X86 Control register

A control register is a processor register which changes or controls the general behavior of a CPU or other digital device. Common tasks performed by control registers include interrupt control, switching the addressing mode, paging control, and coprocessor control.

CR0

The CR0 register is 32 bits long on the 386 and higher processors. On x64 processors in long mode, it (and the other control registers) is 64 bits long. CR0 has various control flags that modify the basic operation of the processor.

Bit	Name	Full Name	Description
0	PE	Protected Mode Enable	If 1, system is in protected mode, else system is in real mode
1	MP	Monitor co-processor	Controls interaction of WAIT/FWAIT instructions with TS flag in CR0
2	EM	Emulation	If set, no x87 floating-point unit present, if clear, x87 FPU present
3	TS	Task switched	Allows saving x87 task context upon a task switch only after x87 instruction used
4	ET	Extension type	On the 386, it allowed to specify whether the external math coprocessor was an 80287 or 80387
5	NE	Numeric error	Enable internal x87 floating point error reporting when set, else enables PC style x87 error detection
16	WP	Write protect	When set, the CPU can't write to read-only pages when privilege level is 0
18	AM	Alignment mask	Alignment check enabled if AM set, AC flag (in EFLAGS register) set, and privilege level is 3

29	NW	Not-write through	Globally enables/disable write-through caching
30	CD	Cache disable	Globally enables/disable the memory cache
31	PG	Paging	If 1, enable paging and use the CR3 register, else disable paging.

CR1

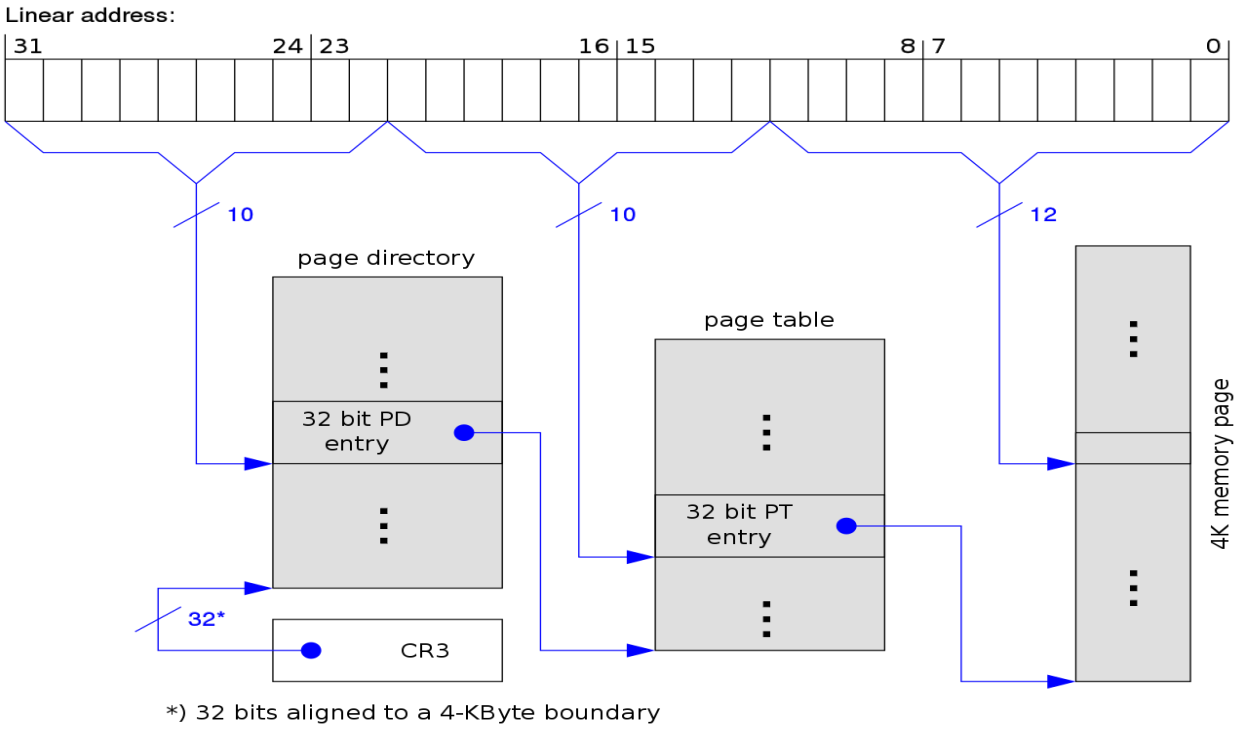
Reserved, the CPU will throw a #UD exception when trying to access it.

CR2

Contains a value called Page Fault Linear Address (PFLA). When a page fault occurs, the address the program attempted to access is stored in the CR2 register.

CR3

Used when virtual addressing is enabled, hence when the PG bit is set in CR0. CR3 enables the processor to translate linear addresses into physical addresses by locating the page directory and page tables for the current task. Typically, the upper 20 bits of CR3 become the page directory base register (PDBR), which stores the physical address of the first page directory entry. If the PCIDE bit in CR4 is set, the lowest 12 bits are used for the process-context identifier (PCID).



CR4

Used in protected mode to control operations such as virtual-8086 support, enabling I/O breakpoints, page size extension and machine-check exceptions.

Bit	Name	Full Name	Description
0	VME	Virtual 8086 Mode Extensions	If set, enables support for the virtual interrupt flag (VIF) in virtual-8086 mode.
1	PVI	Protected-mode Virtual Interrupts	If set, enables support for the virtual interrupt flag (VIF) in protected mode.
2	TSD	Time Stamp Disable	If set, RDTSC instruction can only be executed when in ring 0, otherwise RDTSC can be used at any privilege level.
3	DE	Debugging Extensions	If set, enables debug register-based breaks on I/O space access.
4	PSE	Page Size Extension	If unset, page size is 4 KiB, else page size is increased to 4 MiB If PAE is enabled or the processor is in x86-64 long mode this bit is ignored.[2]
5	PAE	Physical Address Extension	If set, changes page table layout to translate 32-bit virtual addresses into extended 36-bit physical addresses.
6	MCE	Machine Check Exception	If set, enables machine check interrupts to occur.
7	PGE	Page Global Enabled	If set, address translations (PDE or PTE records) may be shared between address spaces.
8	PCE	Performance-Monitoring Counter enable	If set, RDPMC can be executed at any privilege level, else RDPMC can only be used in ring 0.

9	OSFXSR	Operating system support for FXSAVE and FXRSTOR instructions	If set, enables Streaming SIMD Extensions (SSE) instructions and fast FPU save & restore.
10	OSXMMEXCPT	Operating System Support for Unmasked SIMD Floating-Point Exceptions	If set, enables unmasked SSE exceptions.
11	UMIP	User-Mode Instruction Prevention	If set, the SGDT, SIDT, SLDT, SMSW and STR instructions cannot be executed if CPL > 0.[1]
12	LA57	(none specified)	If set, enables 5-Level Paging.[3]
13	VMXE	Virtual Machine Extensions Enable	see Intel VT-x x86 virtualization.
14	SMXE	Safer Mode Extensions Enable	see Trusted Execution Technology (TXT)
16	FSGSBASE	Enables the instructions RDFSBASE, RDGSBASE, WRFSBASE, and WRGSBASE.	
17	PCIDE	PCID Enable	If set, enables process-context identifiers (PCIDs).
18	OSXSAVE	XSAVE and Processor Extended States Enable	
20	SMEP[4]	Supervisor Mode Execution Protection Enable	If set, execution of code in a higher ring generates a fault.
21	SMAP	Supervisor Mode Access Prevention Enable	If set, access of data in a higher ring generates a fault.
22	PKE	Protection Key Enable	See Intel 64 and IA-32 Architectures Software Developer's Manual.

CR5-7

Reserved, same case as CR1.

EFER

Extended Feature Enable Register (EFER) is a model-specific register added in the AMD K6 processor, to allow enabling the SYSCALL/SYSRET instruction, and later for entering and exiting long mode. This register becomes architectural in AMD64 and has been adopted by Intel as IA32_EFER. Its MSR number is 0xC0000080.

Bit	Purpose
0	SCE (System Call Extensions)
1	DPE (AMD K6 only: Data Prefetch Enable)
2	SEWBED (AMD K6 only: Speculative EWBE# Disable)
3	GEWBED (AMD K6 only: Global EWBE# Disable)
4	L2D (AMD K6 only: L2 Cache Disable)
5-7	Reserved, Read as Zero
8	LME (Long Mode Enable)
9	Reserved
10	LMA (Long Mode Active)
11	NXE (No-Execute Enable)
12	SVME (Secure Virtual Machine Enable)
13	LMSLE (Long Mode Segment Limit Enable)

14	FFXSR (Fast FXSAVE/FXRSTOR)
15	TCE (Translation Cache Extension)
16–63	Reserved

CR8

CR8 is a new register accessible in 64-bit mode using the REX prefix. CR8 is used to prioritize external interrupts and is referred to as the task-priority register (TPR).

The AMD64 architecture allows software to define up to 15 external interrupt-priority classes. Priority classes are numbered from 1 to 15, with priority-class 1 being the lowest and priority-class 15 the highest. CR8 uses the four low-order bits for specifying a task priority and the remaining 60 bits are reserved and must be written with zeros.

System software can use the TPR register to temporarily block low-priority interrupts from interrupting a high-priority task. This is accomplished by loading TPR with a value corresponding to the highest-priority interrupt that is to be blocked. For example, loading TPR with a value of 9 (1001b) blocks all interrupts with a priority class of 9 or less, while allowing all interrupts with a priority class of 10 or more to be recognized. Loading TPR with 0 enables all external interrupts. Loading TPR with 15 (1111b) disables all external interrupts.

The TPR is cleared to 0 on reset.

XCR0 and XSS

XCR0, or Extended Control Register 0, is a control register which is used to toggle the storing or loading of registers related to specific CPU features using the XSAVE/XRSTOR instructions. It is also used with some features to enable or disable the processor's ability to execute their corresponding instructions. It can be accessed using the privileged XSETBV and nonprivileged XGETBV instructions.

Bit	Purpose
0	X87 (x87 FPU/MMX State, note, must be '1')
1	SSE (XSAVE feature set enable for MXCSR and XMM regs)
2	AVX (AVX enable, and XSAVE feature set can be used to manage YMM regs)

3	BNDREG (MPX enable, and XSAVE feature set can be used for BND regs)
4	BNDCSR (MPX enable, and XSAVE feature set can be used for BNDCFGU and BNDSTATUS regs)
5	opmask (AVX-512 enable, and XSAVE feature set can be used for AVX opmask, AKA k-mask, regs)
6	ZMM_hi256 (AVX-512 enable, and XSAVE feature set can be used for upper-halves of the lower ZMM regs)
7	Hi16_ZMM (AVX-512 enable, and XSAVE feature set can be used for the upper ZMM regs)
8	Reserved
9	PKRU (XSAVE feature set can be used for PKRU register, which is part of the protection keys mechanism.)
10	Reserved (must be '0')
11	Control-flow Enforcement Technology (CET) User State
12	Control-flow Enforcement Technology (CET) Supervisor State
13–63	Reserved (must be '0')

There is also the IA32_XSS MSR, which is located at address 0DA0h. The IA32_XSS MSR controls bits of XCR0 which are considered to be "supervisor" state and should be invisible to regular programs. It operates with the privileged XSAVES and XRSTORS instructions by adding supervisor state to the data they operate with. Put simply, if the X87 state was enabled in XCR0 and PT state was enabled in IA32_XSS, the XSAVE instruction would only store X87 state, while the privileged XSAVES would store both X87 and PT states. Because it is an MSR, it can be accessed using the RDMSR and WRMSR instructions.

Bit	Purpose
0–7	Reserved; must be 0.
8	PT (Enables the saving and loading of nine Processor Trace MSRs.)
9–12	Reserved; must be 0.
13	HDC (Enables the saving and loading of the IA32_PM_CTL1 MSR.)
14–63	Reserved; must be 0.