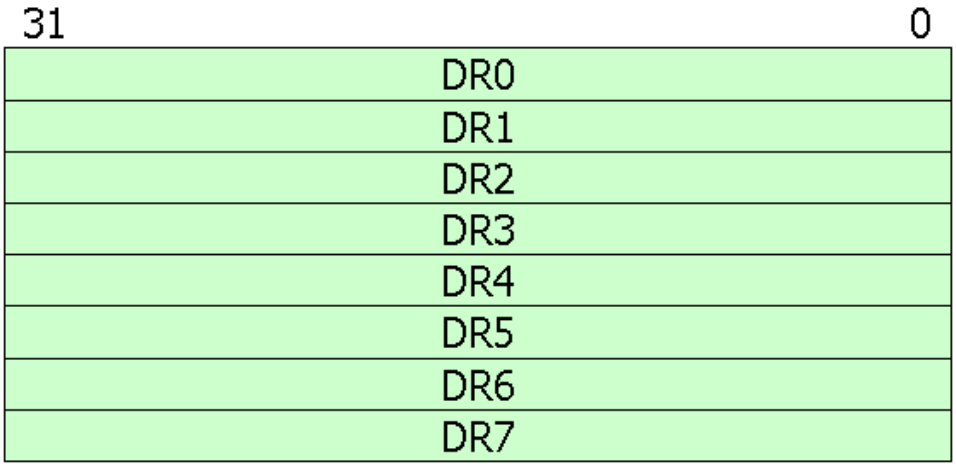


Регистры системных адресов





Регистры отладки



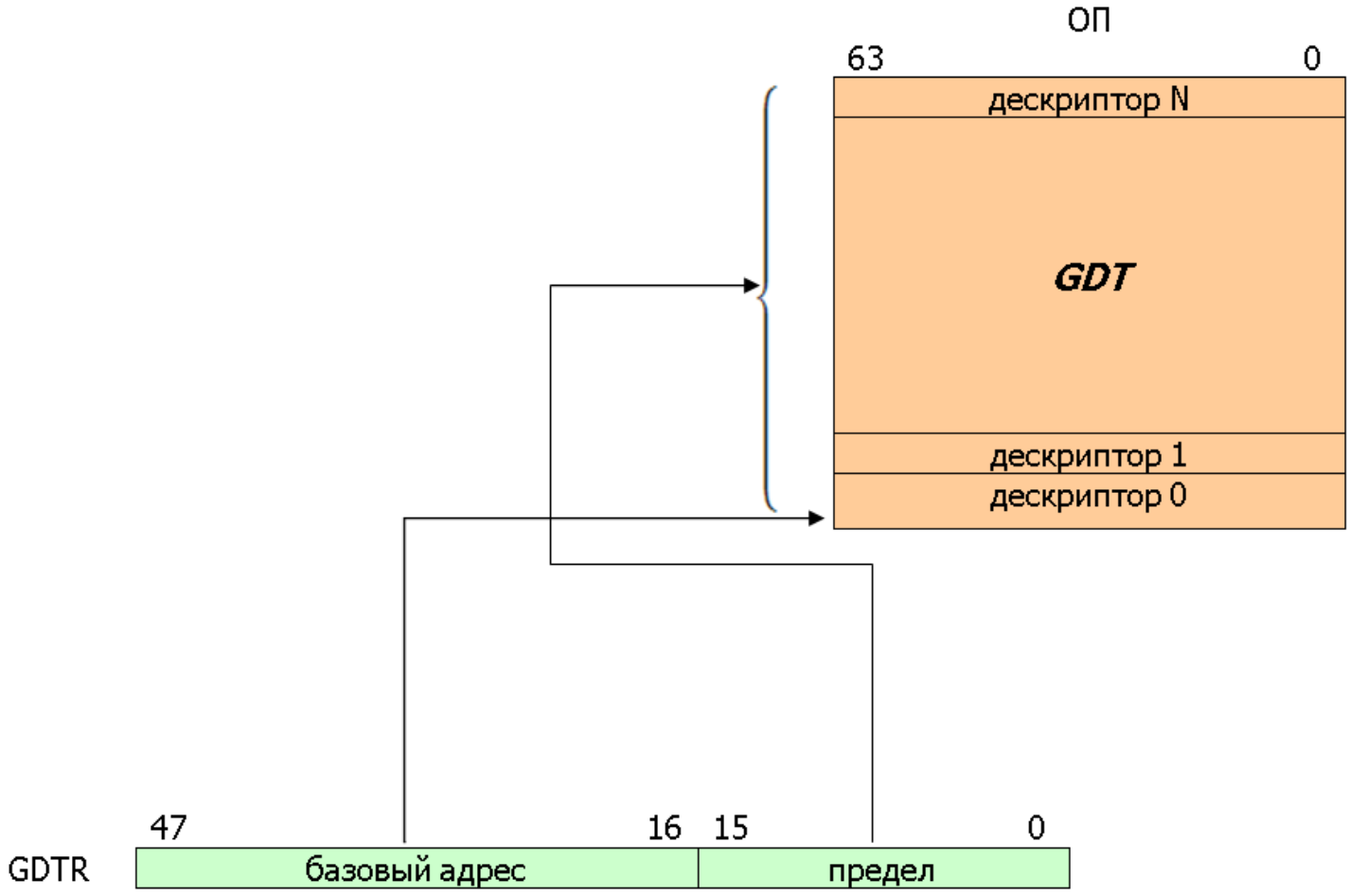
Модельно специфические регистры

Регистр GDTR

GDTR (Global Descriptor Table Register)

Определяет местоположение в ОП глобальной дескрипторной таблицы.

Таблица GDT содержит дескрипторы общесистемных сегментов.

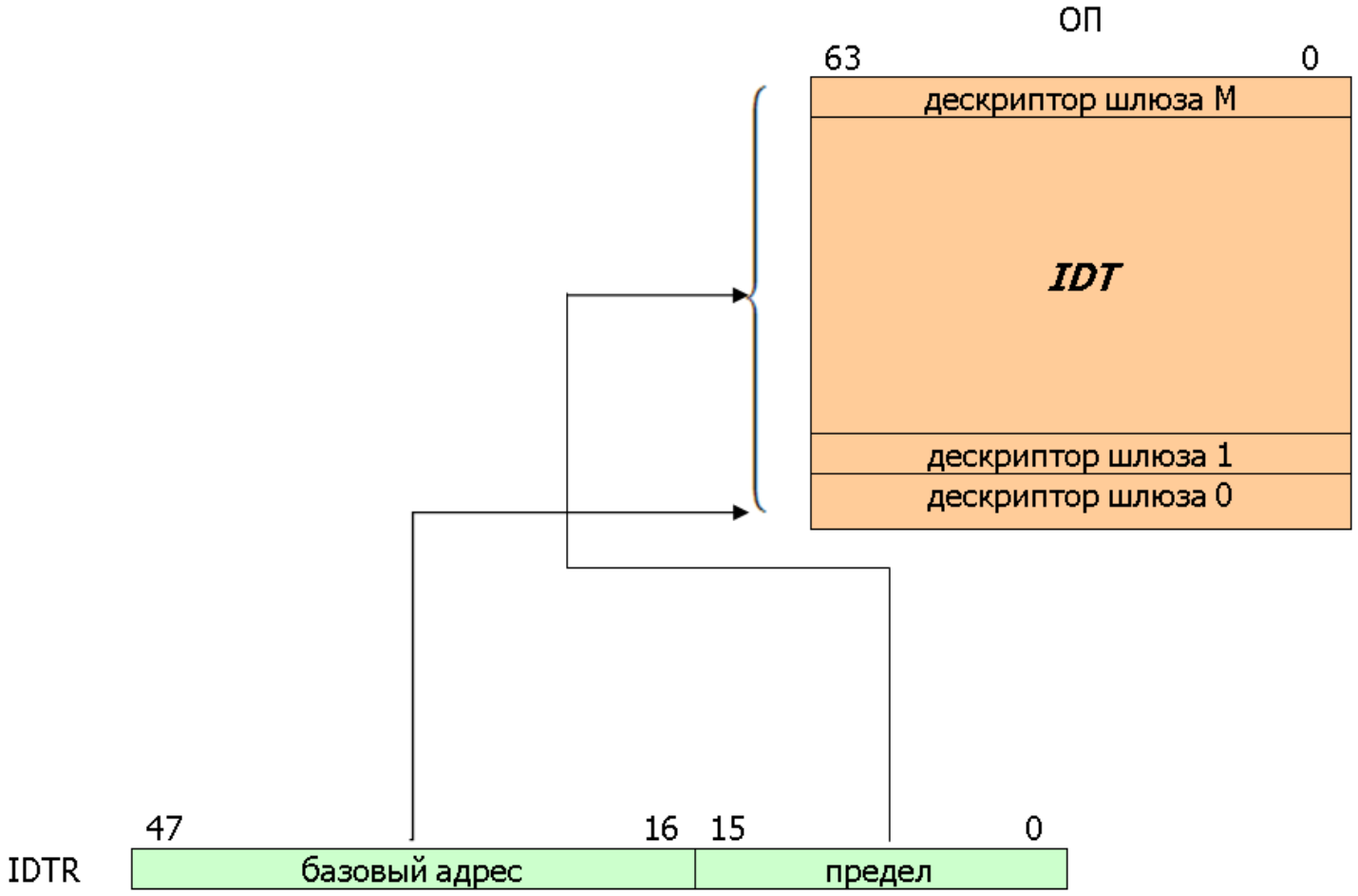


Регистр IDTR

IDTR (Interrupt Descriptor Table Register).

Определяет местоположение в ОП дескрипторной таблицы прерываний.

Таблица IDT содержит дескрипторы шлюзов, необходимые для выполнения процедур обработки прерываний (исключений).

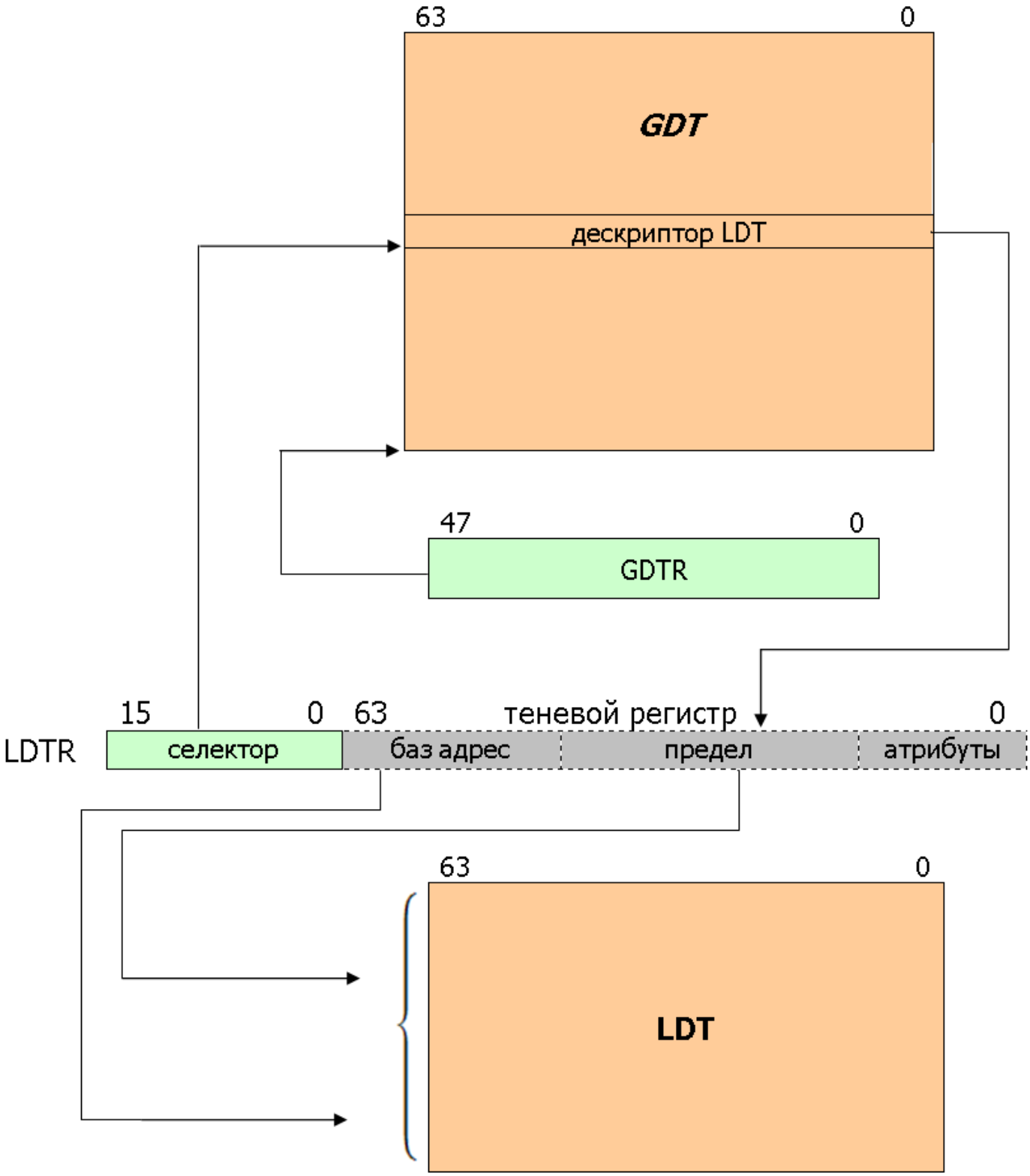


Регистр LDTR

Определяет косвенно через GDT местоположение в оперативной памяти LDT.

Локальная дескрипторная таблица содержит дескрипторы сегментов(кода, стека, данных), относящиеся к текущей задаче.

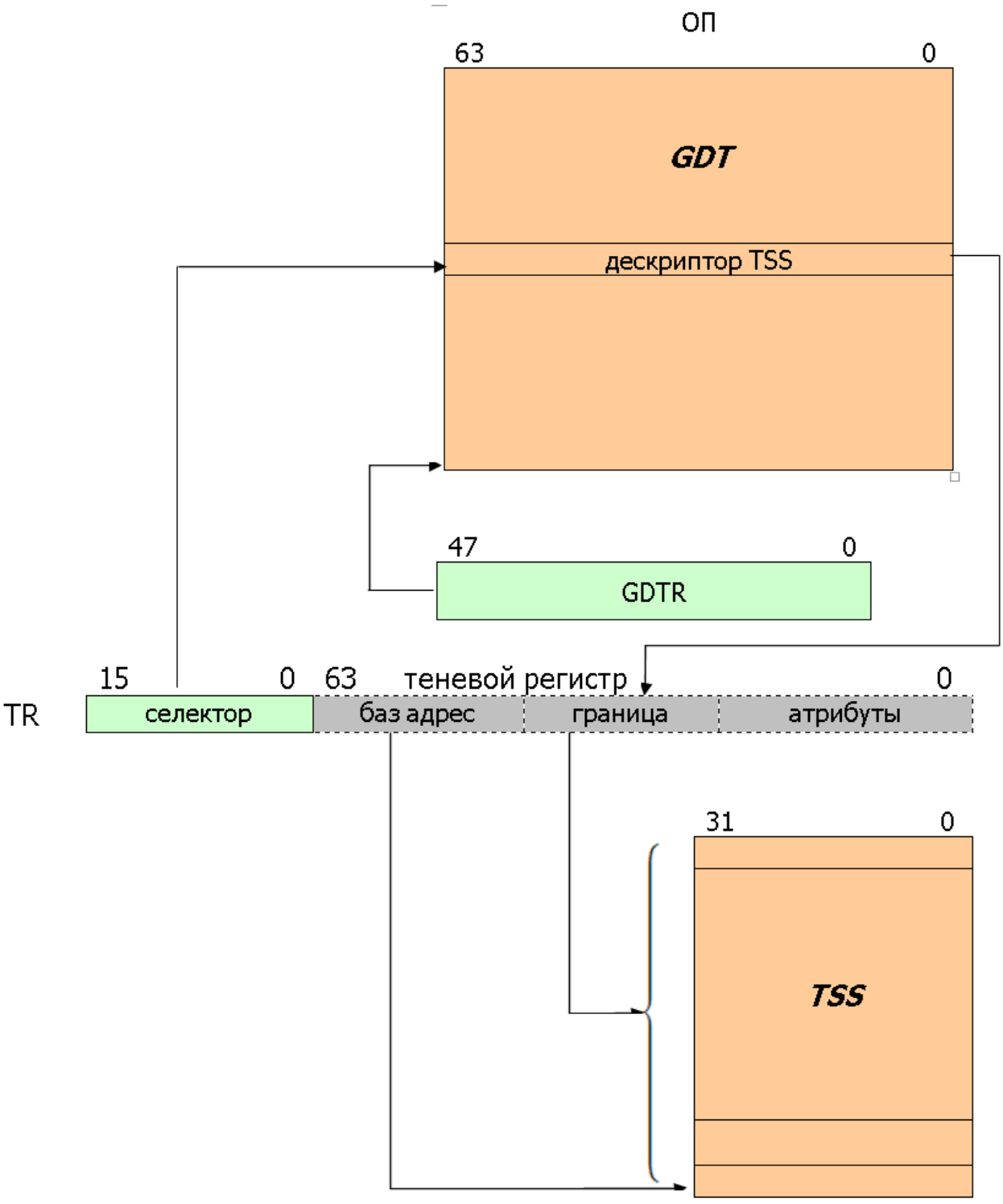
Каждая задача имеет свою LDT.


63
0
GDT
дескриптор LDT
47
0
GDTR
15
0
63
теневой регистр
0
LDTR
селектор
баз адрес
предел
атрибуты
63
0
LDT

Регистр TR

TR (Task Register) определяет косвенно через GDT местоположение сегмента состояния задачи TSS (Task Segment State). TSS предна значен для сохранения контекста задачи.

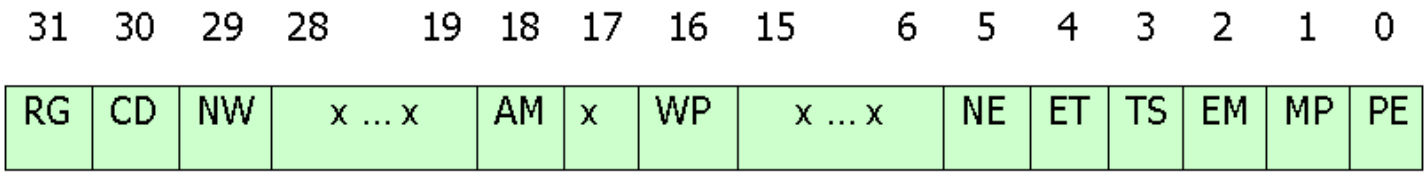
Каждая задача имеет свой TSS.



Регистр управления CR0

CR0 содержит флаги, которые задают режим работы процессора и его блоков и отражают состояние процессора.

Младшие 16 разрядов CR0 образуют слово состояния машины MSW (machine status word)



PE (protect enable) – управляет режимом работы процессора (1-защищенный режим, 0-реальный режим).

MP (math present) – показывает присутствие устройства FPU.

EM (emulation) – бит эмуляции. Когда EM=1 или TS=1 выполнение команды FPU генерирует особый случай недоступного процессора.

TS (task switched) – бит переключения задачи.

ET (extension type) – тип расширения. Этот бит совместно с другими показывает поддержку команд FPU.

NE (numeric error) – численная ошибка.

WP (write protect) – бит защиты от записи. Этот бит защищает от записи страницы пользователя от обращений супервизора.

AM (alignment mask) – маска выравнивания. Этот бит разрешает контроль выравнивания. Контроль производится, когда AM=1 AC=1b EFLAGS и CPL=3.

NW (not write through) – несквозная запись. Этот бит определяет режим работы внутреннего КЭШа.

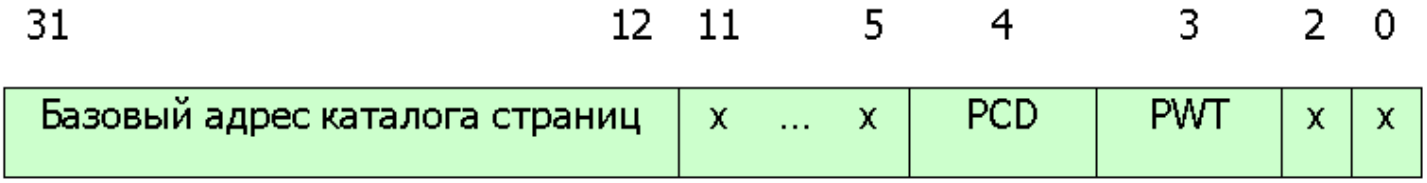
PG (paging) – страничное преобразование. Этот бит разрешает страничное преобразование адреса.

Регистр управления CR2

CR2 содержит 32 битный линейный адрес, при обращении к которому произошло страничное нарушение.

Регистр управления CR3

CR3 используется для управления страничным преобразованием адреса.

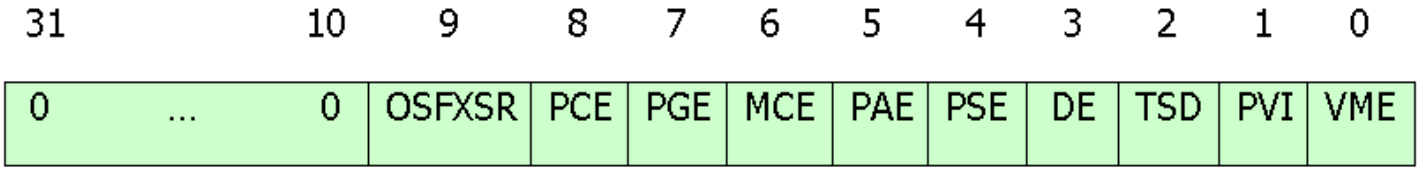


PCD (page cash disable) – бит запрещения кэширования страницы. Этот бит управляет кэшированием во внешней кэш-памяти.

PWT (page write through) – бит сквозной записи. Этот бит управляет режимом кэширования. (1 – сквозная запись, 0-обратная запись).

Регистр управления CR4

CR4 содержит биты, расширяющие возможность процессоров P6.



VME – бит разрешения виртуальных прерываний в защищенном режиме.

TSD – бит разрешения чтения таймера реального времени только программам с максимальным уровнем приоритета.

PSE – бит расширения размера адресуемых страниц до 4 мб.

PAE – бит расширения разрядности физического адреса до 36 бит.

Регистры отладки

DR7	регистр управления отладкой
DR6	регистр состояния отладки
DR5	зарезервирован
DR4	зарезервирован
DR3	линейный адрес КТ3
DR2	линейный адрес КТ2
DR1	линейный адрес КТ1
DR0	линейный адрес КТ0

Модельно-специфические регистры (MSR)

Регистры MSR (Model Special Registers) предназначены для управления процессом отладки, мониторингом производительности, машинным контролем, кэшированием областей физической памяти и другими функциями.

Типы команд процессора P6.

Набор команд процессора P6 обеспечивает выполнение операций над 8,16,32 разрядными операндами. Набор включает:

– безадресные команды, в которых расположение операндов задается мнемоникой команды, Например AAA – десятичная коррекция результата сложения;

– одноадресные команды, для которых исходный операнд и результат может находится в регистре, памяти; inc ax – инкремент регистра ax.

– двухадресные команды: регистр-регистр, регистр-память, память-регистр, непосредственный операнд-регистр, непосредственный операнд-память.

mov ax, 10h

Форматы команд процессора P6.

Префикс	КОП	MOD R/M	SIB	смещение	операнд
0 или 1б	1 или 2б	0 или 1б	0 или 1б	0,1,2 или 4б	0,1,2 или 4б

Префикс – байт со специальным кодированием, который ставится перед командой, чтобы изменить ее действие.

Используются следующие префиксы:

REP (repeat) повторения операции. Применяется в цепочечных операциях для обработки всех элементов цепочки.

OS (operand size) изменения разрядности операнда.

AS (address size) изменения разрядности адреса.

Бит **D** в дескрипторе сегмента кода определяет разрядность операндов и используемых адресов. (D=0 – команды работают с 16 битными операндами и адресами, D=1 - команды работают с 32 битными операндами и адресами).

С помощью префиксов OS и AS можно изменить принимаемые по умолчанию размеры операндов и адресов.

SEG (segment overwrite) замена сегмента. Этот префикс явно определяет сегментный регистр для конкретной команды вместо сегментного регистра, принимаемого по умолчанию.

LOCK блокировка системной шины в течение времени выполнения команды.

Поле КОП содержит бит w, значение которого определяет разрядность операндов (w=0 – операция с байтами, w=1 – операция со словами или двойными словами).

Байт адресации MOD R/M.

7 6	5 4 3	2 1 0
MOD	REG/КОП	R/M

MOD определяет 4 возможных способа адресации.

R/M задает регистр (или смещение).

REG/КОП в одноадресных операциях расширяют поле КОП до11 разрядов. В двухадресных операциях поле REG/КОП определяет регистр, в котором хранится второй из операндов. Тип команды (одно или двухадресная команда) определяется первым битом КОП.

Байт адресации SIB (Scale, Index, Base)

SCALE задает масштабный коэффициент (00-1, 01-2, 10-4, 11-8).

INDEX определяет индексный регистр.

BASE определяет базовый регистр.

Байт SIB может присутствовать только в командах с 32 битной адресацией.

Понятия адреса и адресного пространства

Логический адрес и пространство логических адресов.

Логически программа разбивается на сегменты кода, стека, данных и другие. Сегмент представляет собой область памяти со смежными адресами. Логический адрес состоит из двух компонент, указание сегмента и смещения внутри сегмента. Для локализации сегментов в ОП используются сегментные регистры.

Линейный адрес и линейное адресное пространство.

Линейное адресное пространство представляет множество возможных адресов, формируемых на шине адреса процессора. Разрядность адресной шины определяет диапазон линейных адресов.

8086	n=20	2^{20}=1Мбайт
80286	n=24	2^{24}=16Мбайт
80386	n=32	2^{32}=4Гбайт
P6	n=36	2^{36}=16Гбайт

Пространство линейных адресов

Физический адрес и пространство физических адресов

Физический адрес – это адрес, выводимый процессором на внешнюю шину, который используется для адресации ячеек реально используемой физической оперативной памяти. При отключенном механизме страничной адресации физический адрес совпадает с линейным.

Способы адресации операндов.

Неявная адресация – операнд адресуется неявно, если в команде нет специальных полей для его определения, т.е. операнд задается кодом операции.

std – установка флага направления.

Регистровая адресация – операнд находится во внутреннем регистре процессора.

inc esi –инкремент esi

add al, dl- al = al+dl

Непосредственная адресация – операнд находится в самой команде, т.е. хранится вместе с командой в сегменте кода.

and al, 0fh al=al&0fh

Адресация ввода-вывода. Процессор поддерживает 64к адресов вв/выв, обычно называемых портами. Порт – это регистр внешнего устройства. Область вв/выв может быть разделена на 64к 8разрядных портов, 32к 16разрядных портов, 16к 32разрядных портов. В командах вв/выв источником или приемником является EAX/AX/AL, а порт определяется как непосредственный операнд или содержимое регистра DX.

in al, 40h ввод из порта 40h в регистр al

out 20h, ax вывод из ax в порт 20h

out dx, eax вывод из eax в порт, адрес которого расположен в dx

Адресация операндов в памяти

Эффективный адрес (EA) – это смещение операнда от начала сегмента.

Прямая адресация – эффективный адрес находится в самой команде.

EA=DISP(d16/32)

mov al, ds[2]

Косвенная-регистровая адресация:

EA=BASE(BX/EBX)

EA=INDEX(SI/ESI,DI/EDI)

mov ax, ds:[eax]

Базовая адресация:

EA=BASE(BX,BP) + DISP(d8,d16)

mov al,[BX+4]

Индексная адресация:

EA=INDEX(SI,DI) + DISP(d8,d16)

sub array[SI],2

Базово-индексная адресация:

EA=BASE(BX,BP) + INDEX(SI,DI)

mov ax, [bp+8][SI]

Базово-индексная адресация со смещением:

EA=BASE(BX,BP) + INDEX(SI,DI) + DISP(d8,d16)

mov ax, [bx+14h][DI]

Базово-индексная адресация со смещением и масштабированием (только в защищенном режиме с 32разрядной адресацией):

EA=BASE(РОН) + INDEX(РОН, кроме ESP)*F + DISP(d32)

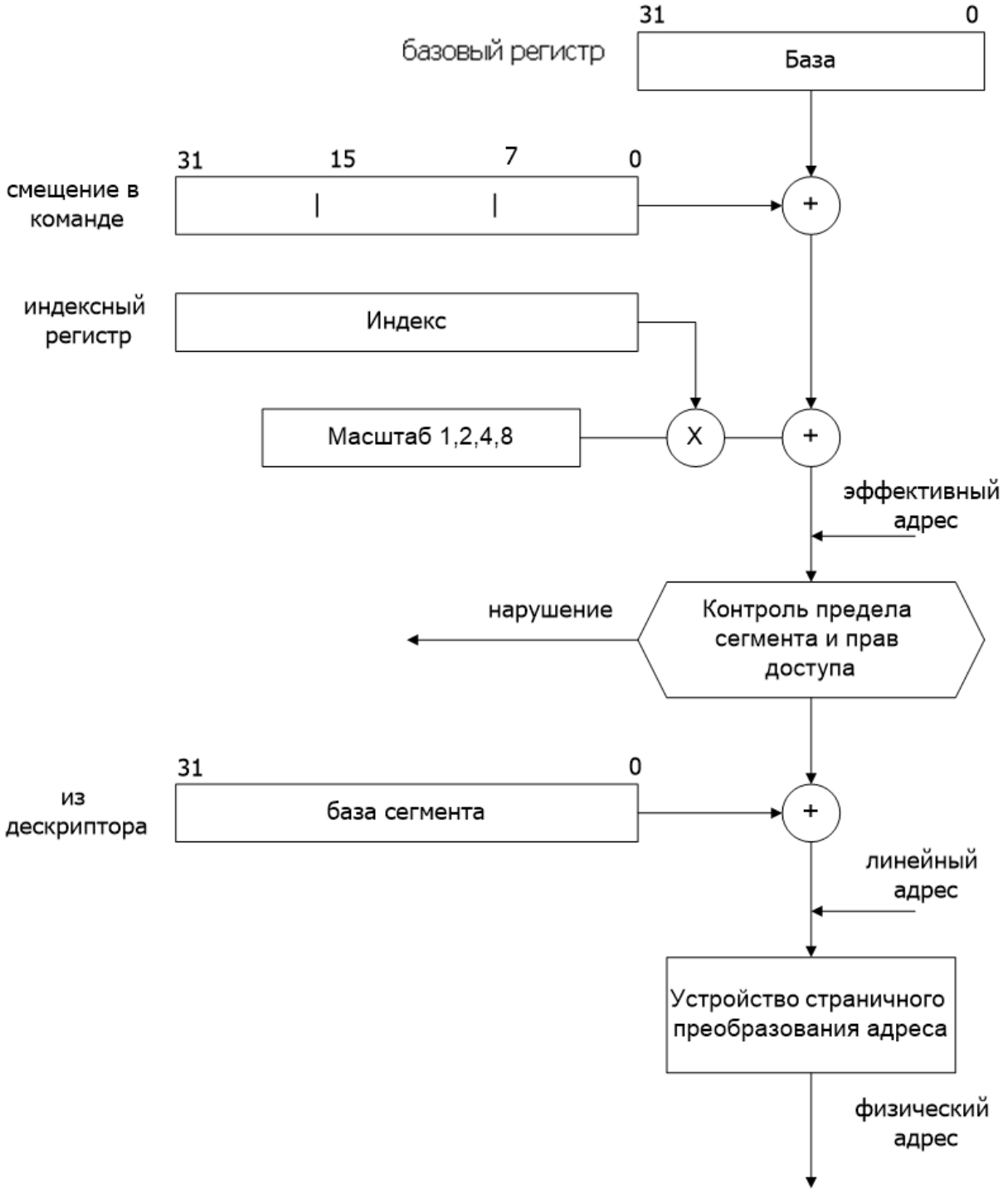
Inc word ptr [ebx+10h] [eax*2]

Стековая адресация:

PUSH – включение в стек

POP – извлечение из стека.

Формирование адреса в защищенном режиме с 32 битной адресацией.



Эффективный адрес (EA) = база(base) + индекс(index)xF + смещение(displacement).