Gate Descriptors provide protection for executable segments operating at different privilege levels. Figure 2-9 illustrates the format for Gate Descriptors and Table 2-9 lists the corresponding bit definitions.

Task Gate Descriptors are used to switch the CPU's context during a task switch. The selector portion of the task gate descriptor locates a Task State Segment. These descriptors can be located in the GDT, LDT or IDT tables.

Interrupt Gate Descriptors are used to enter a hardware interrupt service routine. Trap Gate Descriptors are used to enter exceptions or software interrupt service routines. Trap Gate and Interrupt Gate Descriptors can only be located in the IDT.

Call Gate Descriptors are used to enter a procedure (subroutine) that executes at the same or a more privileged level. A Call Gate Descriptor primarily defines the procedure entry point and the procedure's privilege level.

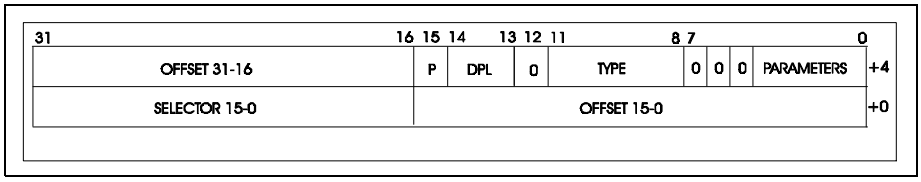

Figure 2-9 **Gate Descriptor**

Table 2-9. Gate Descriptor Bit Definitions

BIT POSITION	MEMORY OFFSET	NAME	DESCRIPTION
31-16 15-0	+4 +0	OFFSET	Offset used during a call gate to calculate the branch target.
31-16	+0	SELECTOR	Segment selector used during a call gate to calculate the branch target.
15	+4	P	Segment present.
14-13	+4	DPL	Descriptor privilege level.
11-8	+4	TYPE	Segment type: 0100 = 16-bit call gate 0101 = task gate 0110 = 16-bit interrupt gate 0111 = 16-bit trap gate 1100 = 32-bit call gate 1110 = 32-bit interrupt gate 1111 = 32-bit trap gate.
4-0	+4	PARAMETERS	Number of 32-bit parameters to copy from the caller's stack to the called procedure's stack (valid for calls).

2.4.3 Task Register

The Task Register (TR) holds a 16-bit selector for the current Task State Segment (TSS) table as shown in Figure 2-10. The TR is loaded and stored via the LTR and STR instructions, respectively. The TR can only be accessed during protected mode and can only be loaded when the privilege level is 0 (most privileged). When the TR is loaded, the TR selector field indexes a TSS descriptor that must reside in the Global Descriptor Table (GDT). The contents of the selected descriptor are cached on-chip in the hidden portion of the TR.

During task switching, the processor saves the current CPU state in the TSS before starting a new task. The TR points to the current TSS. The TSS can be either a 386/486-style 32-bit TSS (Figure 2-11, Page 2-21) or a 286-style 16-bit TSS type (Figure 2-12, Page 2-22). An I/O permission bit map is referenced in the 32-bit TSS by the I/O Map Base Address.

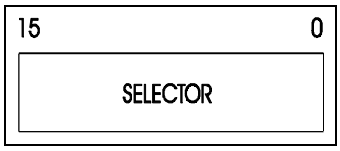

Figure 2-10. Task Register

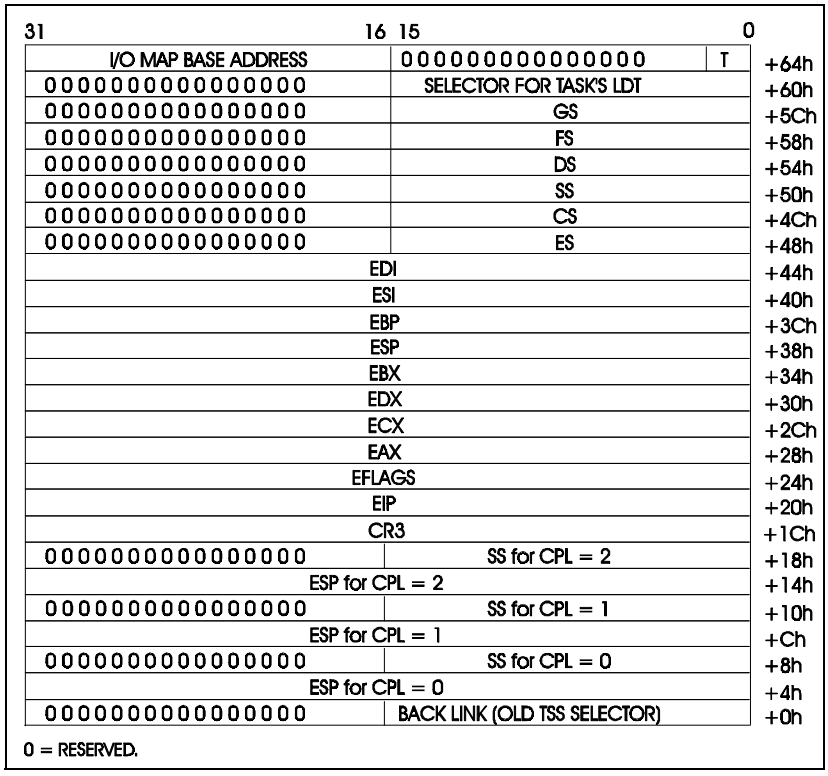

Figure 2-11. 32-Bit Task State Segment (TSS) Table

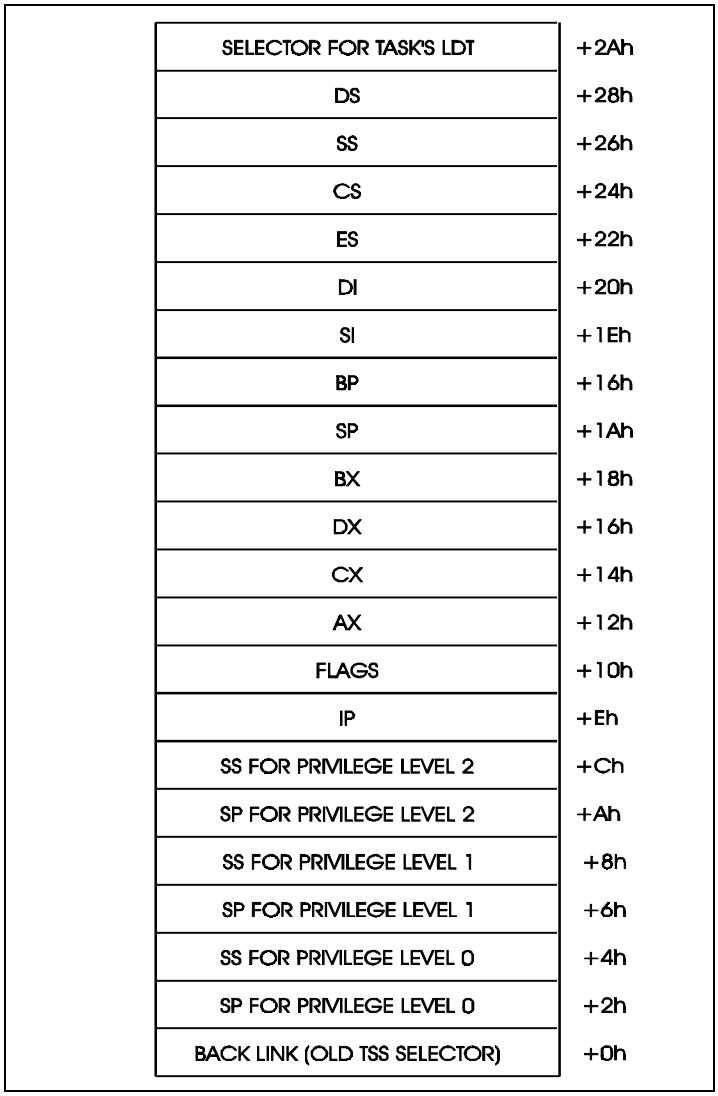

Figure 2-12. 16-Bit Task State Segment (TSS) Table

2.4.4 IBM 6x86 Configuration Registers

A set of 24 on-chip IBM 6x86 configuration registers are used to enable features in the IBM 6x86 CPU. These registers assign non-cached memory areas, set up SMM, provide CPU identification information and control various features such as cache write policy, and bus locking control. There are four groups of registers within the IBM 6x86 configuration register set:

- 6 Configuration Control Registers (CCRx)
- 8 Address Region Registers (ARRx)
- 8 Region Control Registers (RCRx)
- 2 Device Identification Registers (DIRx)

Access to the configuration registers is achieved by writing the register index number for the configuration register to I/O port 22h. I/O port 23h is then used for data transfer.

Each I/O port 23h data transfer must be preceded by a valid I/O port 22h register index selection. Otherwise, the current 22h, and the second and later I/O port 23h operations communicate through the I/O port to produce external I/O cycles. All reads from I/O port 22h produce external I/O cycles. Accesses that hit within the on-chip configuration registers do not generate external I/O cycles.

After reset, configuration registers with indexes CO-CFh and FE-FFh are accessible. To prevent potential conflicts with other devices which may use ports 22 and 23h to access their registers, the remaining registers (indexes D0-FDh) are accessible only if the MAPEN(3-0) bits in CCR3 are set to 1h. See Figure 2-16 (Page 2-28) for more information on the MAPEN(3-0) bit locations.

If MAPEN[3-0] = 1h, any access to indexes in the range 00-FFh will not create external I/O bus cycles. Registers with indexes C0-CFh, FE, FFh are accessible regardless of the state of MAPEN[3-0]. If the register index number is outside the C0-CFh or FE-FFh ranges, and MAPEN[3-0] are set to 0h, external I/O bus cycles occur. Table 2-10 (Page 2-24) lists the MAPEN[3-0] values required to access each IBM 6x86 configuration register. All bits in the configuration registers are initialized to zero following reset unless specified otherwise.

Valid register index numbers include C0h to E3h, E8h, E9h, FEh and FFh (if MAPEN[3-0] = 1).

2.4.4.1 Configuration Control Registers

(CCR0 - CCR5) control several functions, including non-cacheable memory, write-back regions, and SMM features. A list of the configuration registers is listed in Table 2-10 (Page 2-24). The configuration registers are described in greater detail in the following pages.

Table 2-10. IBM 6x86™ CPU Configuration Registers

REGISTER NAME	ACRONYM	REGISTER INDEX	WIDTH (Bits)	MAPEN VALUE NEEDED FOR ACCESS
Configuration Control 0	CCR0	C0h	8	x
Configuration Control 1	CCR1	C1h	8	x
Configuration Control 2	CCR2	C2h	8	x
Configuration Control 3	CCR3	C3h	8	x
Configuration Control 4	CCR4	E8h	8	1
Configuration Control 5	CCR5	E9h	8	1
Address Region 0	ARR0	C4h - C6h	24	x
Address Region 1	ARR1	C7h - C9h	24	x
Address Region 2	ARR2	CAh - CCh	24	x
Address Region 3	ARR3	CDh - CFh	24	x
Address Region 4	ARR4	D0h - D2h	24	1
Address Region 5	ARR5	D3h - D5h	24	1
Address Region 6	ARR6	D6h - D8h	24	1
Address Region 7	ARR7	D9h - DBh	24	1
Region Control 0	RCR0	DCh	8	1
Region Control 1	RCR1	DDh	8	1
Region Control 2	RCR2	DEh	8	1
Region Control 3	RCR3	DFh	8	1
Region Control 4	RCR4	E0h	8	1
Region Control 5	RCR5	E1h	8	1
Region Control 6	RCR6	E2h	8	1
Region Control 7	RCR7	E3h	8	1
Device Identification 0	DIR0	FEh	8	x
Device Identification 1	DIR1	FFh	8	x

Note: x = Don’t Care

7	6	5	4	3	2	1	0
Reserved	*Reserved*	*Reserved*	*Reserved*	*Reserved*	*Reserved*	NC1	*Reserved*

Figure 2-13. IBM 6x86™ Configuration Control Register 0 (CCR0)

Table 2-11. CCR0 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
1	NC1	No Cache 640 KByte - 1 MByte If = 1: Address region 640 KByte to 1 MByte is non-cacheable. If = 0: Address region 640 KByte to 1 MByte is cacheable.

Note: Bits 0, 2 through 7 are reserved.

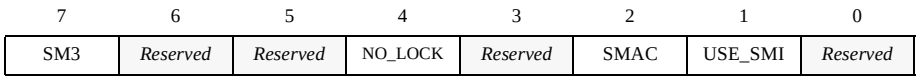

Figure 2-14. IBM 6x86 Configuration Control Register 1 (CCR1)

Table 2-12. CCR1 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
1	USE_SMI	Enable SMM and SMIACT# Pins If = 1: SMI# and SMIACT# pins are enabled. If = 0: SMI# pin ignored and SMIACT# pin is driven inactive.
2	SMAC	System Management Memory Access If = 1: Any access to addresses within the SMM address space, access system management memory instead of main memory. SMI# input is ignored. Used when initializing or testing SMM memory. If = 0: No effect on access.
4	NO_LOCK	Negate LOCK# If = 1: All bus cycles are issued with LOCK# pin negated except page table accesses and interrupt acknowledge cycles. Interrupt acknowledge cycles are executed as locked cycles even though LOCK# is negated. With NO_LOCK set, previously noncacheable locked cycles are executed as unlocked cycles and therefore, may be cached. This results in higher performance. Refer to Region Control Registers for information on eliminating locked CPU bus cycles only in specific address regions.
7	SM3	SMM Address Space Address Region 3 If = 1: Address Region 3 is designated as SMM address space. Note: If USE_SMI is set then SM3 mus also be set.

Note: Bits 0, 3, 5 and 6 are reserved.

7	6	5	4	3	2	1	0
USE_SUSP	*Reserved*	*Reserved*	WPR1	SUSP_HLT	LOCK_NW	SADS	*Reserved*

Figure 2-15. IBM 6x86 Configuration Control Register 2 (CCR2)

Table 2-13. CCR2 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
1	SADS	Slow ADS: For non-pipelinned back-to-back bus cycles only If = 1: CPU inserts an idle following sampling of BRDY# and prior to asserting ADS#. If = 0: No idle cycles are inserted between sampling of BRDY# and assertion of ADS#.
2	LOCK_NW	Lock NW If = 1: NW bit in CR0 becomes read only and the CPU ignores any writes to the NW bit. If = 0: NW bit in CR0 can be modified.
3	SUSP_HLT	Suspend on Halt If = 1: Execution of the HLT instruction causes the CPU to enter low power suspend mode.
4	WPR1	Write-Protect Region 1 If = 1: Designates any cacheable accesses in 640 KByte to 1 MByte address region are write protected.
7	USE_SUSP	Use Suspend Mode (Enable Suspend Pins) If = 1: SUSP# and SUSPA# pins are enabled. If = 0: SUSP# pin is ignored and SUSPA# pin floats.

Note: Bits 0,1, 5 and 6 are reserved.

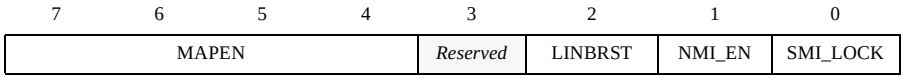

Figure 2-16. IBM 6x86 Configuration Control Register 3 (CCR3)

Table 2-14. CCR3 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
0	SMI_LOCK	SMI Lock If = 1: The following SMM configuration bits can only be modified while in an SMI service routine: CCR1: USE_SMI, SMAC, SM3 CCR3: NMI_EN ARR3: Starting address and block size. Once set, the features locked by SMI_LOCK cannot be unlocked until the RESET pin is asserted.
1	NMI_EN	NMI Enable If = 1: NMI interrupt is recognized while servicing an SMI interrupt. NMI_EN should be set only while in SMM, after the appropriate SMI interrupt service routine has been setup.
2	LINBRST	If = 1: Use linear address sequence during burst cycles. If = 0: Use “1 + 4” address sequence during burst cycles. The “1 + 4” address sequence is compatible with Pentium’s burst address sequence.
4 - 7	MAPEN	MAP Enable If = 1h: All configuration registers are accessible. If = 0h: Only configuration registers with indexes C0-CFh, FEh and FFh are accessible.

Note: Bit 3 is reserved.

7	6	5	4	3	2	1	0
CPUID	*Reserved*	*Reserved*	DTE_EN	*Reserved*	IORT		

Figure 2-17. IBM 6x86 Configuration Control Register 4 (CCR4)

Table 2-15. CCR4 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
0 - 2	IORT	I/O Recovery Time Specifies the minimum number of bus clocks between I/O accesses: 0h = 1 clock delay 1h = 2 clock delay 2h = 4 clock delay 3h = 8 clock delay 4h = 16 clock delay 5h = 32 clock delay (default value after RESET) 6h = 64 clock delay 7h = no delay
4	DTE_EN	Enable Directory Table Entry Cache If = 1: the Directory Table Entry cache is enabled.
7	CPUID	Enable CPUID instruction. If = 1: the ID bit in the EFLAGS register can be modified and execution of the CPUID instruction occurs as documented in section 6.3. If = 0: the ID bit in the EFLAGS register can not be modified and execution of the CPUID instruction causes an invalid opcode exception.

Note: Bits 3 and bits 5 and 6 are reserved.

7	6	5	4	3	2	1	0
Reserved	*Reserved*	ARREN	LBR1	*Reserved*	*Reserved*	SLOP	WT_ALLOC

Figure 2-18. IBM 6x86 Configuration Control Register 5 (CCR5)

Table 2-16. CCR5 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
0	WT_ALLOC	Write-Through Allocate If = 1: New cache lines are allocated for read and write misses. If = 0: New cache lines are allocated only for read misses.
1	SLOP	If set, the LOOP instruction is slowed down to allow programs with poorly written software timing loops to function correctly. If clear, the LOOP instruction executes in one clock.
4	LBR1	Local Bus Region 1 If = 1: LBA# pin is asserted for all accesses to the 640 KByte to 1 MByte address region.
5	ARREN	Enable ARR Registers If = 1: Enables all ARR registers. If = 0: Disables the ARR registers. If SM3 is set, ARR3 is enabled regardless of the setting of ARREN.

Note: Bits 1 through 3 and 6 though 7 are reserved.

2.4.4.2 Address Region Registers

The Address Region Registers (ARR0 - ARR7) (Figure 2-19) are used to specify the location and size for the eight address regions.

Attributes for each address region are specified in the Region Control Registers (RCR0-RCR7). ARR7 and RCR7 are used to define system main memory and differ from ARR0-6 and RCR0-6.

With non-cacheable regions defined on-chip, the IBM 6x86 CPU delivers optimum performance by using advanced techniques to eliminate data dependencies and resource conflicts in its execution pipelines. If KEN# is active for accesses to regions defined as non-cacheable by the RCRs, the region is not cached. The RCRs take precedence in this case.

A register index, shown in Table 2-17 (Page 2-32) is used to select one of three bytes in each ARR.

The starting address of the ARR address region, selected by the START ADDRESS field, must be on a block size boundary. For example, a 128 KByte block is allowed to have a starting address of 0 KBytes, 128 KBytes, 256 KBytes, and so on.

The SIZE field bit definition is listed in (Page 2-32). If the SIZE field is zero, the address region is of zero size and thus disabled.

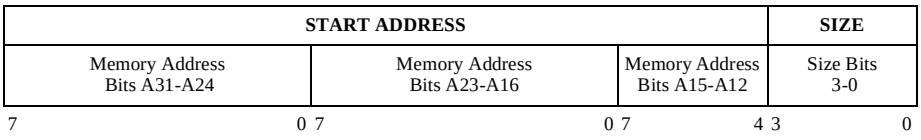

Figure 2-19. Address Region Registers (ARR0 - ARR7)

Table 2-17. ARR0 - ARR7 Register Index Assignments

ARR Register	Memory Address (A31 - A24)	Memory Address (A23 - A16)	Memory Address (A15 - A12)	Address Region Size (3 - 0)
ARR0	C4h	C5h	C6h	C6h
ARR1	C7h	C8h	C9h	C9h
ARR2	CAh	CBh	CCh	CCh
ARR3	CDh	CEh	CFh	CFh
ARR4	D0h	D1h	D2h	D2h
ARR5	D3h	D4h	D5h	D5h
ARR6	D6h	D7h	D8h	D8h
ARR7	D9h	DAh	DBh	DBh

.

Table 2-18. Bit Definitions for SIZE Field

SIZE (3-0)	BLOCK SIZE	
	ARR0-6	ARR7
0h	Disabled	Disabled
1h	4 KBytes	256 KBytes
2h	8 KBytes	512 KBytes
3h	16 KBytes	1 MBytes
4h	32 KBytes	2 MBytes
5h	64 KBytes	4 MBytes
6h	128 KBytes	8 MBytes
7h	256 KBytes	16 MBytes

SIZE (3-0)	BLOCK SIZE	
	ARR0-6	ARR7
8h	512 KBytes	32 MBytes
9h	1 MBytes	64 MBytes
Ah	2 MBytes	128 MBytes
Bh	4 MBytes	256 MBytes
Ch	8 MBytes	512 MBytes
Dh	16 MBytes	1 GBytes
Eh	32 MBytes	2 GBytes
Fh	4 GBytes	4 GBytes

2.4.4.3 Region Control Registers

The Region Control Registers (RCR0 - RCR7) specify the attributes associated with the ARRx address regions. The bit definitions for the region control registers are shown in Figure 2-20 (Page 2-34) and in Table 2-19 (Page 2-34). Cacheability, weak write ordering, weak locking, write gathering, cache write policies and control of the LBA# pin can be activated or deactivated using the attribute bits.

If an address is accessed that is not in a memory region defined by the ARRx registers, the following conditions will apply:

- LBA# pin is asserted
- The memory access is cached, if KEN# is returned asserted.
- If the memory address is cached, write-back is enabled if WB/WT# is returned high.
- Writes are not gathered
- Strong locking takes place
- Strong write ordering takes place

Overlapping Conditions Defined. If two regions specified by ARRx registers overlap and conflicting attributes are specified, the following attributes take precedence:

- LBA# pin is asserted
- The overlapping regions are non-cacheable.
- Write-back is disabled
- Writes are not gathered
- Strong locking takes place
- Strong write ordering takes place

7	6	5	4	3	2	1	0
Reserved	*Reserved*	NLB	WT	WG	WL	WWO	RCD / RCE*

*Note: RCD is defined for RCR0-RCR6. RCE is defined for RCR7.

Figure 2-20. Region Control Registers (RCR0-RCR7)

Table 2-19. RCR0-RCR7 Bit Definitions

RCRx	BIT POSITION	NAME	DESCRIPTION
0 - 6	0	RCD	If = 1: Disables caching for address region specified by ARRx.
7	0	RCE	If = 1: Enables caching for address region ARR7.
0 - 7	1	WWO	If = 1: Weak write ordering for address region specified by ARRx.
0 - 7	2	WL	If = 1: Weak locking for address region specified by ARRx.
0 - 7	3	WG	If = 1: Write gathering for address region specified by ARRx.
0 - 7	4	WT	If = 1: Address region specified by ARRx is write-through.
0 - 7	5	NLB	If = 1:LBA# pin is not asserted for access to address region specified by ARRx

Note: Bits 6 and 7 are reserved.

Region Cache Disable (RCD). Setting RCD to a one defines the address region as non-cacheable. Whenever possible, the RCRs should be used to define non-cacheable regions rather than using external address decoding and driving the KEN# pin.

Region Cache Enable (RCE). Setting RCE to a one defines the address region as cacheable. RCE is used to define the system main memory as cacheable memory. It is implied that memory outside the region is non-cacheable.

Weak Write Ordering (WWO). Setting WWO=1 enables weak write ordering for that address region. Enabling WWO allows the IBM 6x86 CPU to issue writes in its internal cache in an order different than their order in the code stream. External writes always occur in order (strong ordering). Therefore, this should only be enabled for memory regions that are NOT sensitive to this condition. WWO should not be enabled for memory mapped I/O. WWO only applies to memory regions that have been cached and designated as write-back. It also applies to previously cached addresses even if the cache has been disabled (CD=1). Enabling WWO removes the write-ordering restriction and improves performance due to reduced pipeline stalls.

Weak Locking (WL). Setting WL=1 enables weak locking for that address region. With WL enabled, all bus cycles are issued with the LOCK# pin negated except for page table accesses and interrupt acknowledge cycles. Interrupt acknowledge cycles are executed as locked cycles even though LOCK# is negated. With WL=1, previously non-cacheable locked cycles are executed as unlocked cycles and therefore, may be cached, resulting in higher performance. The NO_LOCK

bit of CCR1 enables weak locking for the entire address space. The WL bit allows weak locking only for specific address regions. WL is independent of the cacheability of the address region.

Write Gathering (WG). Setting WG=1 enables write gathering for the associated address region. Write gathering allows multiple byte, word, or dword sequential address writes to accumulate in the on-chip write buffer. (As instructions are executed, the results are placed in a series of output buffers. These buffers are gathered into the finial output buffer).

When access is made to a non-sequential memory location or when the 8-byte buffer becomes full, the contents of the buffer are written on the external 64-bit data bus. Performance is enhanced by avoiding as many as seven memory write cycles.

WG should not be used on memory regions that are sensitive to write cycle gathering. WG can be enabled for both cacheable and non-cacheable regions.

Write Through (WT). Setting WT=1 defines the address region as write-through instead of write-back, assuming the region is cacheable. Regions where system ROM are loaded (shadowed or not) should be defined as write-through.

LBA# Not Asserted (NLB). Setting NLB=1 prevents the microprocessor from asserting the Local Bus Access (LBA#) output pin for accesses to that address region. The RCR regions may be used to define non-local bus address regions. The LBA# pin could then be asserted for all regions, except those defined by the RCRs. The LBA# signal may be used by the external hardware (e.g., chipsets) as an indication that local bus accesses are occurring.

2.4.4.4 Device Identification Registers

The Device Identification Registers (DIR0, DIR1) contain CPU identification, CPU stepping and CPU revision information. Bit definitions are shown in Figure 2-21, Table 2-20, Figure 2-22 and Table 2-21 respectively. Data in these registers cannot be changed. These registers can be read by using I/O ports 22 and 23. The register index for DIR0 is FEh and the register index for DIR1 is FFh.

Figure 2-21. Device Identification Register 0 (DIR0)

Table 2-20. DIR0 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
7 - 0	DEVID	CPU Device Identification Number (read only).

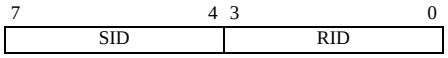

Figure 2-22. Device Identification Register 1 (DIR1)

Table 2-21. DIR1 Bit Definitions

BIT POSITION	NAME	DESCRIPTION
7 - 4	SID	CPU Step Identification Number (read only).
3 - 0	RID	CPU Revision Identification (read only).

2.4.5 Debug Registers

Six debug registers (DR0-DR3, DR6 and DR7), shown in Figure 2-23, support debugging on the IBM 6x86 CPU. The bit definitions for the debug registers are listed in Table 2-22 (Page 2-38).

Memory addresses loaded in the debug registers, referred to as “breakpoints”, generate a debug exception when a memory access of the specified type occurs to the specified address. A data breakpoint can be specified for a particular kind of memory access such as a read or a write. Code breakpoints can also be set allowing debug exceptions to occur whenever a given code access (execution) occurs.

The size of the debug target can be set to 1, 2, or 4 bytes. The debug registers are accessed via MOV instructions which can be executed only at privilege level 0.

The Debug Address Registers (DR0-DR3) each contain the linear address for one of four possible breakpoints. Each breakpoint is further specified by bits in the Debug Control Register (DR7). For each breakpoint address in DR0-DR3, there are corresponding fields L, R/W, and LEN in DR7 that specify the type of memory access associated with the breakpoint.

The R/W field can be used to specify instruction execution as well as data access breakpoints. Instruction execution breakpoints are always taken before execution of the instruction that matches the breakpoint.

The Debug Status Register (DR6) reflects conditions that were in effect at the time the debug exception occurred. The contents of the DR6 register are not automatically cleared by the processor after a debug exception occurs and, therefore, should be cleared by software at the appropriate time.

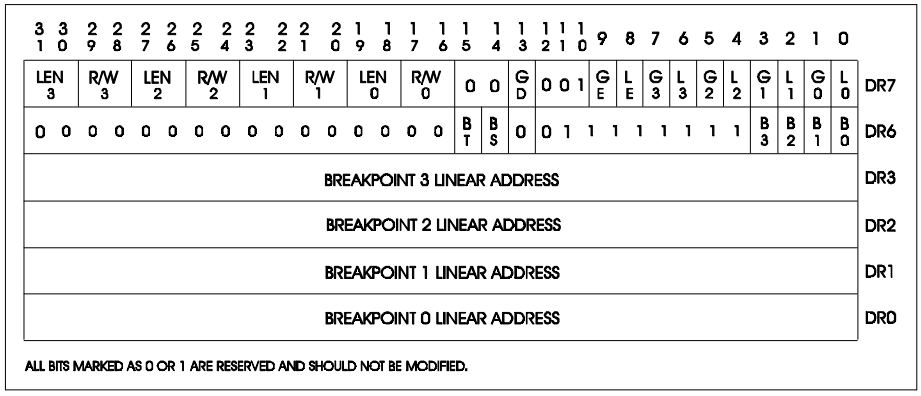

Figure 2-23. Debug Registers

Code execution breakpoints may also be generated by placing the breakpoint instruction (INT 3) at the location where control is to be regained. Additionally, the single-step feature may be enabled by setting the TF flag in the EFLAGS register. This causes the processor to perform a debug exception after the execution of every instruction.

Table 2-22. DR6 and DR7 Debug Register Field Definitions

REGISTER	FIELD	NUMBER OF BITS	DESCRIPTION
DR6	Bi	1	Bi is set by the processor if the conditions described by DRi, R/Wi, and LENi occurred when the debug exception occurred, even if the breakpoint is not enabled via the Gi or Li bits.
	BT	1	BT is set by the processor before entering the debug handler if a task switch has occurred to a task with the T bit in the TSS set.
	BS	1	BS is set by the processor if the debug exception was triggered by the single-step execution mode (TF flag in EFLAGS set).
DR7	R/Wi	2	Specifies type of break for the linear address in DR0, DR1, DR3, DR4: 00 - Break on instruction execution only 01 - Break on data writes only 10 - Not used 11 - Break on data reads or writes.
	LENi	2	Specifies length of the linear address in DR0, DR1, DR3, DR4: 00 - One byte length 01 - Two byte length 10 - Not used 11 - Four byte length.
	Gi	1	If set to a 1, breakpoint in DRi is globally enabled for all tasks and is not cleared by the processor as the result of a task switch.
	Li	1	If set to a 1, breakpoint in DRi is locally enabled for the current task and is cleared by the processor as the result of a task switch.
	GD	1	Global disable of debug register access. GD bit is cleared whenever a debug exception occurs.

2.4.6 Test Registers

The test registers can be used to test the on-chip unified cache and to test the main TLB. The test registers are also used to enable IBM 6x86 CPU variable-size paging.

Test registers TR3, TR4, and TR5 are used to test the unified cache. Use of these registers is described with the memory caches later in this chapter in Section 2.7.1.1.

Test registers TR6 and TR7 are used to test the TLB. Use of these test registers is described in Section 2.6.4.2.

NOTICE TO CUSTOMERS: Some of the information contained in this document was obtained through a third party and IBM has not conducted independent tests of all product characteristics contained herein. The product described in this document is sold under IBM's standard warranty.

The information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not effect or change IBM's product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. All the information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS. In no event will IBM be liable for any damages arising directly or indirectly from any use of the information contained in this document.

Printed in the United States of America
2-96

IBM Corporation
1000 River Street
Essex Junction, Vermont 05452-4299
United States of America