AA-64 architecture model specific registers

note: The model specific registers depend on the implementation.

Time Stamp Counter

name	63		32	31	
TSC 0000_0010h	time stamp counter value				
TSC_AUX C000_0103h	reserved				processor ID value

Feature Control

name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	
TR12 0000_000Eh	reserved or used otherwise																						ITR	reserv used ot				
MISC_CTL 0000_0119h	reserved or used otherwise										PSND	reserved or used otherwise																
MISC_ENABLE 0000_01A0h	ignored					L1DCCM	ETPRD	LCMV	FPRE	?	ASPD	MONE	BDPHE	GV3E	GV1+E	?	TM2E	PEBSU	BTSU	PBE	PQD	SLE	PMA	L3D	TCD		
EFER C000_0080h	reserved or used otherwise															FFXSR	LMSLE		NXE	LMA		LME	reserv used ot				

SYSENTER and SYSEXIT

name	63		32	31		16	15	
							base se	

SEP_SEL 0000_0174h	ignored	scratch	SYSENT a SYSEX
SEP_RSP 0000_0175h	target RSP		
SEP_RIP 0000_0176h	target RIP		

SYSCALL and SYSRET

name	63	48	47	32	31
STAR C000_0081h	base selector for SYSRET CS/SS		base selector for SYSCALL CS/SS		target EIP
LSTAR C000_0082h	target RIP for PM64 callers				
CSTAR C000_0083h	target RIP for CM callers				
FMASK C000_0084h	reserved				RFLAGS mask for SY

FS base and GS base

	63
FS_BAS C000_0100h	FS base
GS_BAS C000_0101h	GS base
KERNEL_GS_BAS C000_0102h	kernel GS base (for SWAPGS)

Page Attribute Table

name	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	3

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
PAT 0000_0277h	reserved					PA7			reserved					PA6			reserved					PA5					reserve	
	reserved					PA3			reserved					PA2			reserved					PA1					reserve	

Memory Type Range Registers

name	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5
MTRR_CAP 0000_00FEh	reserved																										
	reserved																				WC	r.	FIX			VC	
MTRR_DEF_TYPE 0000_02FFh	reserved																										
	reserved																				E	FE	res.			T	

Fixed Range MTRRs

name	63		56	55		48	47		40	39		32	31		24	23		16	15	
MTRR_FIX_64K_00000 0000_0250h	7_0000h 7_FFFFh			6_0000h 6_FFFFh			5_0000h 5_FFFFh			4_0000h 4_FFFFh			3_0000h 3_FFFFh			2_0000h 2_FFFFh			1_0000 1_FFFF	
MTRR_FIX_16K_80000 0000_0258h	9_C000h 9_FFFFh			9_8000h 9_BFFFh			9_4000h 9_7FFFh			9_0000h 9_3FFFh			8_C000h 8_FFFFh			8_8000h 8_BFFFh			8_4000 8_7FFF	
MTRR_FIX_16K_A0000 0000_0259h	B_C000h B_FFFFh			B_8000h B_BFFFh			B_4000h B_7FFFh			B_0000h B_3FFFh			A_C000h A_FFFFh			A_8000h A_BFFFh			A_4000 A_7FFF	
MTRR_FIX_4K_C0000 0000_0268h	C_7000h C_7FFFh			C_6000h C_6FFFh			C_5000h C_5FFFh			C_4000h C_4FFFh			C_3000h C_3FFFh			C_2000h C_2FFFh			C_1000 C_2FFF	

MTRR_FIX_4K_C8000 0000_0269h	C_F000h C_FFFFh	C_E000h C_EFFFh	C_D000h C_DFFFh	C_C000h C_CFFFh	C_B000h C_BFFFh	C_A000h C_AFFFh	C_9000 C_9FFF
MTRR_FIX_4K_D0000 0000_026Ah	D_7000h D_7FFFh	D_6000h D_6FFFh	D_5000h D_5FFFh	D_4000h D_4FFFh	D_3000h D_3FFFh	D_2000h D_2FFFh	D_1000 D_2FFF
MTRR_FIX_4K_D8000 0000_026Bh	D_F000h D_FFFFh	D_E000h D_EFFFh	D_D000h D_DFFFh	D_C000h D_CFFFh	D_B000h D_BFFFh	D_A000h D_AFFFh	D_9000 D_9FFF
MTRR_FIX_4K_E0000 0000_026Ch	E_7000h E_7FFFh	E_6000h E_6FFFh	E_5000h E_5FFFh	E_4000h E_4FFFh	E_3000h E_3FFFh	E_2000h E_2FFFh	E_1000 E_2FFF
MTRR_FIX_4K_E8000 0000_026Dh	E_F000h E_FFFFh	E_E000h E_EFFFh	E_D000h E_DFFFh	E_C000h E_CFFFh	E_B000h E_BFFFh	E_A000h E_AFFFh	E_9000 E_9FFF
MTRR_FIX_4K_F0000 0000_026Eh	F_7000h F_7FFFh	F_6000h F_6FFFh	F_5000h F_5FFFh	F_4000h F_4FFFh	F_3000h F_3FFFh	F_2000h F_2FFFh	F_1000 F_2FFF
MTRR_FIX_4K_F8000 0000_026Fh	F_F000h F_FFFFh	F_E000h F_EFFFh	F_D000h F_DFFFh	F_C000h F_CFFFh	F_B000h F_BFFFh	F_A000h F_AFFFh	F_9000 F_9FFF

Variable Range MTRRs

name	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5
MTRR_PHYS_BASE_n 0000_0200h 0000_0202h ...	reserved												BASE #1														
	BASE																				res.		T				
MTRR_PHYS_MASK_n 0000_0201h 0000_0203h ...	reserved												MASK #1														
	MASK																				V	reserv					

note	description
#1	The number of actually implemented bits depends on the number of phy address bits. The remaining bits are reserved if less than 52 physical address bits are implemented.

Machine Check Exception

name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
MCAR 0000_0000h	ADDR																											
MCTR 0000_0001h	reserved or used otherwise																											LCK

Machine Check Architecture

name	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
MCG_CAP 0000_0179h	reserved																											
	reserved								EXT_COUNT								reserved						EXTP	CTLP	COU			
MCG_STATUS 0000_017Ah	reserved																											
	reserved																											
MCG_CTL 0000_017Bh	reserved																											
	reserved																											

MCA Error-Reporting Register Banks

name	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5
MCn_CTL	EE63	EE62	EE61	EE60	EE59	EE58	EE57	EE56	EE55	EE54	EE53	EE52	EE51	EE50	EE49	EE48	EE47	EE46	EE45	EE44	EE43	EE42	EE41	EE40	EE39	EE38	EE37

0000_0400h 0000_0404h ...	EE31	EE30	EE29	EE28	EE27	EE26	EE25	EE24	EE23	EE22	EE21	EE20	EE19	EE18	EE17	EE16	EE15	EE14	EE13	EE12	EE11	EE10	EE09	EE08	EE07	EE06	EE05			
MCn_STATUS 0000_0401h 0000_0405h ...	VAL	O	UC	EN	MISCV	ADDRV	PCC	OTHER																						
	ERROR_MS																											ERROR_MCA		
MCn_ADDR 0000_0402h 0000_0406h ...	ADDR #1																											ADDR #1		
	ADDR #1																													
MCn_MISC 0000_0403h 0000_0407h ...	reserved																													
	reserved																													

note	description
#1	Depending on the particular error, the address can be virtual or physical.

MCA Extended State Registers

	63	
MCG_RAX 0000_0180h	RAX	
MCG_RBX 0000_0181h	RBX	
MCG_RCX 0000_0182h	RCX	
MCG_RDX 0000_0183h	RDX	
MCG_RSI 0000_0184h	RSI	
MCG_RDI		

0000_0185h	RDI
MCG_RBP 0000_0186h	RBP
MCG_RSP 0000_0187h	RSP
MCG_RFLAGS 0000_0188h	RFLAGS
MCG_RIP 0000_0189h	RIP
MCG_MISC 0000_018Ah	reserved
MCG_RESx 0000_018Bh 0000_018Ch ...	processor-specific information (optional)
	processor-specific information (optional)
MCG_R8 0000_0190h	R8
MCG_R9 0000_0191h	R9
MCG_R10 0000_0192h	R10
MCG_R11 0000_0193h	R11
MCG_R12 0000_0194h	R12
MCG_R13 0000_0195h	R13
MCG_R14 0000_0196h	R14
MCG_R15 0000_0197h	R15

Local APIC

name	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	3
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
APIC_BASE 0000_001Bh	reserved												APIC base [#1]															
	APIC base																						E	ign.	BSP	res		

note	description
#1	The number of actually implemented bits depends on the number of physic bits. The remaining bits are reserved if less than 52 physical address bits are im

---

note: The following SMM related registers are visible in the SMM state save ma

SMM related internal registers

	63		32	31	
SMBASE					SMM base addres
IO RESTART RIP	RIP of most recent IN/OUT instruction (for I/O restart on RSM				
IO RESTART RCX	RCX of most recent IN/OUT instruction (for I/O restart on RSM				
IO RESTART RSI	RSI of most recent IN/OUT instruction (for I/O restart on RSM				
IO RESTART RDI	RDI of most recent IN/OUT instruction (for I/O restart on RSM				

note: Some of the following additional internal flags are visible in the SMM state sav

additional internal flags	
name	description
TEMP_DR6	used to collect breakpoint information for DR6.B?
CAUSING_DB	used to indicate that the CPU is in the process of generat exception
BLOCK_INIT	set by SMI, cleared by IRET/RSM instruction or RESET
BLOCK_SMI	set by SMI, cleared by RSM instruction or RESET/INIT
BLOCK_NMI	set by SMI/NMI, cleared by IRET instruction or RESET/IN
LATCH_INIT	one INIT can be latched while INITs are blocked
LATCH_SMI	one SMI can be latched while SMIs are blocked
LATCH_NMI	one NMI can be latched while NMIs are blocked
IN_REP	used to suppress fetch and decode in subsequent REP s instruction iterations
IN_SMM	set by SMI, cleared by RSM instruction or RESET/INIT
IN_HLT	set by HLT instruction, optionally set by RSM instruction, cleared by RESET/INIT/SMI/NMI/INTR
IN_SHUTDOWN	set by triple fault, cleared by RESET/INIT/SMI/NMI
IN_FP_FREEZE	set by waiting FP instruction if unmasked pending FP exc while CR0.NE=0 and IGNNE#=deasserted, cleared by RESET/INIT/SMI/N
SUPPRESS_INTERRUPTS	used to implement external interrupt suppression