

%st(7)

%cr3control registers  
%db0%db7debug registers  
%tr3%tr7test registers









fadd/faddp/fiaddAdd

faddm32real

faddm64real

faddST(i),ST

faddST,ST(i)

faddpST,ST(i)

fadd

fiaddm32int

fiaddm16int

fbldfbldm80decLoad Binary Coded Decimal

fbstpfbstp80decStore Binary Coded Decimal and  
Pop

fchsfchsChange Sign

fclex/fnclexClear Exceptions

fclex

fnclex

fcom/fcomp/fcomppCompare Real

fcomm32real

fcomm64real

fcomST(i)

fcom

fcompm32real

fcompm64real

fcompST(i)

fcomp

fcompp

fcosfcosCosine

fdecstpfdecstpDecrement Stack-Top Pointer

fdiv/fdivp/fidivDivide

fdivm32real

fdivm64real

fdivST(i),ST

fdivST,ST(i)

fdivpST,ST(i)

fdiv

fidivm32int

fidivm16int

fdivr/fdivpr/fidivrReverse Divide

ffreeffreeST(i)Free Floating-Point Register

ficom/ficompCompare Integer

ficomm16real

ficomm32real

ficompm16int

ficompm32int

fildfldsm16intLoad Integer

fildlm32int

fildqm64int

fincstpfincstpIncrement Stack-Top Pointer

finit/fninitfinitInitialize Floating-Point Unit

fninit

fist/fistpfistsm16intStore Integer

fistlm32int

fistpsm16int

fistplm32int

fistpqm64int

fldfldsm32realLoad Real

fldlm64real

fldtm80real

fldST(i)

fld1/fldl2t/fldl2e/fldpi/fldlg2/gldln2/fldzLoad Constant

fld1

fld2t

fld2e

fldpi

fldlg2

fldln2

fldz

fldcwfldcwm2byteLoad Control Word

fldenvfldenvm14/28byteLoad FPU Environment

fmul/fmulp/fimulMultiply

fmulm32real

fmulm64real

fmulST(i),ST

fmulST(i),ST

fmulpST,ST(i)

fmul

fprem1fprem1Partial Remainder

fptanfptanPartial Tangent

frndintfrndintRound to Integer

fstorfstorm94/108byteRestore FPU State

fsave/fnsaveStore FPU State

fsavem94/108byte

fnsavem94/108byte

fscalefscaleScale

fsinfsinSine

fsincosfsincosSine and Cosine

fsqrtfsqrtSquare Root

fst/fstpfstm32realStore Real

fstm64real

fstST(i)

fstpm32real

fstpm64real

fstpm80real

fstpST(i)

fstcw/fnstcwStore Control Word

fstcwm2byte

fnstcwm2byte

fstenv/fnstenvStore FPU Environment

fstenvm14/28byte

fnstenvm14/28byte

fstsw/fnstswStore Status Word

fstswm2byte

fstsw%ax

fnstswm2byte

fnstsw%ax

fsub/fsubp/fisubSubtract

fsubm32real

fsubm64real

fsubST(i),ST

fsubST,ST(i)

fsubpST,ST(i)



fsubprST,ST(i)

fsubr

fsubrm32int

fsubrm16int

ftstftstTest

fucom/fucomp/fucomppUnordered Compare Real

fucomST(i)

fucom

fucompST(i)

fucomp

fucompp

fwaitfwaitWait

fxamfxamExamine

fxchfxchST(i)Exchange Register Contents

fxch

fxtractfxtractExtract Exponent and  
Significand

fyl2xfyl2xCompute y







3,r32  
mov%dr6/%dr7,r32  
movr32,%dr03  
movr32,%dr6/%dr7  
mov%tr4/%tr5/%tr6/%tr7,r32  
movr32,%tr4/%tr5/%tr6/%tr7  
mov%tr3,r32  
movr32,%tr3

movs/movsb/movsw/movsdMove Data from String to String  
movsm8,m8  
movsm16,m16  
movsm32,m32  
movsb  
movsw  
movsd

(optional forms with segment override)  
movsb%seg:0(%esi),%es:0(%edi)  
movsw%seg:0(%esi),%es:0(%edi)  
movsd%seg:0(%esi),%es:0(%edi)

movsxmovsxr/m8,r16Move with Sign-Extend  
movsxr/m8,r32  
movsxr/m16,r32

movzxmovzxr/m8,r16Move with Zero-Extend  
movzxr/m8,r32  
movzxr/m16,r32

mulmulr/m8,%alUnsigned Multiplication of AL  
mulr/m16,%axor AX  
mulr/m32,%eax

negnegr/m8Two's Complement Negation  
negr/m16  
negr/m32

nopnopNo Operation

notnotr/m8One's Complement Negation

orr16,r/m16  
orr32,r/m32  
orr/m8,r8  
orr/m16,r16  
orr/m32,r32

outout%al,\$imm8Output to Port  
out%ax,\$imm8  
out%eax,\$imm8  
out%al,%dx  
out%ax,%dx  
out%eax,%dx

outs/outsb/outsw/outsdOutput String to Port  
outsr/m8,%dx  
outsr/m16,%dx  
outsr/m32,%dx  
outsb  
outsw  
outsd

poppopm16Pop a Word from the Stack  
popm32  
popr16  
popr32  
pop%ds  
pop%es  
pop%ss  
pop%fs  
pop%gs

popa/popadPop all General Registers  
popa  
popad

popf/popfdpopfPop Stack into FLAGS or  
popfdEFLAGS Register

pushpushm16Push Operand onto the Stack  
pushm32  
pushr16  
pushr32  
push\$imm8  
push\$imm16  
push\$imm32  
pushSreg

pusha/pushadPush all General Registers

rcl%cl,r/m8  
rcl\$imm8,r/m8  
rcll,r/m16  
rcl%cl,r/m16  
rcl\$imm8,r/m16  
rcll,r/m32  
rcl%cl,r/m32  
rcl\$imm8,r/m32  
rcr1,r/m8  
rcr%cl,r/m8  
rcr\$imm8,r/m8  
rcr1,r/m16  
rcr%cl,r/m16  
rcr\$imm8,r/m16  
rcr1,r/m32  
rcr%cl,r/m32  
rcr\$imm8,r/m32  
roll,r/m8  
rol%cl,r/m8  
rol\$imm8,r/m8  
roll,r/m16  
rol%cl,r/m16  
rol\$imm8,r/m16  
roll,r/m32  
rol%cl,r/m32  
rol\$imm8,r/m32  
ror1,r/m8  
ror%cl,r/m8  
ror\$imm8,r/m8  
ror1,r/m16  
ror%cl,r/m16  
ror\$imm8,r/m16  
ror1,r/m32  
ror%cl,r/m32  
ror\$imm8,r/m32

rdmsrrdmsrRead from Model-Specific  
Register (Pentium-specific)

rdstcrdstcRead from Time Stamp Counter  
(Pentium-specific)

rep/repe/repz/repne/repnzRepeat Following String  
rep ins%dx,rm8Operation  
rep ins%dx,rm16  
rep ins%dx,rm32  
rep movsm8,m8  
rep movsm16,m16

rep stosm32  
repe cmpsm8,m8  
repe cmpsm16,m16  
repe cmpsm32,m32  
repe scasm8  
repe scasm16  
repe scasm32  
repne cmpsm8,m8  
repne cmpsm16,m16  
repne cmpsm32,m32  
repne scasm8  
repne scasm16  
repne scasm32

retretReturn from Procedure  
ret\$imm16

rsmrsmResume from System-  
Management Mode  
(Pentium-specific)

sahfsahfStore AH into Flags

sal/sar/shl/shrShift Instructions

sal1,r/m8  
sal%cl,r/m8  
sal\$imm8,r/m8  
sal1,r/m16  
sal%cl,r/m16  
sal\$imm8,r/m16  
sal1,r/m32  
sal%cl,r/m32  
sal\$imm8,r/m32  
sar1,r/m8  
sar%cl,r/m8  
sar\$imm8,r/m8  
sar1,r/m16  
sar%cl,r/m16  
sar\$imm8,r/m16  
sar1,r/m32  
sar%cl,r/m32  
sar\$imm8,r/m32  
shl1,r/m8  
shl%cl,r/m8  
shl\$imm8,r/m8  
shl1,r/m16  
shl%cl,r/m16  
shl\$imm8,r/m16



shr1,r/m32  
shr%cl,r/m32  
shr\$imm8,r/m32

sbbsbb\$imm8,r/m8Integer Subtraction with Borrow  
sbb\$imm16,r/m16  
sbb\$imm32,r/m32  
sbb\$imm8,r/m16  
sbb\$imm8,r/m32  
sbb8,r/m8  
sbb16,r/m16  
sbb32,r/m32  
sbb/m8,r8  
sbb/m16,r16  
sbb/m32,r32

scas/scasb/scasw/scasdCompare String Data  
scasm8  
scasm16  
scasm32  
scasb  
scasw  
scasd

(optional forms with segment override)  
scasb%al,%seg:0(%edi)  
scasw%ax,%seg:0(%edi)  
scasd%eax,%seg:0(%edi)

setccByte Set on Condition  
setar/m8above  
setaer/m8above or equal  
setbr/m8below  
setber/m8below or equal  
setcr/m8carry  
seter/m8equal  
setgr/m8greater  
setger/m8greater or equal  
setlr/m8less  
setler/m8less or equal  
setnar/m8not above  
setnaer/m8not above or equal  
setnbr/m8not below  
setnber/m8not below or equal  
setncr/m8not carry  
setner/m8not equal  
setngr/m8not greater  
setnger/m8not greater or equal

setpor/m8parity odd  
setsr/m8sign  
setzr/m8zero

sgdt/sidtsgdtmStore Global/Interrupt  
sidtmDescriptor Table Register

shldshld\$imm8,r16,r/m16Double Precision Shift Left  
shld\$imm8,r32,r/m32  
shld%cl,r16,r/m16  
shld%cl,r32,r/m32

shrdshrd\$imm8,r16,r/m16Double Precision Shift Right  
shrd\$imm8,r32,r/m32  
shrd%cl,r16,r/m16  
shrd%cl,r32,r/m32

sldtsldtr/m16Store Local Descriptor Table  
Register

smswsmswr/m16Store Machine Status Word

stcstcSet Carry Flag

stdstdSet Direction Flag

stistiSet Interrupt Flag

stos/stosb/stosw/stosdStore String Data  
stosm8  
stosm16  
stosm32  
stosb  
stosw  
stosd

(optional forms with segment override)  
stosb%al,%seg:0(%edi)  
stosw%ax,%seg:0(%edi)  
stosd%eax,%seg:0(%edi)

strstrr/m16Store Task Register

subsub\$imm8,r/m8Integer Subtraction  
sub\$imm16,r/m16  
sub\$imm32,r/m32  
sub\$imm8,r/m16  
sub\$imm8,r/m32

test\$imm32,r/m32  
testr8,r/m8  
testr16,r/m16  
testr32,r/m32

verr, verwverrr/m16Verify a Segment for Reading or  
verwr/m16Writing

waitwaitWait

wbinvdwbinvdWrite-Back and Invalidate  
Cache (i486-specific)

wrmsrwmsrWrite to Model-Specific  
Register (Pentium-specific)

xaddxaddr8,r/m8Exchange and Add  
xaddr16,r/m16(i486-specific)  
xaddr32,r/m32

xchgxchgr16,%axExchange Register/Memory  
xchg%ax,r16with Register  
xchg%eax,r32  
xchgr32,%eax  
xchgr8,r/m8  
xchgr/m8,r8  
xchgr16,r/m16  
xchgr/m16,r16  
xchgr32,r/m32  
xchgr/m32,r32

xlat/xlatbxlatm8Table Look-up Translation  
xlatb

xorxor\$imm8,r/m8Logical Exclusive OR  
xor\$imm16,r/m16  
xor\$imm32,r/m32  
xor\$imm8,r/m16  
xor\$imm8,r/m32  
xorr8,r/m8  
xorr16,r/m16  
xorr32,r/m32  
xorr/m8,r8  
xorr/m16,r16  
xorr/m32,r32