

# **ADVANCED 80386 PROGRAMMING TECHNIQUES**

James L. Turley

Osborne McGraw-Hill  
Berkeley, California

# CONTENTS

Introduction		xiii
<b>Chapter 1</b>	<b>BASICS</b>	
80386 Predecessors	3	
Overview of the 80386	8	
Byte Ordering	12	
Data Types	14	
Register Set	18	
Instruction Set	34	
Memory Segmentation	38	
<b>Chapter 2</b>	<b>MEMORY SEGMENTATION</b>	45
Definition of a Segment	45	
Segment Selectors	62	
Local Descriptor Tables	71	
Segment Aliasing, Overlapping, and More	74	
Examples	78	
<b>Chapter 3</b>	<b>PRIVILEGE LEVELS</b>	85
Privilege Protection	86	
Defining Privilege Levels	88	
Changing Privilege Levels	96	
Defensive Programming	112	

<b>Chapter 4</b>	<b>PAGING</b>	<b>117</b>
	Required Elements	119
	Address Translation	120
	Enabling and Disabling Paging	142
	Page-Level Protection	144
	Paging MMU Cache	148
	Demand Paging and Virtual Memory Management	150
	Using Segmentation and Paging Together	160
	Examples	163
<b>Chapter 5</b>	<b>MULTITASKING</b>	<b>169</b>
	Task State Segments	173
	Moving Between Tasks	179
	Task Scheduling	186
	Busy Bit, NT Bit, Back Link Field, TS Bit	192
	Extensions to the TSS	193
	I/O Permission Bit Map	194
	Changing Privilege Levels Within a Task	196
	Changing LDTs	197
<b>Chapter 6</b>	<b>COMMUNICATING AMONG TASKS</b>	<b>199</b>
	Total Separation of Address Spaces	200
	Total Overlap of Address Spaces	201
	Partial Overlap of Address Spaces	203
	Maintaining Aliases	212
	Modifying Unshared Data	218
<b>Chapter 7</b>	<b>HANDLING FAULTS AND INTERRUPTS</b>	<b>221</b>
	Exception Processing	225
	The Interrupt Descriptor Table	229
	Deciding on the Type of Gate to Use	232
	Exception Handlers and Privilege Levels	238
	Error Codes	240
	Responsibilities of an Exception Handler	243
	Halt and Shutdown	245
	Using Exceptions for Emulation	246
	Exception Reference	248

<b>Chapter 8</b>	<b>80286 EMULATTON</b>	<b>259</b>
Operand Sizing	261	
Address Sizing	264	
Stack Sizing	266	
Branching Between 16-bit and 32-bit Code	268	
Calling Between 16-bit and 32-bit Code	270	
16-bit Interrupt Handlers	276	
16-bit Descriptors	277	
16-bit Tasks	277	
80286 Compatibility	279	
<b>Chapter 9</b>	<b>8086 EMULATION</b>	<b>283</b>
Virtual 8086 Tasks	285	
Memory Segmentation	286	
Privilege and Protection	288	
80386 Enhancements	291	
Entering Virtual 8086 Mode	294	
Interrupts and Exceptions	295	
Using IOPL for Emulation	306	
Using Paging in Virtual 8086 Mode	307	
Differences from a Real 8086	311	
<b>Chapter 10</b>	<b>DEBUGGING</b>	<b>317</b>
Memory Segmentation	319	
Single-Stepping	323	
Software Breakpoints	326	
Break on Task Switch	329	
Debug Registers	330	
Hardware Instruction Breakpoints	333	
Multiple Breakpoint Hits	349	
Debug Register Access Protection	350	
Debugging with a Separate Task	351	
<b>Chapter 11</b>	<b>THE 80387 NUMERIC PROCESSOR EXTENSION</b>	<b>355</b>
Why Floating Point?	856	
The 80387 Register Set	358	
The 80387 Number System	365	
The Processor-to-Coprocessor Interface	377	
Differences Among the 80387, 80287, and 8087	380	

<b>Chapter 12</b>	<b>PROGRAMMING FOR PERFORMANCE</b>	<b>385</b>
	Optimal Addressing	387
	Using Registers	390
	Instruction Execution Time	391
<b>Chapter 13</b>	<b>RESET AND REAL MODE</b>	<b>399</b>
	Resetting the 80386	399
	Real Mode Operation	407
	Preparing for Protected Mode	414
	Returning to Real Mode	421
	Testing the TLB	422
<b>Chapter 14</b>	<b>HARDWARE</b>	<b>429</b>
	The Microprocessor	430
	Address Decoding	436
	Different Kinds of Memory	440
	Bus Locking and Read-Modify-Write Cycles	446
	Interrupt Acknowledge Cycles	448
	Self-Test Request and Coprocessor Sense	449
<b>Appendix A</b>	<b>GLOSSARY</b>	<b>451</b>
<b>Appendix B</b>	<b>NEW INSTRUCTIONS</b>	<b>459</b>
<b>Appendix C</b>	<b>MEMORY STRUCTURES AND 80386 REGISTERS</b>	<b>477</b>
	Segment Descriptors	478
	Task State Segments	481
	Page Descriptors	483
	Selected 80386 Registers	484
<b>Appendix D</b>	<b>EVOLUTION OF THE 80386</b>	<b>487</b>
	A1 Stepping	487
	B0 Stepping	492
	Bi Stepping	494
<b>Index</b>		<b>501</b>

---

---

# INTRODUCTION

*We still have judgment here that we but teach bloody  
instructions, which, being taught, return  
To plague the inventor*

—William Shakespeare, *Macbeth*

This book shows how a programmer can make use of the newest and most advanced features of the 80386 microprocessor. Since many of the most powerful features of the 80386 can be controlled only at the machine level, it is assumed that the reader is already familiar with the concepts of assembly language programming. Furthermore, a familiarity with Intel microprocessors in general, and the 8086 and/or 80286 in particular, will be helpful. Throughout the book, parallels are drawn between the 80386 and earlier members of the iAPX 86 family. Based on this foundation, the book progresses through the features of the 80386, showing how improved or unique capabilities can be used to build a truly powerful microcomputer system.

A conscious effort has been made to avoid assumptions about the reader's hardware (or lack thereof). In particular, there is no reference to MS-DOS or to "PC-compatible" hardware features that may not exist on your system. Therefore, readers who are developing software for custom hardware, workstations, intelligent controllers, or other applications need not be continually disappointed by examples or descriptions that they cannot use. The concepts and techniques discussed in this book are relevant to any

advanced 80386 programmer, regardless of the hardware platform.

Several diagrams and memory maps have been strategically placed throughout the book to illustrate various points in the text. To cut down on confusion, all memory maps have been drawn in the same way. That is, low memory addresses are always toward the bottom of the page, and high addresses grow toward the top of the page. When data values are illustrated horizontally, the most significant bit, byte, or address is always to the left, with the least significant end to the right. Because this is a programmer's book, numeric values that are not spelled out are usually given in hexadecimal (base sixteen) rather than "normal" decimal (base ten) form. Occasionally, numbers are given in two or more radices if it will make things clearer.

I wrote this book for two reasons. First, after spending almost two years developing software for the 80386, I felt that the vast majority of 80386-based machines on the market completely wasted the unique power of the processor. Serious 32-bit computers were operating in "brain-dead" mode because of old software and old hardware architecture. I hoped that by generating enthusiasm for Protected mode 80386 programming I could help us all to benefit from some truly amazing microcomputer horsepower. Second, like innumerable others, I have read technical books that either were so badly written that reading them was painful, or were well written but contained no information of value. So, like any novice, I figured that I could do better. The result, you now hold in your hands; I am much the wiser and more humble for it.

The text of this book was written entirely by (left) hand with a Cross mechanical pencil on a ruled tablet. It was then entered into an aging PC-compatible computer running Microsoft Word version 3.00. The program source code was written with PC/VI from Custom Software Systems and assembled and linked with the excellent tools available from Phar Lap Software. The majority of these programs were then tested on a CPU-386 from Force Computers, as well as a Compaq Deskpro 386 and a PC's Limited 386-16.

It has been said that there are two kinds of programmers: those who say they hate Intel processors, and liars. Be that as it may, the 80386 opens up whole new applications to the programmer who is willing to tackle it. Once you understand how the 80386 works, and what it can do, I think you'll be as excited about it as I am.

—James L. Turley

# INDEX

---

## A

- Aborts, 224
- Accessed bit, 52, 217
- Access Rights byte, 55
- Accumulator, 451
- Address
  - bus, 4, 432-434, 451
  - decoding, 436-440
  - map, after reset, 407
  - sizing (80286 emulation), 264-266
- Addressable space, 38
- Addresses, 12-14
  - translating, 120-142
- Addressing
  - optimal, 387-390
  - real mode, 401-411
- Addressing modes, 42
- Address space
  - partial overlap of, 203
  - total overlap of, 201-202
  - total separation of, 200-201
- Affine closure, 376
- Aliases, maintaining, 212-217
- Aliasing, 451
  - page-level, 141
  - segment, 74-83

- Aligned transfers, 433
- AMD processors, 18
- AND versus BTR, 393
- AI stepping, 487-492
- Architectures, instruction set, 2
- Arrays, vector. 18
- Attributes
  - privilege level, 85-115
  - segment, 47
- Automatic stack references, 267
- Auxiliary flag, 22

## B

- BO stepping, 492-494
- Bi stepping, 494-498
- Back Link field, 184, 192-193
- Base Address field, 209
- BCD numbers, 16
- Benchmarking factors, 386
- Binary compatibility, 7
- Binary numbers, format of, 16
- Bit-fields, segment descriptor, 51-52
- Bit map, I/O permission, 194-195
- Bit-wise instructions, 37
- Boundaries, aligning segment with page, 163

Bound error, 250  
 Break on task switch, 329-330  
 Breakpoint error, 250  
 Breakpoints, 451
 

- hardware data reference, 343-349
- hardware instruction, 333-343
- multiple exception status bits, 349-350
- selective, 341-343
- software, 326-328
- task-specific, 340

 BTC versus XOR, 395  
 BTR versus AND, 393  
 Bus
 

- address, 4, 432-434
- data, 4, 431-432

 Bus locking, 446-448  
 Busy bit, 192-193  
 Byte, 451
 

- ordering, 12-14

## C

Cache, 442-443, 452
 

- paging MMU, 148-149

 Call gates, 98-106
 

- 80286 emulation and, 274-276

 Calling between 16-bit and 32-bit code, 270-276  
 Call instructions, 98-106  
 Carry flag, 21  
 CBW versus MOVSB, 397  
 CISC processors, 2, 452  
 Clock frequency, 12  
 Code
 

- alignment, 389
- branching between 16-bit and 32-bit, 268-270
- executing 80286, 259-282
- references, privileged, 94-95
- sharing, 204-205

 Codes, error, 240-242  
 Compatibility
 

- binary, 7
- 80286, 279

Complex instruction set architecture, 2  
 Conditional Jumps, 393  
 Conforming code segments, 97-98  
 Context switch, 171-172  
 Contiguous linear addresses, 139  
 Control registers, 9, 26-28, 484  
 Control word, 80387, 359  
 Conversion instructions, 36  
 Coprocessor, 452
 

- automatic detection of 80387, 402-403, 449
- error, 256
- interface, 377-380
- saving the state of a, 193
- Segment Overrun error, 252

 CPL, 452  
 CWDE versus MOVSB, 397

## D

Data
 

- bus, 4, 431-432, 452
- operand alignment, 345-346
- reference breakpoints, 343-349
- references, privileged, 92-93
- segments, 56
- sharing, 207-211
- types, 14-18
- unshared, 218-220

 Debug
 

- exception, 249

 Debugging, 317-353  
 Debug registers, 9, 28-32, 330-333, 485
 

- access protection, 350-351

 Decimal point placement, 80387, 369  
 Decoding logic, 437  
 Default bit, 53  
 Demand paging, 11, 150-159  
 Denormal numbers, 376  
 Descriptor Privilege Level, 88-89  
 Descriptors, 452
 

- aliased, 212-217
- differences in 80286, 280

Descriptors, *continued*  
 gate, 230-238  
 multiple tasks and, 199-200  
 page, 483  
 page directory entry, 124-127  
 page table entry, 126-127  
 pairing, 213  
 segment, 48-59, 478-480  
 16-bit, 277  
 task state segment, 174-175,  
 450-482  
 Descriptor tables, 59-61  
 interrupt, 227, 229-232  
 local, 71-74  
 Device Not Available error, 251  
 Direction flag, 23  
 Direct mapping, 141  
 Disabling paging, 142-144  
 Divide Error, 248  
 Double Fault error, 251-252  
 Double-precision shift  
 instructions, 38  
 DPL, 452  
 fields, 208  
 DRAM, 441, 453  
 Dwords, 15, 453  
 Dynamic RAM, 441

## E

EFLAGS register, 21-24, 484  
 checking, 114  
 8086/8088 microprocessors, 4-5  
 8086 emulation, 283-315  
 differences from real 8086,  
 311-315  
 8086, Virtual mode, 8-9  
 80387  
 automatic detection, 402-403  
 differences with 80287  $\overline{\text{and}}$   
 8087, 380-383  
 numeric processor extension,  
 355-383  
 80286 emulation, 259-282

## Emulation

8086, 283-315  
 80286, 259-282  
 using exceptions for, 246  
 Enabling  
 paging, 142-144, 419  
 Protected mode, 419  
 EPROM, 453  
 Error codes, 240-242  
 Errors, stack, 321-323  
 Evolution of 80386, 487-498  
 Exception  
 stacks, 228  
 vectors, 226  
 Exceptional conditions, 221-257  
 Exception processing, 225-229  
 privilege levels and, 238-239  
 real mode, 411-413  
 responsibilities of, 243-245  
 using for emulation, 246  
 Virtual 8086 mode and,  
 295-305  
 Exceptions, 453  
 debug, 350-351  
 software, 222-223  
 summary of predefined,  
 248-257  
 Execution time, 391-398  
 Explicit stack references, 267  
 Extended real format, 17

## F

Family tree, 80386, 6  
 FAR CALL, 98-106  
 FAR JMP, 103-106  
 Far pointer, 15  
 Faults, 224  
 using as debugging tools, 321  
 Fields, bit, 51-52  
 Floating point  
 data registers, 364  
 numbers, 17  
 operations, 356-357

## Formats

- data, 14-18
- floating point number, 365-377

Fragmented memory space, 138

Functions, paging, 119-120

## G

Gate descriptors, 230-238

- 16-bit, 274

Gates, 453

- task, 180

GB, 453

GDT, 453. *See also* Descriptor

- Tables

GDTR, 453

General Protection Fault error, 255

Global descriptor table, 60, 418

Granularity bit, 54-55

## H

Halt condition, 245-246

Hardware, 429-450

- data reference breakpoints, **343-349**

- initiated interrupts, 234

- instruction breakpoints, **333-343**

- interrupts, 222-223

Hits, breakpoint, 349-350

## I

iAPX 86 family of microprocessors, 5

IC, 453

Identity mapping, 141

IDT, 453. *See also* Descriptor tables

IDTR, 454

IEEE standard for numeric processor, 357

Implicit stack references, 267

Indefinite numbers, 377

Indexing, scaled, 35

Infinity, 376

Input/Output Privilege Level flag, 23. *See also* IOPL

Instruction

- pipeline, 387
- prefetch, 390
- set, 10, 34-38

Instructions

- breakpoint, 326-328, 333-343
- execution time of, 391-398
- fetching first, 403-406
- IOPL-sensitive, 91
- new 80386, 459-475
- privileged, 87, 90-92
- real mode, 408-409

Integer formats, 17

Integer multiply instruction, 37

Intel microprocessors, history of, 3-8

Interface, processor to coprocessor, 377-380

Interleaved alias information, 215

Interlevel stack switch, 109

Interrupt

- acknowledge cycles, 448-449
- descriptor table, 60, 227, 418
- flag, 23
- gates, 230, 234-235
- latency, 113

Interrupt service routines, 229

- task scheduling and, 188-190

- Virtual 8086 mode, 300-301

Interrupt-handlers, 16-bit, 276

Interrupts, 221-224, 454

- disabling, 143

- Virtual 8086 mode, 295-305

Intersegment, 454

Intrasegment, 454

Invalid Opcode error, 251

Invalid TSS error, 253

I/O instructions, 36

I/O permission bit map, 194-195

IOPL, 454

- using for emulation, 306-307

IOPL-sensitive instructions. 91

## K

KB, 454

## L

LDT, 454. *See also* Descriptor tables

LDTR, 455

LEA versus MUL, 396

Limit fields, 209

Linear addresses, 455

80286 and, 279

translating logical to, 121

translating to physical, 131

Virtual 8086 mode and, 309

Lists, alias, 212

Load full pointer instruction, 37

Loading segment selectors, 66-69

Local descriptor table, 60, 71-74

Locality of reference, 442

LOCK prefix, 252, 445

Logical addresses, 39, 455

translating to linear, 121

Long integer format, 17

Long real format, 17

Loop counter, 36

Loop/iteration counters, 112

## M

Machine Status Word, 119

Mapping

direct or identity, 141

linear address, 134

Maskable hardware interrupts, 223

MB, 455

Memory

addressable space, 38

allocating for demand paging, 154

cache, 442-443

different kinds of, 440-446

paging, 120-142

reallocated, 156

remapping pages of, 157

Memory, *continued*

restoring reallocated, 159

sharing, 199-220

virtual, 150-159

Virtual 8086 mode and

segmentation of 286-288

wait states, 443-446

Memory management unit. *See* MMU

Memory maps, 166, 436-440

Memory references, verifying, 113

Memory segmentation, 10, 38-43, 45-84

using as debugging tool, 319-322

Memory structures, 80386, 477-454

Microcode, 455

Microprocessor

hardware details of, 430-436

history of, 1

predecessors to 80386, 3-8

MMU, 455

cache, paging, 148-149

features of, 7

integration of, 118

paging, 11

translation lookaside buffer, 32

using, for paging, 117-168

Mnemonic, 456

Modes

addressing, 42

operating, 8-9

Modifying unshared data, 218-220

Motorola processors, 18

MOV 36, 396

versus XOR, 393

MOVSX versus CBW and CWDE, 397

MSW, 456

Multiply and divide versus shift, 397

Multitasking, 169-197, 12. *See also* Tasks

preparing for, 421

Multuser versus multitasking, 173  
 MUL versus LEA, 396

## N

National Semiconductor  
 processors, 18  
 Near pointer, 15  
 Nested Task flag, 23  
 Nested tasks, 184-186  
 NMI, 456  
   disabling, 143  
   error, 249  
 Non-maskable hardware  
   interrupts, 223. *See also* NMI  
 NPX, 456. *See also* Numeric  
   processor *and* 80387  
 NT bit, 192-193  
 Null descriptor, 64  
 Null selectors, 115  
 Numbers  
   denormal, 376  
   indefinite, 377  
   special, 375-377  
 Number system, 80387, 365-377  
 Numeric processor extension,  
   355-383. *See also* Coprocessor  
   *and* 80387

## O

Offsets, segment, 50  
 Opcodes, 456  
   previously undefined, 281  
 Operand sizing (80286 emulation),  
   261-264  
 Operating modes, 8-9  
 Optimal addressing, 387-390  
 Overflow flag, 23, 250  
 Overlapping  
   address spaces, 201-202  
   segment, 74-83  
 Overview, 80386. 8-12

## P

### Page

  descriptors, 483  
   directory entry descriptor,  
     124-127  
   Fault error, 256  
   faults, 154, 242-243  
   frames, 130  
   protection, segment versus, 123  
   table entry descriptor, 126-127  
   tables, 132-142  
   translation, using, 127-132

Page-level aliasing, 141

Page-level protection, 144-147

Pages, 456

  remapping, 157  
   restoring reallocated, 159

Paging, 11, 117-168

  demand, 150-159  
   enabling and disabling, 142-144  
   using in Virtual 8086 mode,  
     307-311  
   using, with segmentation,  
     160-168

Parameters,

  8-bit and 16-bit, 112  
   variable number of, 115

Parity flag, 22

PDBR, 456

PDE, 456

Physical addresses, 39, 456  
   translating linear to, 131

Pinout, 80386, 430

Pins, miscellaneous, 436

Pointers, near and far, 15

Precision, floating point, 367

Prefetching instructions, 390

Present bit, 53, 144-145

Privilege

  protection, 10-11, 86-88  
   violations, using as debugging  
     tools, 322-323

Privileged

  code references, 94-95  
   data references, 92-93  
   instructions, 90-92

Privilege levels, 85-115, 452  
 changing, 96-111  
 changing within a task, 196  
 defining, 88-96  
 exception processing and,  
 238-239  
 Virtual 8086 mode and, 288-291

Processor and revision ID,  
 401-402

Programming  
 defensive, 112-115  
 performance issues, 385-398

Projective closure, 376

Protected mode, 8-9  
 defining segments in, 45-61  
 enabling, 419  
 exceptions, 226  
 preparing for power-up or  
 reset, 414-421  
 running 80286, 259-282

Protected mode registers, 24-25

Protected Virtual Address mode.  
*See* Protected mode

Protection  
 page-level, 144-147  
 privilege, 86-88  
 segment versus page, 123  
 Virtual 8086 mode and,  
 288-291

PTE, 456

## R

RAM, 457  
 chips, types of, 441

Read-modify-write, 446-448

Read/write bit, 147

Real Address mode. *See* Real  
 mode

Real mode, 8-9, 399, 407-414  
 returning to, 421-422

Real number formats, 17

Reduced instruction set  
 architecture, 2

Register contents, initial, 403

## Registers

control, 9, 26-28

debug, 9, 28-32, 330-333

EFLAG, 21-24

floating point data, 364

general-purpose, 20

new 80386, 25-34

new in the 80386, 6-7

often used, 484-485

Protected mode, 24-25

Real mode. 408-409

segment, 66

task, 181-184

test, 9, 32-34

using, 390-391

Register set, 18-21  
 80386, 9  
 80387, 358-365  
 general purpose versus special  
 purpose, 18

Reserved error, 256

Resetting the 80386, 399-406

Resume flag, 24, 840-341

RET instructions, matching, 1143

RISC, 457  
 based processors, 2, 18

ROM, 457

Rounding mode, 80387, 361

RPL, 457  
 field, 93

## S

Scaled indexing, 35

Scheduling, task, 186-192

Segment  
 aliasing, 74-83  
 descriptions, real-mode-like,  
 422  
 descriptors, 48-59, 478-480  
 descriptors, multi pie tasks and,  
 199-200  
 Not Present error, 253  
 overlapping, 74-83  
 protection, page versus, 123

Segmentation  
 memory, 10, 38-43, 45-84

Segmentation, *continued*  
  using as debugging tool,  
   319-322  
  using with paging together,  
   160-16S  
  Virtual 8086 mode, 286-288

Segment registers, 66  
  saving and storing, 115

Segments, 457  
  conforming code, 97-98  
  data and stack, 56  
  definition of, 45-61  
  sharing, 199-220  
  summary of types, 59  
  task state, 106-105, 173-179,  
   21S-220, 480-482

Segment selectors, 62-70  
  loading, 66-69

Selectors, 457  
  segment, 62-70

Selector sleuthing, 69-70

Self-test, 400, 449

Semaphore, 447

Separate address spaces, 201

Set/clear byte on condition  
  instruction, 38

Shadow descriptor table, 215-216

Sharing memory, 199-220

Shift versus multiply and divide,  
  397

Short real format, 17

Shutdown condition, 245-246

Signed and unsigned storage, 14

Sign extension instruction, 37

Sign flag, 22

Single-stepping debugging,  
  323-325

16-bit code and 32-bit code,  
  268-274

Sleuthing, selector, 69-70

Software  
  breakpoints, 326-325  
  exceptions, 222-223

SRAM, 457

Stack  
  errors, using as debugging  
   tools, 321-323  
  Fault error, 254  
  operations, 36  
  references, 267  
  segments, 56  
  sizing (80286 emulation),  
   266-267

Stacks  
  changing, 106-108  
  exception, 228  
  floating-point, 365  
  modifying, 21S

State frame, 171

Static RAM, 441

Status indicators, debug, 332

Status word, 80387, 362

Steppings, 80386, 457-498

Storage formats, 80387, 365-377

String instructions, 36

System bit, 55

## T

Tag word, 80387, 363

Target memory map, 166

Task  
  gates, 150, 230, 235-238  
  registers, 181-184  
  scheduling, 186-192

Tasks, 457  
  communicating among, 199-220  
  creating, 179  
  debugging separate, 351-353  
  moving between, 179-186  
  nested, 1S4-1S6  
  sharing code among, 204-206  
  sharing data among, 207-211  
   16-bit, 277  
  Virtual SOS6 mode, 285-286

Task state segments, 106-108,  
  173-179, 480-482  
  modifying, 21S-220

Task switch, break on, 329-330

Task switch operation, 182  
 Test registers, 9, 32-34  
 Timesharing, 170  
 Time slice, 170  
 TLB, 457  
 Trace flag, 22  
 Transfers  
     aligned, 433  
     unaligned, 435  
 Translation  
     address, 120-142  
     page, 127-132  
 Translation lookaside buffer,  
     422-427  
 Trap gates, 230, 233-234  
 Traps, 224  
 Trap, task switch, 329-330  
 TS bit, 192-193  
 TSS, 458  
 Type field, 55

## U

Unaligned transfers, 435  
 Unsigned values, 15  
 User/supervisor bit, 146

## V

Vector arrays, 18  
 Vectors, exception, 225-226  
 Vector table, real mode, 412

Virtual 8086 mode, 8-9, 24,  
     283-315  
     changing to Protected mode,  
         296-302  
     entering, 294-295  
     exceptions and, 295-305  
     interrupts in, 295-305  
     memory addressing options in,  
         293  
     memory segmentation and,  
         286-288  
     paging in, 307-311  
     privilege protection and,  
         288-291  
     tasks in, 285-286  
 Virtual memory management,  
     150-159

## W

Wait states, memory, 443-446  
 WC field, 114  
 Words, 15, 458

## X

XOR  
     versus BTC, 395  
     versus MOV, 393

## Z

Zero, 375  
     extension, 38  
     flag, 22