

# Регистр процессора

Материал из Википедии — свободной энциклопедии

**Регистр процессора** — поле заданной длины во *внутрипроцессорной* сверхбыстрой *оперативной памяти* (СОЗУ). Используется самим процессором, может быть как доступным, так и недоступным программно. Например, при выборке из памяти очередной команды она помещается в *регистр команд*, обращение к которому программист прописать не может.

Содержание
<div><div><div><div><div><span></span></div><div>Программно недоступные регистры</div></div><div><div><span></span></div><div>Программно доступные регистры</div></div><div><div><span></span></div><div>Некоторые примеры</div></div><div><div><span></span></div><div>Архитектура x86</div></div><div><div><span></span></div><div>Счётчик команд</div></div><div><div><span></span></div><div>IP</div></div><div><div><span></span></div><div>EIP</div></div><div><div><span></span></div><div>RIP</div></div></div><div><div><span></span></div><div>Примечания</div></div><div><div><span></span></div><div>Ссылки</div></div></div></div>

## Программно недоступные регистры

Программно недоступные регистры — любые процессорные регистры, к которым невозможно так или иначе обратиться из выполняемой программы. Пример таких регистров — уже упомянутый *регистр команд*.

## Программно доступные регистры

Программно-доступные регистры есть регистры, к которым возможно так или иначе обратиться из выполняемой программы. Практически каждый такой регистр обозначается своим именем-идентификатором на уровне языка ассемблера и соответствующим числовым кодом-идентификатором на уровне машинного языка. По уровню доступности программно доступные регистры и неодинаковы и практически делятся на две большие подгруппы:

- системные регистры, — любые регистры, программно доступные только системным программам (например ядру операционной системы), имеющим достаточный для этого уровень системных привилегий/прав. В терминах многих машинных систем такой уровень привилегий часто называется «уровнем/режимом ядра» или «режимом супервизора». Для всех прочих программ, — работающих в «режиме пользователя», — эти регистры недоступны. Примеры таких регистров: управляющие регистры и теневые регистры дескрипторов сегментов.
- регистры общего назначения** (РОН) — регистры, доступные любым программам. В частности регистры, используемые без ограничения в арифметических и логических операциях, но имеющие определённые аппаратные ограничения (например, в строковых РОН). Эти регистры не характерны для эпохи *мейнфреймов* типа *IBM/370*<sup>[1]</sup> и стали популярными в микропроцессорах архитектуры *X86* — *Intel 8085*, *Intel 8086* и последующих<sup>[2]</sup>.

**Специальные регистры**<sup>[3]</sup> содержат данные, необходимые для работы процессора — смещения базовых таблиц, уровни доступа и т. д.

Часть специальных регистров принадлежит *устройству управления*, которое управляет процессором путём генерации последовательности микрокоманд.

Доступ к значениям, хранящимся в регистрах, осуществляется непосредственно на *такты*вой частоте процессора и, как правило, в несколько раз быстрее, чем доступ к полям в оперативной памяти (даже если *кеш-память* содержит нужные данные), но объём оперативной памяти намного превосходит суммарный объём процессорных регистров, суммарная «ёмкость» регистров *общего назначения/данных* для x86-процессоров (например *Intel 80386* и более новых) 8 регистров по 4 байта = 32 байта (В *x86-64*-процессорах — 16 по 8 байт = 128 байт и некоторое количество векторных регистров).

## Некоторые примеры

В таблице показано количество регистров общего назначения в нескольких распространённых архитектурах микропроцессоров. Стоит отметить, что в некоторых архитектурах использование отдельных регистров может быть осложнено. Так, в SPARC и MIPS регистр номер 0 не сохраняет информацию и всегда считывается как 0, а в процессорах x86 с регистром ESP (указатель на стек) могут работать лишь некоторые команды.

Архитектура	Целочисленных регистров	FP регистров	Примечания
x86-32	8	8	
x86-64	16	16	
IBM System/360	16	4	
z/Architecture	16	16	
Itanium	128	128	
SPARC	31	32	Регистр 0 (глобальный) всегда занулен
IBM Cell	4–16	1–4	
IBM POWER	32	32	
Power Architecture	32	32	
Alpha	32	32	
6502	3	0	
W65C816S	5	0	
PIC	1	0	
AVR	32	0	
ARM 32-bit <sup>[4]</sup>	16	различное	
ARM 64-bit <sup>[5]</sup>	31	32	
MIPS	31	32	Регистр 0 всегда занулён
RISC-V	31	32	Дополнительно есть регистр 0, который всегда возвращает ноль
Эльбрус 2000	256	совмещены с целочисленными	32 двухразрядных регистра, 32 глобальных регистров, 224 регистра стека процедур <sup>[6]</sup>

Архитектура x86

**IP** (англ. *Instruction Pointer*) — регистр, указывающий на смещение (адрес) инструкций в сегменте кода (1234:0100h сегмент/смещение).

IP — 16-битный (младшая часть EIP)

EIP — 32-битный аналог (младшая часть RIP)

RIP — 64-битный аналог

**Сегментные регистры** — регистры, указывающие на сегменты.

Все сегментные регистры — 16-разрядные.

CS (англ. *Code Segment*), DS (англ. *Data Segment*), SS (англ. *Stack Segment*), ES (англ. *Extra Segment*), FS, GS

В реальном режиме работы процессора сегментные регистры содержат адрес начала 64Kb сегмента, смещённый вправо на 4 бита.

В защищённом режиме работы процессора сегментные регистры содержат селектор сегмента памяти, выделенного ОС.

CS — указатель на кодовый сегмент. Связка CS:IP (CS:EIP/CS:RIP — в защищённом/64-битном режиме) указывает на адрес в памяти следующей команды.

В 64-разрядном режиме сегментные регистры CS, DS, ES и SS в формировании линейного (непрерывного) адреса не участвуют, поскольку сегментация в этом режиме не поддерживается.

**Регистры данных** — служат для хранения промежуточных вычислений.

RAX, RCX, RDX, RBX, RSP, RBP, RSI, RDI, R8 — R15 — 64-битные (registry AX)

EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI, R8D — R15D — 32-битные (extended AX)

AX (англ. *Accumulator*), CX (англ. *Count Register*), DX (англ. *Data Register*), BX (англ. *Base Register*), SP (англ. *Stack Pointer*), BP (англ. *Base Pointer*), SI (англ. *Source Index*), DI (англ. *Destination Index*), R8W — R15W — 16-битные

AH, AL, CH, CL, DH, DL, BH, BL, SPL, BPL, SIL, DIL, R8B — R15B — 8-битные (половинки 16-битных регистров)

например, AH — high AX — старшая половинка 8 бит

AL — low AX — младшая половинка 8 бит



Таблица x86-регистров.

RAX		RCX		RDX		RBX	
	EAX		ECX		EDX		EBX
	AX		CX		DX		BX
	AH   AL		CH   CL		DH   DL		BH   BL

RSP		RBP		RSI		RDI		Rx	
	ESP		EBP		ESI		EDI		RxD
	SP		BP		SI		DI		RxW
	SPL		BPL		SIL		DIL		RxB

где x — 8..15.

Регистры RAX, RCX, RDX, RBX, RSP, RBP, RSI, RDI, Rx, RxD, RxW, RxB, SPL, BPL, SIL, DIL доступны только в 64-битном режиме работы процессора.

**Регистр флагов** FLAGS (16 бит) / EFLAGS (32 бита) / RFLAGS (64 бита) — содержит текущее состояние процессора.

**Системные регистры** GDTR, LDTR и IDTR введены в процессорах начиная с Intel286 и предназначены для хранения базовых адресов *таблиц дескрипторов* — важнейших составляющих системной архитектуры при работе в защищённом режиме.

Регистр GDTR содержит 32-битный (24-битный для Intel286) базовый адрес и 16-битный предел *глобальной таблицы дескрипторов (GDT)*.

Видимая часть регистра LDTR содержит только селектор дескриптора *локальной таблицы дескрипторов (LDT)*. Сам дескриптор LDT автоматически загружается в скрытую часть LDTR из глобальной таблицы дескрипторов.

Регистр IDTR содержит 32-битный (24-битный для Intel286) базовый адрес и 16-битный предел *таблицы дескрипторов прерываний (IDT)*. В реальном режиме может быть использован для изменения местоположения таблицы векторов прерываний.

Видимая часть регистра TR содержит селектор дескриптора сегмента состояния задачи (TSS). Сам дескриптор TSS автоматически загружается в скрытую часть TR из глобальной таблицы дескрипторов.

**Регистром** называется функциональный узел, осуществляющий приём, хранение и передачу информации. Регистры состоят из группы триггеров, обычно D. По типу приёма и выдачи информации различают 2 типа регистров:

- С последовательным приёмом и выдачей информации — сдвиговые регистры.
- С параллельным приёмом и выдачей информации — параллельные регистры.

Сдвиговые регистры представляют собой последовательно соединённую цепочку триггеров. Основной режим работы — сдвиг разрядов кода от одного триггера к другому на каждый импульс тактового сигнала.

По назначению регистры различаются на:

- аккумулятор — используется для хранения промежуточных результатов арифметических и логических операций и инструкций ввода-вывода;
- флаговые — хранят признаки результатов арифметических и логических операций;
- общего назначения — хранят операнды арифметических и логических выражений, индексы и адреса;
- индексные — хранят индексы исходных и целевых элементов массива;
- указательные — хранят указатели на специальные области памяти (указатель текущей операции, указатель базы, указатель стека);
- сегментные — хранят адреса и селекторы сегментов памяти;
- управляющие — хранят информацию, управляющую состоянием процессора, а также адреса системных таблиц.

## Счётчик команд

### IP

**IP** (англ. *Instruction Pointer*) — регистр, содержащий адрес-смещение следующей команды, подлежащей исполнению, относительно кодового сегмента CS в процессорах семейства x86.

Регистр IP связан с CS в виде CS:IP, где CS является текущим кодовым сегментом, а IP — текущим смещением относительно этого сегмента.

Регистр IP является 16-разрядным регистром-указателем. Кроме него, в состав регистров этого типа входят SP (англ. *Stack Pointer* — указатель стека) и BP (англ. *Base Pointer* — базовый указатель).

### Принцип работы

Например, CS содержит значение 2CB5[0]H, в регистре IP хранится смещение 123H.

Адрес следующей инструкции, подлежащей исполнению, вычисляется путём суммирования адреса в CS (сегменте кода) со смещением в регистре IP:

$$2CB50H + 123H = 2CC73H$$

Таким образом, адрес следующей инструкции для исполнения равен 2CC73H.

При выполнении текущей инструкции процессор автоматически изменяет значение в регистре IP, в результате чего регистровая пара CS:IP всегда указывает на следующую подлежащую исполнению инструкцию.

## EIP

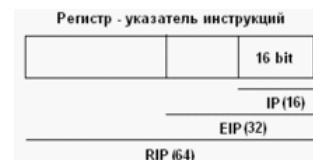
Начиная с процессора 80386 была введена 32-разрядная версия регистра-указателя — **EIP** (англ. *Extended Instruction Pointer*). В данном случае IP является младшей частью этого регистра (первые 16 разрядов). Принцип работы EIP в целом схож с работой регистра IP. Основная разница состоит в том, что в защищённом режиме, в отличие от реального режима, регистр CS является селектором (селектор указывает не на сам сегмент в памяти, а на дескриптор сегмента в таблице дескрипторов).

## RIP

В 64-разрядных процессорах используется свой регистр-указатель инструкций — **RIP**.

Младшей частью этого регистра является регистр EIP.

На основе RIP в 64-разрядных процессорах введён новый метод адресации RIP-relative. В остальном работа RIP аналогична работе регистра EIP.



## Примечания

1. *Barbara J. Burian*. Программирование на языке ассемблера системы IBM/370 упрощённый подход = A simple approach to S/370 assembly language programming. — New York: Prentice-Hall, Inc, 1977.
2. *Погорелый С. Д., Слободянюк Т. Ф.* Программное обеспечение микропроцессорных систем. Справочник. — 2-е, переработанное и дополненное. — Киев: Тэхника, 1989. — С. 7, 48-51. — 301 с. — (Справочник специалиста). — 50 000 экз. — ISBN 5-335-00169-0.
3. Intel 64 and IA-32 Architectures Software Developer’s Manual. Volume 1: Basic Architecture. 3.4 BASIC PROGRAM EXECUTION REGISTERS (<http://download.intel.com/design/processor/manuals/253665.pdf>) (англ.)
4. Procedure Call Standard for the ARM Architecture ([http://infocenter.arm.com/help/topic/com.arm.doc.ih0055a/IH0055A\\_aapcs64.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ih0055a/IH0055A_aapcs64.pdf)) . ARM Holdings (16 October 2009). Дата обращения: 24 апреля 2012. Архивировано ([https://www.webcitation.org/6GDOoNh8S?url=http://infocenter.arm.com/help/topic/com.arm.doc.ih0055a/IH0055A\\_aapcs64.pdf#](https://www.webcitation.org/6GDOoNh8S?url=http://infocenter.arm.com/help/topic/com.arm.doc.ih0055a/IH0055A_aapcs64.pdf#)) 28 апреля 2013 года.
5. Procedure Call Standard for the ARM 64-bit Architecture ([http://infocenter.arm.com/help/topic/com.arm.doc.ih0042d/IH0042D\\_aapcs.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.ih0042d/IH0042D_aapcs.pdf)) . ARM Holdings (25 November 2011). Дата обращения: 24 апреля 2012. Архивировано ([https://www.webcitation.org/6GDOosNTA?url=http://infocenter.arm.com/help/topic/com.arm.doc.ih0042e/IH0042E\\_aapcs.pdf#](https://www.webcitation.org/6GDOosNTA?url=http://infocenter.arm.com/help/topic/com.arm.doc.ih0042e/IH0042E_aapcs.pdf#)) 28 апреля 2013 года.
6. *МЦСТ*. Введение в архитектуру «Эльбрус» ([http://ftp.altlinux.org/pub/people/mike/elbrus/docs/elbrus\\_prog/html/chapter4.html#id3](http://ftp.altlinux.org/pub/people/mike/elbrus/docs/elbrus_prog/html/chapter4.html#id3)) (рус.) (неопр.) ?. Альт Линукс (31.05.2020).

## Ссылки

- Внутренние регистры: Системные регистры (<http://www.club155.ru/x86internalreg-system>)

Источник — [https://ru.wikipedia.org/w/index.php?title=Регистр\\_процессора&oldid=113682107](https://ru.wikipedia.org/w/index.php?title=Регистр_процессора&oldid=113682107)

Эта страница в последний раз была отредактирована 18 апреля 2021 в 13:09.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.  
Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.