Chapter 2
Tables

Table 2.1 K6 Family Members

Processor	Clock	Bus	Process[a]	Die-Size	TC[b]	Comments
K6 MMX	166-MHz	66-MHz	0.35	162-mm2	8.8	original release
	200-MHz	66-MHz	0.35	162-mm2		higher clock
	233-MHz	66-MHz	0.35	162-mm2		higher clock
	266-MHz	66-MHz	0.35	162-mm2		higher clock
	300-MHz	66-MHz	0.25	68-mm2		K6 MMX shrink
K6 3D	300-MHz	100-MHz	0.25	81-mm2	9.3	K6 with 3D [c] & MMX[d]
	350-MHz	100-MHz	0.25	81-mm2		higher clock
K6 3D	350-MHz	100-MHz	0.25	135-mm2	21.3	K6 3D with L2-Cache on-chip
	400-MHz	100-MHz	0.25	135-mm2		higher clock

[a] in microns

[b] transistor count in millions

[c] also called "AMD-3D Technology"

[d] a superscalar dual-pipeline implementation of the x86 MMX instruction set extensions

Table 2.2 TYPES OF RISC86 OPS

Types of Ops	Mnemonic
memory load operations	LdOps
memory store operations	StOps
integer register operations, MMX register operations, and 3D register operations	Integer, MMX, and 3D RegOps
floating-point register operations	FpOps
branch condition evaluations	BrOps
special operations (such as load immediate constant into a register)	SpecOps

Table 2.3 x86 General Purpose Register Names and Sizes

32-Bit Name (dword)	16-bit Name (word)	8-bit Name (high order byte)	8-bit Name (low order byte)
EAX	AX	AH	AL
EBX	BX	BH	BL
ECX	CX	CH	CL
EDX	DX	DH	DL
EDI	DI	—	—
ESI	SI	—	—
ESP	SP	—	—
EBP	BP	—	—

.

Table 2.4 REGISTER NUMBER/NAME CORRESPONDENCE

Register Number	32-bit Register Name	1-Byte Register Name	64-bit MMX/ 3D Register Name
00000	EAX	AL	MMreg
00001	ECX	CL	MMreg
00010	EDX	DL	MMreg
00011	EBX	BL	MMreg
00100	ESP	AH	MMregm
00101	EBP	CH	MMregm
00110	ESI	DH	MMregm
00111	EDI	BH	MMregm
01000	t1	t1L	MMt1
01001	t2	t2L	—
01010	t3	t3L	—
01011	t4	t4L	—
01100	t5	t1H	—
01101	t6	t2H	—
01110	t7	t3H	—
01111	t0/_[a]	t4H	—
10000	t8	t8L	—
10001	t9	t9L	—
10010	t10	t10L	—
10011	t11	t11L	—
10100	t12	t8H	—
10101	t13	t9H	—
10110	t14	t10H	—
10111	t15	t11H	—
11000	reg	reg	MM0
11001	reg	reg	MM1

Table 2.4 REGISTER NUMBER/NAME CORRESPONDENCE (CONT)

Register Number	32-bit Register Name	1-Byte Register Name	64-bit MMX/ 3D Register Name
11010	reg	reg	MM2
11011	reg	reg	MM3
11100	regm	regm	MM4
11101	regm	regm	MM5
11110	regm	regm	MM6
11111	regm	regm	MM7

[a] The "t0" and "_" mnemonics are synonymous. "_" is used when an operand or result value is a "don't care." t0 is like the "traditional" RISC R0 register.

.

Table 2.5 SR0, General Control and Status Register

Bit	Name	Function	Access
0:1	CPL	Copy of architectural current privilege level (CPL)	R
2	IOS	I/O sensitivity status	R
3	V86	V86 mode = EFlags.VM && CR0.PE	R
4	REAL	Real Mode = !CR0.PE	R
5	EWBE	External write buffer empty	R
6	BusBsy	Indicates if there are any active/asserted internal requests for bus cycles in the processor system bus	R
7	PMSP	POP memory base = SP (from OpQuad Sequence environment)	R
8	FLUSHP	FLUSH# request pending	R
9	SMIP	SMI# request pending	R
10	INITP	INIT request pending	R
11	NMIP	NMI request pending	R
12	INTRP	INTR request pending	R
13	STPCLKP	STPCLK request pending	R
14	VME	Virtual Mode Extension	R/W
15	PVI	Protected Virtual Interrupt	R/W
16	ClrFLUSHP	Clear FLUSH# edge latch	W
17	ClrSMIP	Clear SMI# edge latch	W
18	ClrINITP	Clear INIT edge latch	W
19	ClrNMIP	Clear NMI edge latch	W
20	ClrISTF	Clear INTR/STPCLK# temporary mask flag	W
21	ClrBSNTF	Clear IBrkPt/SMI#/NMI temporary mask flg	W
22	NF	NMI mask flag	R/W
23	RIF	Halt instruction fetch	W
24	SMIACT	System management mode active	R/W
25	FERR	Floating-point error pending	R/W
26	StopClk	Allows an OpQuad sequence to stop clock	R/W

Table 2.5 SR0, General Control and Status Register (Cont)

Bit	Name	Function	Access
27	HaltClk	Allows an OpQuad sequence to stop clock	R/W
28	IGNNE	Ignore CR0.NE	R
29	RBGO	RAM BIST go/initiate	W
30	RBDN	RAM BIST done status	R
31	RBPF	RAM BIST pass/fail status	R

.

Table 2.6 SR1, FAULT CONTROL AND STATUS REGISTER

Bit	Name	Function	Access
2:0	FID	Fault ID from the OCU	R
3	TSA	TS access fault	R
4	ClrDTF	Clear x86 debug trap pending flag	W
5	ClrSSTF	Clear x86 single-step trap pending flag	W
6	FPF	FpOp fault	R
7	EF	OpQuad Sequence fault	R
10:8	IPFI	Instruction page fault information	R
11	DlyPG	Delay new CR0 PG bit effect	W
14:12	DPFI	Data page fault information	R
15	BIM	Burn-In Mode	R
19:16	DBN	x86 Data Break Point debug status	R
23:20	IBN	x86 Instruction Break Points debug status	R
26:24	SubOpcd	Sub-Opcode (MODR/M[5:3] from OpQuad Sequence Environment)	R
28:27	OCPL	Old CPL (from OpQuad Sequence Environment)	R
29	RBD	RAM BIST Disable	R
30	SSTF	x86 single-step trap pending flag	R
31	SDM	Select Direct Mapped	R

Table 2.7 SR2, INSTRUCTION AND PAGE FAULT REGISTER

Bit	Name	Function	Access
31:0	—	Logical address of last instruction fetch page fault	R

Table 2.8 SR3, PAGE FAULT REGISTER

BIT	NAME	FUNCTION	ACCESS
31:0	—	Logical address of last operand page fault	R

Table 2.9 SR4, FAULT PC REGISTER

BIT	NAME	FUNCTION	ACCESS
31:0	—	Logical address of last operand page fault	R

Table 2.10 SR5, CONFIGURATION REGISTER (IN RUX)

Bit	Name	Function	Access
0	L1ICD	L1 I-Cache disable	E/W
1	L1DCD	L1 D-Cache disable	E/W
2	L1CI	L1 Cache inhibit (TR12.CI)	E/W
3	DE	Debug extension enable (CR4.DE)	E/W
4	PSE	Page size enable (CR4.PSE)	E/W
5	WAD	Write allocate disable	E/W
6	PDD	Power down disable	E/W
7	NPFCD	NP freeze clock disable	E/W
8	SMO	Strong memory order	E/W
9	VSMO	Very strong memory order	E/W
10	SMCD	Self-Modifying Code trap disable	E/W
11	BPTD	Branch Prediction Table disable	E/W
12	BTBD	Branch Target Buffer disable	E/W
13	ROBD	RegOp bumping disable	E/W
14	LCKD	Lock disable	E/W
15	STQFD	STQ forward data disable	E/W
16	DCERLR	D-Cache enable random line replacement	E/W
17	DCSLD	D-Cache speculative load disable	E/W
18	ICERLR	I-Cache enable random line replacement	E/W
19	WBCD	Write back cache disable	E/W
20	SLDD	Speculative load disable	E/W
21	DTBDM	DTB direct mapped	E/W
22	DCDM	D-Cache direct mapped	E/W
23	ICDM	I-Cache direct mapped	E/W
28:24	REGN	Register number	E/W
29	RUYD	RUY disable	E/W

Table 2.10 SR5, CONFIGURATION REGISTER (IN RUX) (CONT)

Bit	Name	Function	Access
30	BPTNT	When BPTD=1, BPTNT indicates prediction direction; 1 = not taken, 0 = taken	E/W
31	ICPFD	I-Cache prefetch disable	E/W

Table 2.11 SR13, Instruction Decode Control Register

Bit	Name	Function	Access
7:0	SDD	Short decode disable bit mask	E/W
9:8	MDD	Multiple decode disable	E/W
10	LDD	Long decode disable	E/W
11	SetVEC1	Set “force HDD for one decode”	W
12	ExtExcpVEC	External OpQuad Sequence exception group	E/W
13	ESCDD	ESC (FPU) decode disable	E/W
14	MMXDD	MMX/3D decode disable	E/W
15	SD2D	OF opcode short decode disable	E/W
31:16	ExtVEC	External OpQuad Sequence decode group	E/W

Table 2.12 SR16, MMX/3D Status Bits Register

Bit	Name	Function	Access
7:0	MMXD	MMX/3D data register dirty bits	R
8	MMXSTC	MMX/3D store instruction committed	R
31:9	Reserved	—	—

.

Table 2.13 SR17 AND SR18, TIME STAMP CONTROL REGISTERS

Bit	Name	Function	Access
31:0	TSCL	Must synchronize read and write with TSCH to avoid overflow to TSCH.	R/W
63:32	TSCH	Must synchronize read and write with TSCL to avoid overflow from TSCL.	R/W

Table 2.14 SR21, CONFIGURATION REGISTER (IN RUX)

Bit	Name	Function	Access
0	Reserved	—	—
1	NAD	NA# Disable	E/W
2	SIE	Stop Interrupt Enable	E/W
3	FEEC	Force External OpQuad Sequence Cacheable	E/W
4	SSD	String SMI Disable	E/W
14:5	Reserved	—	—
15	INVC	INValidate Caches	E/W
16	WAE15M	Write Allocate Enable 15M-16M	E/W
23:17	WAELIM[6:0]	Write Allocate Enable Limit	E/W
24	PDED	Predecode Cache Disable	E/W
31:25	Reserved	—	—

Table 2.15 SR24, NP Presence and Opcode Register

Bit	Name	Function	Access
10:0	FpOpcd	NP opcode register	R/W
30:11	Reserved	—	—
31	NPNotPres	NP Not Present	R

Table 2.16 SR25, NP Code Selector Register

Bit	Name	Function	Access
15:0	FpOpcdSelNP	Code pointer (selector part)	R/W
27:16	Reserved	—	—
31:28	PrfxCnt	Prefix count from the OpQuad Sequence execution environment	R

Table 2.17 SR26, NP Code Offset Register

Bit	Name	Function	Access
31:0	FpCodeOffs	NP code pointer (offset part)	R/W

Table 2.18 SR27, NP DATA SELECTOR REGISTER

Bit	Name	Function	Access
15:0	FpDataSel	NP data pointer (selector part)	R/W
31:16	Reserved	—	—

Table 2.19 SR28, NP DATA OFFSET REGISTER

Bit	Name	Function	Access
31:0	FpDateOffs	NP data pointer (offset part)	R/W

Table 2.20 SR29, NPCFG (FPU Configuration) Register

Bit	Name	Function	Access
0	ClearBeforeException	Clears the Before Exception (stack fix up) bit in NP	E/W
1	Do Shared State	Allows "share-state" overlapping of p-ops which write their result one cycle after execution, with the following p-op (assuming dependencies are met).	E/W
2	Enable Hyper-Flg	Enables NPPop[12] (hyper flag). When not set, has the effect of always asserting NPPop[12].	E/W
3	StoreExMode	Used when issuing a "dummy" store (i.e., emulating an FST which does not store to memory). Specifically, inhibits hardware checking of result precision and rounding.	E/W
5:4	Reserved	—	—
6	FastFXCH	Enables single-pop FXCH mode	E/W
7	false_dependency_suppress	Enhancement to the share-state mechanism which eliminates false dependencies. See bit 1 below.	E/W
8	Disable0Cycle	Disables handling of 0-cycle Ops in NP	E/W
13:9	Reserved	—	—
14	mask_hyperterm	Disables hyper termination for the next p-op only. This bit then resets itself (internal to NP).	E/W
15	Reserved	—	E/W
16	Busy	Force pending error (i.e., always hyper terminate)	E/W
21:17	Reserved	—	E/W
22	UpperTSC-Word	Upper 16 bits of 32-bit Tag/Status/Control store will be filled with inverse value of this bit	E/W
31:23	Reserved	—	E/W

Table 2.21 LINE/SUBBLOCK L1 I-CACHE ORGANIZATION

Tag	Byte 31	Byte 30	…	Byte 1	Byte 0	MESI Bits	Subblock 0
Address	Byte 31	Byte 30	…	Byte 1	Byte 0	MESI Bits	Subblock 1

Table 2.22 DECODER OPQUAD LDOP AND STOP FORMAT

37 36	35 32	31 30	29 26	25 24	23 22	21 17	16	15 12	11 4	3 0
0 1	Type	ISF	Seg	ASz	DSz	Data	LD	Base	Disp8	Index

Table 2.23 LdOp and StOp Type(3:0) Field

Type(3:0)	Op Symbol	Type of LdOp or StOp to be Performed
0000	LD, LDL	Load integer data
0001	LDF	Load floating-point data
0010	LDST	Load integer data with store check
0011	LDM	Load MMX or 3D data
0100	CDAF(X)	CDA (see below) plus flush cache line(s)
0101	LDPF	Load Prefetch (prefetches a block)
0110	LDSTL	Load integer data with store check, locked
0111	LDMSTL	Load MMX or 3D data with store check, locked
1000	ST	Store integer data
1001	STF	Store floating-point data
1010	STUPD	Store integer data and update base register
1011	STM	Store MMX or 3D data
1100	CDA	Check “data” effective address (segment and page protection)
1101	CIA	Check “instruction” effective address (segment protection only)
1110	TIA	TLB invalidate address (based on TLB index only)
1111	LEA	Load effective address

Table 2.24 LDOP AND STOP ASZ(1:0) FIELD

ASz(1:0)	Address Calculation Size Before Environment Substitution	Address Calculation Size After Environment Substitution
00	Asize	2 Bytes
01	Ssize	—
10	4 Bytes	4 Bytes
11	Dsize	—

Table 2.25 DSz(1:0) Field for LdStOps other than LDF STF, LDM, & CDAF

DSz(1:0)	Data Size Before Environment Substitution	Data Size After Environment Substitution
00	1 Byte	1 Byte
01	2 Bytes	2 Bytes
10	4 Bytes	4 Bytes
11	Dsize	—

Table 2.26 DSz(1:0) Field for LdStOps LDF, STF, LDM & CDAF

DSz(1:0)	Data Size Before Environment Substitution	Data Size After Environment Substitution
00	FpDSize	—[a]
01	2 Bytes	2 Bytes
10	4 Bytes	4 Bytes
11	8 Bytes	8 Bytes

[a] LDF and STF only.

Table 2.27 LdOp and StOp Seg(3:0) Field

Seg(3:0)	Register	Description
0000	ES	x86 architectural ES register
0001	CS	x86 architectural CS register
0010	SS	x86 architectural SS register
0011	DS	x86 architectural DS register
0100	FS	x86 architectural FS register
0101	GS	x86 architectural GS register
0110	HS	microarchitectural temporary segment register
0111	—	reserved
100x	TS	"descriptor table segment register" for accessing the x86 architectural global and local descriptor tables (GDT and LDT respectively)
1010	LS	"linear segment register" (i.e., the segment base = 0)
1011	MS	"special memory" memory segment register (the special memory contains the "scratchpad" memory, as well as other special address spaces—I/O, cache flush, and special bus cycles)
11xx	OS	effective x86 architectural operand segment register

Table 2.28 LdOp and StOp ISF(1:0) Field

ISF(1:0)	Index Register Scale Factor
00	1 X
01	2 X
10	4 X
11	8 X

Table 2.29 LdOp and StOp LD (Large Displacement) Field

LD	Field to Use as the Displacement
0	Use the Disp8 field of this Op.
1	Use the 32-bit displacement from the appropriate decoder displacement bus

Table 2.30 DECODER OPQUAD REGOP FORMAT

37 36	35 30	29 26	25	24 20	21 17	16 12	11 10	9	8	7 0
0 0	Type	Ext	RX	DSz	Dest	Src1	—	SS	I	Imm8/Src2

Table 2.31 RegOp Type(5:1) Field General Arithmetic Operations

Type(5:1)	DSz Other Than 1 Byte	DSz = 1 Byte	cc-dep	RUX Only
00000	ADD, INC, CADD	ADD, INC	—	—
00001	MOV, OR, EOR	OR, EOR	—	O
00010	ADC	ADC	X	X
00011	SBB	SBB	X	X
00100	AND, EAND, BAND	AND, EAND	—	—
00101	SUB, ESUB, DEC, CSUB	SUB, DEC, ESUB	—	—
00110	XOR, EXOR	XOR, EXOR	—	—
00111	CMP	CMP	—	—

Table 2.32 REGOP TYPE(5:1) FIELD SHIFT AND MMX OPERATIONS

Type(5:1)	DSz Other Than 1 Byte	DSz = 1 Byte	cc-dep	RUX Only
01000	SLL, BLL	SLL	—	X
01001	SRL	SRL	—	X
01010	SLC, RLC	—	X	X
01011	SRC, RRC	—	X	X
01100	SLA	SLA	—	X
01101	SRA	SRA	—	X
01110	SLD, RLD	RLS	—	X
01111	SRD, RRD, MMX	RRS	—	—

Table 2.33 RegOp Type(5:1) Field for Other Arithmetic RegOps

Type(5:1)	DSz Other Than1 Byte Type(0)= 0	DSz Other Than1 Byte Type(0)=1	DSz = 1 Byte Type(0)= 0	DSz = 1 Byte Type(0)=1	cc-dep	RUX Only
10000	ZEXT8	SEXT8	—		—	X
10001	ZEXT16	SEXT16	—		—	X
10010	RDFLGS		DAA	DAS	X	X
10011	MOVcc	MOVcc	—		X	X
10100	MUL1S	MUL1U	—		—	X
10101	MULEH	MULEL	—		—	X
10110	DIV1	DIV2	—		—	X
10111	DIVER	DIVEQ	—		—	X

Table 2.34 REGOP TYPE(5:1) FIELD FOR SPECIAL REGOPS

Type(5:1)	DSz Other Than 1 Byte Type(0)= 0	DSz Other Than 1 Byte Type(0)=1	DSz = 1 Byte Type(0)= 0	DSz = 1 Byte Type(0)=1	cc-dep	RUX Only
11000	RDxxx	RDxxx	—		—	X
11001	RDFLG	BSWAP	—		—	X
11010	RDSEG	RDSEG	—		X	X
11011	—		—		—	—
11100	WRDR	WRDL	—		—	X
11101	WRxxx	WRxxx	—		—	X
11110	WRIP	WRDLIP	—		—	X
11111	CHKS	WRDH	—		—	X

Table 2.35 REGOP EXT(3:0) FIELD

Type of Ops	Field Combination	Used to Specify
MOVcc	Type[0] Ext[3:0]	a 5-bit condition code
RDxxx and WRxxx	Type[0] Ext[3:0]	a 5-bit special register number
RDSEG	Type[0] Ext[3:0]	a 5-bit segment descriptor register number
(other) Ops with SS = 1	Ext[3:0]	four status flag modification bits stored in the scheduler
(other) Ops with SS = 0	Ext[3:0]	not used

Table 2.36 RegOp DSz(2:0) Field

DSz(2:0)	Data Size Before Environment Substitution	Data Size After Environment Substitution
000	1 Byte	1 Byte
001	2 Bytes	2 Bytes
010	4 Bytes	4 Bytes
011	Dsize	—
100	Asize	—
101	Ssize	—
110	—	—
111	—	—

Table 2.37 RegOp I (Immediate) Field

I Field	Imm8/Src2	Source for Second Operand
0	Imm8/Src2[4:0]	a general register to be the source register Src2
1	Imm8/Src2[7:0]	an 8-bit immediate value that is extended to the DSz size

Table 2.38 DECODER OPQUAD SPECOP FORMAT

37 35	34 31	30 26	25 24	23 22	21 17	16 0
1 0 1	Type	CC	—	DSz	Dest	Imm17

Table 2.39 SpecOp Type(3:0) Field

Type(3:0)	Op Symbol	Type of SpecOp
00xx	BRCOND	Branch condition
0100	LDDHA	Load default handler address
0101	LDDHAB	Load binary default handler address
0110	LDAHA	Load alternate handler address
0111	LDAHAB	Load binary alternate handler address
1000	LDK	Load constant
1001	FPOP	Floating-point Op
1010	LDKD	Load DSz-modified constant
1011	FPOPE	Floating-point Op from the OPQuad Sequence Environment
11xx	FAULT	Unconditional fault

Table 2.40 SpecOp CC(4:1) Field

CC(4:1)	Mnemonic	Condition to be Tested	Usage
0000	True	1	Always TRUE
0001	ECF	ECF	OpQuad Sequence Carry Flag
0010	EZF	EZF	OpQuad Sequence Zero Flag
0011	SZnZF	EZF | ~ZF	Early termination of string instructions due to debug trap or hardware interrupt
0100	MSTRZ	~EZF & ~IP & ~ (DTF | SSTF | MDD)	String instruction exit condition
0101	STRZ	~EZF & ~IP & ~ (DTF | SSTF | MDD)	String instruction exit condition
0110	MSTRC	~EZF & ~IP & ~ (DTF | SSTF | MDD)	String instruction exit condition
0111	STRZnZF	~EZF & ~IP & ~ (DTF | SSTF | MDD) & ZF	String instruction exit condition
1000	OF	OF	Overflow Flag
1001	CF	CF	Carry Flag
1010	ZF	ZF	Zero Flag
1011	CvZF	CF | ZF	Used for “below or equal”, “not below or equal”, “above”, “not above” conditions
1100	SF	SF	Sign Flag
1101	PF	PF	Parity Flag
1110	SxOF	SF ^ 0F	Used for “less”, “not less”, “greater or equal”, “not greater or equal” conditions
1111	SxOvZF	(SF ^ 0F) | ZF	Used for “greater”, “not greater”, “less or equal”, “not less or equal” conditions

:

Table 2.41 SPECOP DSZ(1:0) FIELD

DSz(1:0)	Data Size Before Environment Substitution	Data Size After Environment Substitution
00	1 Byte	1 Byte
01	2 Bytes	2 Bytes
10	4 Bytes	4 Bytes
11	Dsize	—

Table 2.42 DECODER OPQUAD LIMM OP FORMAT

37 36	35 21	20 17	16 0
1 1	ImmHi	Dest	ImmLo

Table 2.43 PIPELINE NOTATIONAL CORRESPONDENCE

Figure 2.16 and Figure 2.19	Figure 2.12, Figure 2.14 and Figure 2.18
S0	Operand Fetch Stage
S1	Execution Stage 1
S2	Execution Stage 2
C	Commit Stage