Contents

Foreword xxxi
Introduction xxxiii

Part I 1

Chapter 1  The Best Optimizer Is between Your Ears 3
The Human Element of Code Optimization 5
Understanding High Performance 6
   When Fast Isn't Fast 6
Rules for Building High-Performance Code 7
   Know Where You're Going 8
   Make a Big Map 8
   Make Lots of Little Maps 8
   Know the Territory 12
   Know When it Matters 13
   Always Consider the Alternatives 14
   Know How to Turn On the Juice 16
Where We've Been, What We've Seen 19
   Where We're Going 19

Chapter 2  A World Apart 21
The Unique Nature of Assembly Language Optimization 23
Instructions: The Individual versus the Collective 23
Assembly Is Fundamentally Different 25
   Transformation Inefficiencies 25
   Self-Reliance 27
Local Optimization 140
Unrolling Loops 143
  Rotating and Shifting with Tables 145
  NOT Flips Bits—Not Flags 146
  Incrementing with and without Carry 147

Chapter 8  Speeding Up C with Assembly Language 149
Jumping Languages When You Know It’ll Help 151
  Billy, Don’t Be a Compiler 152
Don’t Call Your Functions on Me, Baby 153
Stack Frames Slow So Much 153
Torn Between Two Segments 154
  Why Speeding Up Is Hard to Do 154
Taking It to the Limit 155
  A C-to-Assembly Case Study 156

Chapter 9  Hints My Readers Gave Me 167
Optimization Odds and Ends from the Field 169
  Another Look at LEA 170
  The Kennedy Portfolio 171
  Speeding Up Multiplication 173
  Optimizing Optimized Searching 174
  Short Sorts 180
  Full 32-Bit Division 181
  Sweet Spot Revisited 184
  Hard-Core Cycle Counting 185
  Hardwired Far Jumps 186
  Setting 32-Bit Registers: Time versus Space 187

Chapter 10  Patient Coding, Faster Code 189
How Working Quickly Can Bring Execution to a Crawl 191
  The Case for Delayed Gratification 192
The Brute-Force Syndrome 193
  Wasted Breakthroughs 196
Contents

Chapter 11  Pushing the 286 and 386  205
New Registers, New Instructions, New Timings,
New Complications  207
Family Matters  208
Crossing the Gulf to the 286 and the 386  208
In the Lair of the Cycle-Eaters, Part I  209
System Wait States  210
Data Alignment  213
Code Alignment  215
Alignment and the 386  218
Alignment and the Stack  218
The DRAM Refresh Cycle-Eater: Still an Act of God  219
The Display Adapter Cycle-Eater  219
New Instructions and Features: The 286  221
New Instructions and Features: The 386  222
Optimization Rules: The More Things Change...  223
Detailed Optimization  223
popf and the 286  225

Chapter 12  Pushing the 486  233
It's Not Just a Bigger 386  235
Enter the 486  236
Rules to Optimize By  236
The Hazards of Indexed Addressing  237
Calculate Memory Pointers Ahead of Time  238
Caveat Programmor  241
Stack Addressing and Address Pipelining  241
Problems with Byte Registers  242
More Fun with Byte Registers  244
Timing Your Own 486 Code  245
The Story Continues  246

Chapter 13  Aiming the 486  247
Pipelines and Other Hazards of the High End  249
Watch Out for Luggable Assumptions! 306
The Astonishment of Right-Brain Optimization 307
Levels of Optimization 312
  Optimization Level 1: Good Code 312
Level 2: A New Perspective 315
  Level 3: Breakthrough 316
  Enough Word Counting Already! 319

Chapter 17  The Game of Life 321
The Triumph of Algorithmic Optimization in a Cellular Automata Game 323
Conway’s Game 324
  The Rules of the Game 324
Where Does the Time Go? 329
The Hazards and Advantages of Abstraction 330
Heavy-Duty C++ Optimization 336
Bringing In the Right Brain 338
  Re-Examining the Task 338
  Acting on What We Know 340
  The Challenge That Ate My Life 346

Chapter 18  It’s a Wonderful Life 347
Optimization beyond the Pale 349
Breaking the Rules 350
Table-Driven Magic 351
Keeping Track of Change with a Change List 363
  A Layperson’s Overview of QLIFE 366

Chapter 19  Pentium: Not the Same Old Song 369
Learning a Whole Different Set of Optimization Rules 371
The Return of Optimization as Art 372
The Pentium: An Overview 373
   Crossing Cache Lines 373
   Cache Organization 374
Faster Addressing and More 375
Branch Prediction 377
Miscellaneous Pentium Topics 378
   486 versus Pentium Optimization 378
   Going Superscalar 379

Chapter 20 Pentium Rules 381
   How Your Carbon-Based Optimizer Can
   Put the “Super” in Superscalar 383
An Instruction in Every Pipe 384
V-Pipe-Capable Instructions 386
Lockstep Execution 390
Superscalar Notes 394
   Register Starvation 395

Chapter 21 Unleashing the Pentium’s V-pipe 397
   Focusing on Keeping Both
   Pentium Pipes Full 399
Address Generation Interlocks 400
Register Contention 403
   Exceptions to Register Contention 404
Who’s in First? 405
Pentium Optimization in Action 406
   A Quick Note on the 386 and 486 411

Chapter 22 Zenning and the Flexible Mind 413
   Taking a Spin through What
   You’ve Learned 415
Zenning 415
Part II  421

Chapter 23  Bones and Sinew  423
   At the Very Heart of Standard PC
      Graphics  425
   The VGA  426
   An Introduction to VGA Programming  427
   At the Core  427
      Linear Planes and True VGA Modes  430
      Smooth Panning  441
      Color Plane Manipulation  443
      Page Flipping  444
   The Hazards of VGA Clones  446
   Just the Beginning  447
   The Macro Assembler  447

Chapter 24  Parallel Processing
   with the VGA  449
   Taking on Graphics Memory Four Bytes
      at a Time  451
   VGA Programming: ALUs and Latches  451
   Notes on the ALU/Latch Demo Program  458

Chapter 25  VGA Data Machinery  461
   The Barrel Shifter, Bit Mask, and
      Set/Reset Mechanisms  463
   VGA Data Rotation  463
   The Bit Mask  464
   The VGA’s Set/Reset Circuitry  471
      Setting All Planes to a Single Color  473
      Manipulating Planes Individually  476
   Notes on Set/Reset  478
   A Brief Note on Word OUTs  479
Chapter 26  VGA Write Mode 3  481
The Write Mode That Grows on You  483
A Mode Born in Strangeness  483
A Note on Preserving Register Bits  496

Chapter 27  Yet Another VGA Write Mode  499
Write Mode 2, Chunky Bitmaps, and Text-Graphics Coexistence  501
Write Mode 2 and Set/Reset  501
A Byte’s Progress in Write Mode 2  502
Copying Chunky Bitmaps to VGA Memory Using Write Mode 2  504
Drawing Color-Patterned Lines Using Write Mode 2  509
When to Use Write Mode 2 and When to Use Set/Reset  515
Mode 13H—320×200 with 256 Colors  515
Flipping Pages from Text to Graphics and Back  515

Chapter 28  Reading VGA Memory  523
Read Modes 0 and 1, and the Color Don’t Care Register  525
Read Mode 0  525
Read Mode 1  531
When all Planes “Don’t Care”  534

Chapter 29  Saving Screens and Other VGA Mysteries  539
Useful Nuggets from the VGA Zen File  541
Saving and Restoring EGA and VGA Screens  541
16 Colors out of 64  548
Overscan  555
Chapter 30 Video Est Omnis Divisa 561
The Joys and Galling Problems of Using Split Screens on the EGA and VGA 563
How the Split Screen Works 563
The Split Screen in Action 565
VGA and EGA Split-Screen Operation Don't Mix 572
Setting the Split-Screen-Related Registers 573
The Problem with the EGA
Split Screen 573
Split Screen and Panning 574
The Split Screen and Horizontal Panning: An Example 575
Notes on Setting and Reading Registers 582
Split Screens in Other Modes 584
How Safe? 585

Chapter 31 Higher 256-Color Resolution on the VGA 587
When Is 320×200 Really 320×400? 589
Why 320×200? Only IBM Knows for Sure 590
320×400 256-Color Mode 590
Display Memory Organization in 320×400 Mode 591
Reading and Writing Pixels 593
Two 256-Color Pages 600
Something to Think About 605

Chapter 32 Be It Resolved: 360×480 607
Taking 256-Color Modes About as Far as the Standard VGA Can Take Them 609
Extended 256-Color Modes: What's Not to Like? 610
360×480 256-Color Mode 611
How 360×480 256-Color Mode Works 619
Chapter 33  Yogi Bear and Eurythmics Confront VGA Colors  623

The Basics of VGA Color Generation  625

VGA Color Basics  626
  The Palette RAM  626
  The DAC  626
  Color Paging with the Color Select Register  628

256-Color Mode  629
  Setting the Palette RAM  629
  Setting the DAC  630

If You Can’t Call the BIOS, Who Ya Gonna Call?  631

An Example of Setting the DAC  632

Chapter 34  Changing Colors without Writing Pixels  637

Special Effects through Realtime Manipulation of DAC Colors  639

Color Cycling  639

The Heart of the Problem  640
  Loading the DAC via the BIOS  641
  Loading the DAC Directly  642

A Test Program for Color Cycling  643

Color Cycling Approaches that Work  649

Odds and Ends  651
  The DAC Mask  651
  Reading the DAC  651
  Cycling Down  652
Chapter 35  Bresenham Is Fast, and Fast Is Good  653
Implementing and Optimizing Bresenham’s Line-Drawing Algorithm  655
The Task at Hand  656
Bresenham’s Line-Drawing Algorithm  657
Strengths and Weaknesses  660
An Implementation in C  661
Looking at EVGALine  665
Drawing Each Line  668
Drawing Each Pixel  669
Comments on the C Implementation  670
Bresenham’s Algorithm in Assembly  671

Chapter 36  The Good, the Bad, and the Run-Sliced  679
Faster Bresenham Lines with Run-Length Slice Line Drawing  681
Run-Length Slice Fundamentals  683
Run-Length Slice Implementation  685
Run-Length Slice Details  687

Chapter 37  Dead Cats and Lightning Lines  695
Optimizing Run-Length Slice Line Drawing in a Major Way  697
Fast Run-Length Slice Line Drawing  698
How Fast Is Fast?  704
Further Optimizations  705
Chapter 42  Wu’ed in Haste; Fried, Stewed at Leisure  773
   Fast Antialiased Lines Using Wu’s Algorithm  775
   Wu Antialiasing  776
   Tracing and Intensity in One  778
   Sample Wu Antialiasing  782
      Notes on Wu Antialiasing  791

Chapter 43  Bit-Plane Animation  793
   A Simple and Extremely Fast Animation
      Method for Limited Color  795
   Bit-Planes: The Basics  796
      Stacking the Palette Registers  799
   Bit-Plane Animation in Action  801
   Limitations of Bit-Plane Animation  811
   Shearing and Page Flipping  813
   Beating the Odds in the Jaw-Dropping Contest  814

Chapter 44  Split Screens Save the Page Flipped Day  817
   640x480 Page Flipped Animation in
      64K...Almost  819
   A Plethora of Challenges  819
   A Page Flipping Animation Demonstration  820
      Write Mode  3  831
      Drawing Text  832
      Page Flipping  833
      Knowing When to Flip  835
   Enter the Split Screen  836

Chapter 45  Dog Hair and Dirty Rectangles  839
   Different Angles on Animation  841
   Plus ça Change  842
Chapter 49  Mode X 256-Color Animation  913
How to Make the VGA Really Get up and Dance  915
Masked Copying  915
  Faster Masked Copying  918
  Notes on Masked Copying  923
Animation  924
Mode X Animation in Action  924
Works Fast, Looks Great  930

Chapter 50  Adding a Dimension  931
3-D Animation Using Mode X  933
References on 3-D Drawing  934
The 3-D Drawing Pipeline  935
  Projection  937
  Translation  937
  Rotation  938
A Simple 3-D Example  939
  Notes on the 3-D Animation Example  943
An Ongoing Journey  949

Chapter 51  Sneakers in Space  951
Using Backface Removal to Eliminate Hidden Surfaces  953
One-sided Polygons: Backface Removal  954
  Backface Removal in Action  957
Incremental Transformation  964
A Note on Rounding Negative Numbers  966
Object Representation  967

Chapter 52  Fast 3-D Animation: Meet X-Sharp  969
The First Iteration of a Generalized 3-D Animation Package  971
This Chapter’s Demo Program 972
A New Animation Framework: X-Sharp 984
Three Keys to Realtime Animation
   Performance 985
   Drawbacks 986
   Where the Time Goes 987

Chapter 53 Raw Speed and More 989
   The Naked Truth About Speed in
      3-D Animation 991
   Raw Speed, Part I: Assembly Language 992
   Raw Speed, Part II: Look it Up 999
      Hidden Surfaces 1000
      Rounding 1002
   Having a Ball 1003

Chapter 54 3-D Shading 1005
   Putting Realistic Surfaces on Animated
      3-D Objects 1007
   Support for Older Processors 1007
   Shading 1023
      Ambient Shading 1023
      Diffuse Shading 1023
   Shading: Implementation Details 1027

Chapter 55 Color Modeling in
       256-Color Mode 1031
   Pondering X-Sharp’s Color Model in an
      RGB State of Mind 1033
   A Color Model 1034
   A Bonus from the BitMan 1039

Chapter 56 Pooh and the Space
       Station 1045
   Using Fast Texture Mapping to Place Pooh
      on a Polygon 1047
Chapter 57  10,000 Freshly Sheared Sheep on the Screen  1061
The Critical Role of Experience in Implementing Fast, Smooth Texture Mapping  1063
Visual Quality: A Black Hole ... Er, Art  1064
Fixed-point Arithmetic, Redux  1064
Texture Mapping: Orientation Independence  1065
Mapping Textures across Multiple Polygons  1068
Fast Texture Mapping  1068

Chapter 58  Heinlein’s Crystal Ball, Spock’s Brain, and the 9-Cycle Dare  1077
Using the Whole-Brain Approach to Accelerate Texture Mapping  1079
Texture Mapping Redux  1080
Left-Brain Optimization  1081
A 90-Degree Shift in Perspective  1084
That’s Nice—But it Sure as Heck Ain’t 9 Cycles  1086
Don’t Stop Thinking about Those Cycles  1091
Texture Mapping Notes  1092

Chapter 59  The Idea of BSP Trees  1095
What BSP Trees Are and How to Walk Them  1097
BSP Trees 1098
  Visibility Determination 1099
  Limitations of BSP Trees 1100
Building a BSP Tree 1101
  Visibility Ordering 1104
Inorder Walks of BSP Trees 1107
  Know It Cold 1109
  Measure and Learn 1111
Surfing Amidst the Trees 1113
  Related Reading 1114

Chapter 60  Compiling BSP Trees 1115
Taking BSP Trees from Concept
to Reality 1117
Compiling BSP Trees 1119
  Parametric Lines 1119
  Parametric Line Clipping 1121
  The BSP Compiler 1123
Optimizing the BSP Tree 1128
BSP Optimization: an Undiscovered
  Country 1129

Chapter 61  Frames of Reference 1131
The Fundamentals of the Math behind 3-D
  Graphics 1133
  3-D Math 1134
  Foundation Definitions 1134
The Dot Product 1135
  Dot Products of Unit Vectors 1136
Cross Products and the Generation of
  Polygon Normals 1137
Using the Sign of the Dot Product 1140
Using the Dot Product for Projection 1141
  Rotation by Projection 1143
Chapter 65 3-D Clipping and Other Thoughts 1191

Determining What’s Inside Your Field of View 1193

3-D Clipping Basics 1195

Intersecting a Line Segment with a Plane 1195

Polygon Clipping 1197

Clipping to the Frustum 1200

The Lessons of Listing 65.3 1206

Advantages of Viewspace Clipping 1207

Further Reading 1208

Chapter 66 Quake’s Hidden-Surface Removal 1209

Struggling with Z-Order Solutions to the Hidden Surface Problem 1211

Creative Flux and Hidden Surfaces 1212
Chapter 67 Sorted Spans in Action 1223
Implementing Independent Span Sorting for Rendering without Overdraw 1225
Quake and Sorted Spans 1226
Types of 1/z Span Sorting 1228
Intersecting Span Sorting 1228
Abutting Span Sorting 1229
Independent Span Sorting 1230
1/z Span Sorting in Action 1230
Implementation Notes 1239

Chapter 68 Quake’s Lighting Model 1243
A Radically Different Approach to Lighting Polygons 1245
Problems with Gouraud Shading 1247
Perspective Correctness 1248
Decoupling Lighting from Rasterization 1250
Size and Speed 1251
Mipmapping To The Rescue 1254
Two Final Notes on Surface Caching 1255

Chapter 69 Surface Caching and Quake’s Triangle Models 1257
Letting the Graphics Card Build the Textures 1261
The Light Map as Alpha Texture 1262
Drawing Triangle Models Fast 1263
Trading Subpixel Precision for Speed 1265
An Idea that Didn’t Work 1265