HLA Reference Manual

1 HLA Overview 1
1.1.1 What is a "High Level Assembler"? 1
1.1.2 What is an "Assembler" 4
1.1.3 Is HLA a True Assembly Language? 4
1.1.4 HLA Design Goals 5
1.1.5 How to Learn Assembly Programming Using HLA 7
1.1.6 Legal Notice 7
1.1.7 Teaching Assembly Language using HLA 8
2 The Quick Guide to HLA 25
2.2.1 Overview 25
2.2.2 Running HLA 25
2.2.3 HLA Language Elements 26
2.2.3.1 Comments 26
2.2.3.2 Special Symbols 26
2.2.3.3 Reserved Words 27
2.2.3.4 External Symbols and Assembler Reserved Words 27
2.2.3.5 HLA Identifiers 27
2.2.3.6 External Identifiers 27
2.2.4 Data Types in HLA 27
2.2.4.1 Native (Primitive) Data Types in HLA 27
2.2.4.2 Composite Data Types 28
2.2.4.3 Array Data Types 28
2.2.4.4 Record Data Types 28
2.2.5 Literal Constants 29
2.2.5.1 Numeric Constants 29
2.2.5.1.1 Decimal Constants 29
2.2.5.1.2 Hexadecimal Constants 29
2.2.5.1.3 Binary Constants 29
2.2.5.1.4 Real (Floating Point) Constants 29
2.2.5.1.5 Boolean Constants 29
2.2.5.1.6 Character Constants 29
2.2.5.1.7 String Constants 30
2.2.5.1.8 Pointer Constants 30
2.2.5.1.9 Structured Constants 30
2.2.6 Constant Expressions in HLA 30
2.2.7 Program Structure 31
2.2.8 Procedure Declarations 31
2.2.8.1 Declarations 32
2.2.8.2 Type Section 32
2.2.8.3 Const Section 33
2.2.8.4 Static Section 33
2.2.8.4.1 The @NOSTORAGE Option 33

2.2.8.4.2 The EXTERNAL Option 33
2.2.8.5 Macros 34
2.2.9 The #Include Directive 35
2.2.10 The Conditional Compilation Statements (#if) 35
2.2.11 The 80x86 Instruction Set in HLA 36
2.2.11.1 Zero Operand Instructions (Null Operand Instructions) 36
2.2.11.2 General Arithmetic and Logical Instructions 36
2.2.11.3 The XCHG Instruction 37
2.2.11.4 The CMP Instruction 37
2.2.11.5 The Multiply Instructions 37
2.2.11.6 The Divide Instructions 38
2.2.11.7 Single Operand Arithmetic and Logical Instructions 38
2.2.11.8 Shift and Rotate Instructions 38
2.2.11.9 The Double Precision Shift Instructions 38
2.2.11.10 The Lea Instruction 39
2.2.11.11 The Sign and Zero Extension Instructions 39
2.2.11.12 The Push and Pop Instructions 39
2.2.11.13 Procedure Calls 39
2.2.11.14 The Ret Instruction 40
2.2.11.15 The Jmp Instructions 40
2.2.11.16 The Conditional Jump Instructions 40
2.2.11.17 The Conditional Set Instructions 40
2.2.11.18 The Input and Output Instructions 41
2.2.11.19 The Interrupt Instruction 41
2.2.11.20 Bound Instruction 41
2.2.11.21 The Enter Instruction 41
2.2.11.22 CMPXCHG Instruction 41
2.2.11.23 The XADD Instruction 42
2.2.11.24 BSF and BSR Instructions 42
2.2.11.25 The BSWAP Instruction 42
2.2.11.26 Bit Test Instructions 42
2.2.11.27 Floating Point Instructions 42
2.2.11.28 MMX and SSE Instructions 42
2.2.12 Memory Addressing Modes in HLA 42
2.2.13 Type Coercion in HLA 44
3 Installing HLA 45
3.3.1 Installing HLA Under Windows 45
3.3.1.1 New Easy Installation: 45
3.3.1.2 Manual Installation under Windows 45
3.3.1.2.1 What You've Just Done 46
3.3.1.2.2 Running HLA 49
3.3.1.3 Standard Configurations Under Windows 52
3.3.2 Installing HLA Under Linux, Mac OSX, or FreeBSD (*NIX) 54
3.3.2.1 Standard Configurations under Linux/FreeBSD/Mac OSX 57
3.3.3 Non-Standard Configurations under Windows and Linux 57
3.3.4 Customizing HLA 57

3.3.4.1 Changing the Location of HLA 58
3.3.4.2 Setting Auxiliary Paths 59
3.3.4.3 Setting the Default Back-End Assembler 59
4 Using HLA with the HIDE Integrated Development Environment 1
4.4.1 The HLA Integrated Development Environment (HIDE) 1
4.4.1.1 Description 1
4.4.1.2 Operation 1
4.4.1.3 First Execution 1
4.4.1.4 The Windows 1
4.4.1.4.1 Editor 2
4.4.1.4.2 Output 2
4.4.1.4.3 Tool Bar 2
4.4.1.4.4 Tab Bar 2
4.4.1.4.5 Status Bar 2
4.4.1.4.6 Panel 2
4.4.1.4.7 Project Panel 3
4.4.1.4.8 Properties 4
4.4.1.5 Compiling Simple Programs 4
4.4.1.6 Menus 4
4.4.1.6.1 Edit 4
4.4.1.6.2 View 5
4.4.1.6.3 Project 6
4.4.1.6.4 Make 6
4.4.1.6.5 Tools 7
4.4.1.6.6 Options 9
4.4.1.6.7 HIDE Settings 10
4.4.1.6.8 SetPaths 12
4.4.1.6.9 User 13
4.4.1.6.10 Help 14
4.4.1.7 HIDE Macros 14
4.4.1.8 Project Manager 14
4.4.1.9 Auto Completion 17
4.4.1.10 CommandLine Tools 18
4.4.1.10.1 kMake 18
4.4.1.11 Project File Format 18
4.4.1.12 Licences 22
4.4.1.12.1 HIDE 22
4.4.1.12.2 PellesC 23
4.4.1.12.3 HLA 23
4.4.2 The RadASM/HLA Integrated Development Environment 24
4.4.2.1 Integrated Development Environments 24
4.4.2.2 HLA Project Organization 24
4.4.2.3 Using Makefiles 25
4.4.2.4 Installing RadASM 31
4.4.2.5 Running RadASM 31
4.4.2.6 The RadASM Project Management Window 32

4.4.2.7 Compiling and Executing an Existing RadASM Project....38
4.4.2.8 Creating a New Project in RadASM....41
4.4.2.9 Working With RadASM Projects....48
4.4.2.10 Build Options with RadASM/HLA....50
4.4.2.11 Editing HLA Source Files Within RadASM....55
4.4.2.12 Managing Complex Projects with RadASM....59
4.4.2.13 Project Maintenance with Batch Files....60
4.4.2.14 Project Maintenance with Make Files....61
4.4.2.15 RadASM Menus....64
4.4.2.15.1 The RadASM File Menu....64
4.4.2.15.2 Edit Menu Items....67
4.4.2.15.3 The View Menu....67
4.4.2.15.4 Format Menu....68
4.4.2.15.5 The Project Menu....68
4.4.2.15.6 Make Menu....72
4.4.2.15.7 The Tools Menu....72
4.4.2.15.8 The Window Menu....72
4.4.2.15.9 The Option Menu....72
4.4.2.16 Customizing RadASM....74
4.4.2.16.1 The RADASM.INI Initialization File....74
4.4.2.16.2 The HLA.INI Initialization File....77
5 HLA Internal Operation....84
6 Using the HLA Command-Line Compiler....86
7 HLA v2.x Language Reference Manual....93
7.7.1 HLA Language Elements....93
7.7.2 Comments....93
7.7.3 Special Symbols....93
7.7.4 Reserved Words....93
7.7.5 External Symbols and Assembler Reserved Words....100
7.7.6 HLA Identifiers....100
7.7.7 External Identifiers....100
7.7.8 HLA Literal Constants....101
8 HLA Data Types....102
8.8.1 Data Types in HLA....102
8.8.2 Native (Primitive) Data Types in HLA....102
8.8.2.1 Enumerated Data Types....103
8.8.2.2 HLA Type Compatibility....104
8.8.3 Composite Data Types....105
8.8.4 Array Data Types....105
8.8.5 Union Data Types....105
8.8.6 Record Data Types....106
8.8.7 Pointer Types....111
8.8.8 Thunks....112
8.8.9 Class Types....114
8.8.10 Regular Expression Types....114
9 HLA Literal Constants and Constant Expressions....115

9.9.1 HLA Literal Constants 115
9.9.1.1 Numeric Constants 115
9.9.1.1.1 Decimal Constants 115
9.9.1.1.2 Hexadecimal Constants 115
9.9.1.1.3 Binary Constants 116
9.9.1.1.4 Numeric Set Constants 116
9.9.1.1.5 Real (Floating-Point) Constants 116
9.9.1.2 Boolean Constants 117
9.9.1.3 Character Constants 117
9.9.1.4 Unicode Character Constants 117
9.9.1.5 String Constants 117
9.9.1.6 Unicode String Constants 117
9.9.1.7 Character Set Constants 118
9.9.2 Structured Constants 118
9.9.2.1 Array Constants 118
9.9.2.2 Record Constants 119
9.9.2.3 Union Constants 120
9.9.2.4 Pointer Constants 123
9.9.2.5 Regular Expression Constants 123
9.9.3 Constant Expressions in HLA 124
9.9.3.1 Type Checking and Type Promotion 124
9.9.3.2 Type Coercion in HLA 125
9.9.3.3 !expr 126
9.9.3.4 - expr (unary negation operator) 127
9.9.3.5 expr1 * expr2 128
9.9.3.6 expr1 div expr2 129
9.9.3.7 expr1 mod expr2 129
9.9.3.8 expr1 / expr2 129
9.9.3.9 expr1 << expr2 130
9.9.3.10 expr1 >> expr2 130
9.9.3.11 expr1 + expr2 130
9.9.3.12 expr1 - expr2 130
9.9.3.13 Comparisons (=, ==, <>, !=, <, <=, >, and >=) 131
9.9.3.14 expr1 & expr2 131
9.9.3.15 expr1 in expr2 131
9.9.3.16 expr1 | expr2 131
9.9.3.17 expr1 ^ expr2 131
9.9.3.18 (expr) 132
9.9.3.19 [comma_separated_list_of_expressions] 132
9.9.3.20 record_type_name : [comma separated list of field expressions] 132
9.9.3.21 identifier 132
9.9.3.22 identifier1.identifier2 {...} 132
9.9.3.23 identifier [index_list] 133
10 HLA Program Structure and Organization 134
10.10.1 HLA Program Structure 134
10.10.2 The HLA Declaration Section 135

10.10.2.1 The HLA LABEL Declaration Section 135
10.10.2.2 The HLA CONST Declaration Section 142
10.10.2.3 The HLA VAL Declaration Section and the Compile-Time "?" Statement .. 146
10.10.2.4 The HLA TYPE Declaration Section 150
10.10.2.4.1 typeID 151
10.10.2.4.2 newTypeID : typeID; 152
10.10.2.4.3 newTypeID : typeID [list_of_array_bounds]; 152
10.10.2.4.4 newTypeID : procedure (<<optional_parameter_list>>); 153
10.10.2.4.5 newTypeID : record <<record_field_declarations>> endrecord; 153
10.10.2.4.6 newTypeID : union <<union_field_declarations>> endunion; 153
10.10.2.4.7 newTypeID : class <<class_field_declarations>> endclass; 153
10.10.2.4.8 newTypeID : pointer to typeID; 153
10.10.2.4.9 newTypeID : enum{ <<list_of_enumeration_identifiers>> }; 153
10.10.2.5 The HLA VAR Declaration Section 153
10.10.2.6 The HLA STATIC Declaration Section 160
10.10.2.7 The HLA STORAGE Declaration Section 164
10.10.2.8 The HLA READONLY Declaration Section 165
10.10.2.9 The HLA PROC Declaration Section 167
10.10.2.10 THE HLA NAMESPACE Declaration Section 167
11 HLA Procedure Declarations and Procedure Calls 171
11.11.1 Procedure Declarations 171
11.11.1.1 Original Style Procedure Declarations 171
11.11.1.2 "New Style" Procedure Declarations 175
11.11.2 Overloaded Procedure/Iterator/Method Declarations 177
11.11.3 The _vars_ and _parms_ Constants and the _display_ Array 182
11.11.4 External Procedure Declarations 183
11.11.5 Forward Procedure Declarations 184
11.11.6 Setting Default Procedure Options 185
11.11.7 Disabling HLA's Automatic Code Generation for Procedures 186
11.11.8 Procedure Calls and Parameters in HLA 191
11.11.9 Calling HLA Procedures 192
11.11.10 Parameter Passing in HLA, Value Parameters 193
11.11.10.1 Passing Byte-Sized Parameters by Value 194
11.11.10.2 Passing Word-Sized Parameters by Value 198
11.11.10.3 Passing Double-Word-Sized Parameters by Value 200
11.11.10.4 Passing Quad-Word-Sized Parameters by Value 200
11.11.10.5 Passing Tbyte-Sized Parameters by Value 201
11.11.10.6 Passing Lword-Sized Parameters by Value 201
11.11.10.7 Passing Large Parameters by Value 202
11.11.11 Parameter Passing in HLA, Reference, Value/Result, and Result Parameters .. 203
11.11.12 Untyped Reference Parameters 207
11.11.13 Pass by Value/Result and Pass by Result Parameters 208
11.11.14 Parameter Passing in HLA, Name and Lazy Evaluation Parameters 213
11.11.15 Hybrid Parameter Passing in HLA 215

11.11.16 Parameter Passing in HLA, Register Parameters 216
11.11.17 Instruction Composition and Parameter Passing in HLA 216
11.11.18 Lexical Scope 218
12 HLA Classes and Object-Oriented Programming 222
12.12.1 Class Data Types 222
12.12.2 Classes, Objects, and Object-Oriented Programming in HLA 222
12.12.3 The THIS and SUPER Reserved Words 223
12.12.4 Class Procedure and Method Prototypes 225
12.12.5 Inheritance 228
12.12.6 Abstract Methods 232
12.12.7 Classes versus Objects 232
12.12.8 Initializing the Virtual Method Table Pointer 233
12.12.9 Creating the Virtual Method Table 234
12.12.10 Calling Methods and Class Procedures 234
12.12.11 Non-object Calls of Class Procedures 236
12.12.12 Static Class Fields 237
12.12.13 Taking the Address of Class Procedures, Iterators, and Methods 239
12.12.14 Program Unit Initializers and Finalizers 240
13 The HLA Compile-Time Language 245
13.13.1 HLA Compile-Time Language, Macros, and Pragmas 245
13.13.2 Viewing the Output of the HLA Compile-Time Language 245
13.13.3 #linker Directive 246
13.13.4 The #Include Directive 246
13.13.5 The #IncludeOnce Directive 247
13.13.6 Macros 248
13.13.6.1 Standard Macros 248
13.13.6.2 Where You Declare a Macro Affects its Visibility 251
13.13.6.3 Multi-part (Context Free) Macro Invocations: 252
13.13.6.4 Macro Invocations and Macro Parameters: 256
13.13.6.5 Processing Macro Parameters 257
13.13.7 Built-in Functions: 259
13.13.8 Constant Type Conversion Functions 260
13.13.8.1 Bitwise Type Transfer Functions 261
13.13.8.2 General functions 261
13.13.8.3 String functions: 265
13.13.8.4 String/Pattern matching functions 266
13.13.8.5 Symbol and constant related functions and assembler control functions 272
13.13.8.6 Pseudo-Variables 277
13.13.8.7 Text emission functions 280
13.13.8.8 Miscellaneous Functions 280
13.13.9 #Text and #endtext Text Collection Directives 281
13.13.10 #String and #endstring Text Collection Directives 281
13.13.11 Regular Expression Macros and the @match/@match2 Functions 281
13.13.11.1 #regex..#endregex 283
13.13.11.2 The #return Clause 283
13.13.11.3 Regular Expression Elements 284

13.13.11.4 Kleene Star, Plus, and Numeric Range Specifications 284
13.13.11.5 Matching Characters in a Regular Expression 285
13.13.11.6 Case-insensitive Character Matching in a Regular Expression 286
13.13.11.7 Negated Character Matching 286
13.13.11.8 String Matching in Regular Expressions 286
13.13.11.9 Case-insenstive String Matching in Regular Expressions 287
13.13.11.10 Negated String Matching 287
13.13.11.11 String List Matching 288
13.13.11.12 Character Set Matching in a Regular Expression 288
13.13.11.13 Negated Character Set Matching 289
13.13.11.14 Matching Arbitrary Characters 289
13.13.11.15 Sequences (Concatenation) - The ‘,’ Operator 289
13.13.11.16 Alternation - The "|" Operator 289
13.13.11.17 Subexpressions - The "()" operator 290
13.13.11.18 Extracting Substrings - The Extraction Operator "<>:" 291
13.13.11.19 Invoking Other #regex Macros in a Regular Expression 291
13.13.11.20 Lookahead (peeking) 292
13.13.11.21 Utility Matching Functions 292
13.13.11.22 Backtracking 294
13.13.11.23 Lazy Versus Greedy Evaluation 295
13.13.11.24 The @match and @match2 Functions 296
13.13.11.25 Compiling and Precompiling Regular Expressions 297
13.13.11.26 The #match..#endmatch Block 298
13.13.11.27 Using Regular Expressions in Your Assembly Programs 299
13.13.12 The #asm..#endasm and #emit Directives 299
13.13.13 The #system Directive 300
13.13.14 The #print and #error Directives 301
13.13.15 Compile-Time File Output (#openwrite, #append, #write, #closewrite) 301
13.13.16 Compile-time File Input (#openread, @read, #closeread) 302
13.13.17 The Conditional Compilation Statements (#if) 302
13.13.18 The Compile-Time Loop Statements (#while and #for) 303
13.13.19 Compile-Time Functions (macros) 305
13.13.20 Sample Macro: A Modified IF..ELSE..ENDIF Statement 306
13.13.21 Text Processing, Lexical Analysis and the #text..#endtext Block 309
14 HLA Language Reference and User Manual 321
14.14.1 High Level Language Statements 321
14.14.2 Exception Handling in HLA:try..exception..endtry 321
14.14.3 Exception Handling in HLA:try..always..endtry 326
14.14.4 Exception Handling in HLA:raise 327
14.14.5 IF..THEN..ELSEIF..ELSE..ENDIF Statement in HLA 328
14.14.6 Boolean Expressions for High-Level Language Statements 329
14.14.7 WHILE..WELSE..ENDWHILE Statement in HLA 333
14.14.8 REPEAT..UNTIL Statement in HLA 334
14.14.9 The FOR..ENDFOR Statement in HLA 334
14.14.10 The FOREVER..ENDFOR Statement in HLA 336
14.14.11 The BREAK and BREAKIF Statements in HLA 336

14.14.12 The CONTINUE and CONTINUEIF Statements in HLA....336
14.14.13 The BEGIN..END, EXIT, and EXITIF Statements in HLA....337
14.14.14 The SWITCH/CASE/DEFAULT/ENDSWITCH Statement in HLA....339
14.14.15 The JT and JF Medium Level Instructions in HLA....341
14.14.16 Iterators and the HLA Foreach Loop....342
15 HLA Units and External Compilation....345
15.15.1 HLA Units and External Compilation....345
15.15.2 External Declarations....345
15.15.3 HLA Naming Conventions and Other Languages....347
15.15.4 HLA Calling Conventions and Other Languages....348
15.15.5 Calling Procedures Written in a Different Language....349
15.15.6 Calling HLA Procedures From Another Language....349
15.15.7 Linking in Code Written in Other Languages....349
15.15.8 Calling HLA Code From Other Languages....349
15.15.9 Exercising Complete Control with HLA....356
15.15.9.1 Overhead Present in an HLA Program....357
15.15.9.1.1 The "empty" Program....357
15.15.9.2 The empty Program, Part II....362
15.15.9.3 Overhead Associated With Exceptions....364
15.15.9.4 Overhead Associated with Procedures, Iterators, and Methods....371
15.15.9.5 Overhead Associated with Procedure Calls....379
15.15.9.6 Bloat in the HLA Standard Library....384
15.15.9.7 Taking Control with HLA Units....384
15.15.9.8 Hello World, Revisited....387
16 The HLA Memory Model and Memory Addressing Modes....390
16.16.1 The HLA Memory Model....390
16.16.2 Memory Addressing Modes in HLA....390
16.16.3 Type Coercion in HLA....394
17 HLA v2.x Language Reference Manual....397
17.17.1 The 80x86 Instruction Set in HLA....397
17.17.2 Zero Operand Instructions (Null Operand Instructions)....398
17.17.3 General Arithmetic and Logical Instructions....402
17.17.4 The XCHG Instruction....403
17.17.5 The CMP Instruction....404
17.17.6 The Multiply Instructions....404
17.17.7 The Divide Instructions....406
17.17.8 Single Operand Arithmetic and Logical Instructions....408
17.17.9 Shift and Rotate Instructions....409
17.17.10 The Double Precision Shift Instructions....409
17.17.11 The Lea Instruction....410
17.17.12 The Sign and Zero Extension Instructions....411
17.17.13 The Push and Pop Instructions....411
17.17.14 Procedure Calls....412
17.17.15 The Ret Instruction....414
17.17.16 The Jmp Instructions....414
17.17.17 The Conditional Jump Instructions....415

17.17.18 The Conditional Set Instructions 415
17.17.19 The Conditional Move Instructions 415
17.17.20 The Input and Output Instructions 416
17.17.21 The Interrupt Instruction 416
17.17.22 Bound Instruction 416
17.17.23 The Enter Instruction 417
17.17.24 CMPXCHG Instruction 417
17.17.25 CMPXCHG8B Instruction 418
17.17.26 The XADD Instruction 418
17.17.27 BSF and BSR Instructions 419
17.17.28 The BSWAP Instruction 419
17.17.29 Bit Test Instructions 419
17.17.30 Floating Point Instructions 420
17.17.31 Additional Floating-Point Instructions for Pentium Pro and Later Processors .. 423
17.17.32 MMX Instructions 423
17.17.33 SSE Instructions 425
17.17.34 OS/Priviledged Mode Instructions 429
17.17.35 Other Instructions and features 431
18 Advanced HLA Programming 433
18.18.1 Writing a DLL in HLA 433
18.18.1.1 Creating a Dynamic Link Library 433
18.18.1.2 Linking and Calling Procedures in a Dynamic Link Library 436
18.18.1.3 Going Farther 437
18.18.2 Compiling HLA 438
18.18.3 Code Generation for HLA HLL Control Structures 440
18.18.3.1 The HLA Standard Library 440
18.18.3.2 Compiling to MASM Code -- The Final Word 441
18.18.3.3 The HLA if..then..endif Statement, Part I 446
18.18.3.4 Boolean Expressions in HLA Control Structures 447
18.18.3.5 The JT/JF Pseudo-Instructions 453
18.18.3.6 The HLA if..then..elseif..else..endif Statement, Part II 453
18.18.3.7 The While Statement 457
18.18.3.8 repeat..until 459
18.18.3.9 for..endfor 459
18.18.3.10 forever..endfor 459
18.18.3.11 break, breakif 459
18.18.3.12 continue, continueif 460
18.18.3.13 begin..end, exit, exitif 460
18.18.3.14 foreach..endfor 460
18.18.3.15 try..unprotect..exception..anyexception..endtry, raise 460
18.18.4 A Modified IF..ELSE..ENDIF Statement 461
18.18.5 Object Oriented Programming in Assembly 468
18.18.5.1 Hoopla and Hyperbole 468
18.18.5.2 Some Basic Definitions 468
18.18.5.3 OOP Language Facilities 469

18.18.5.4 Classes in HLA 469
18.18.5.5 Objects 471
18.18.5.6 Inheritance 473
18.18.5.7 Overriding 473
18.18.5.8 Virtual Methods vs. Static Procedures 474
18.18.5.9 Writing Class Methods, Iterators, and Procedures 476
18.18.5.10 Object Implementation 479
18.18.5.10.1 Virtual Method Tables 482
18.18.5.10.2 Object Representation with Inheritance 484
18.18.5.11 Constructors and Object Initialization 487
18.18.5.12 Dynamic Object Allocation Within the Constructor 488
18.18.6 Compiling Resource Scripts Using HLA 491
18.18.6.1 The Motivation 491
18.18.6.2 The HLA Solution 491
18.18.6.3 The Resource..Endresource Declaration Section 492
18.18.7 Structures in Assembly Language Programs 493
18.18.7.1 What is a Record (Structure)? 493
18.18.7.2 Record Constants 494
18.18.7.3 Arrays of Records 495
18.18.7.4 Arrays and Records as Record Fields 495
18.18.7.5 Controlling Field Offsets Within a Record 496
18.18.7.6 Aligning Fields Within a Record 497
18.18.7.7 Using Records/Structures in an Assembly Language Program 499
18.18.7.8 Implementing Structures in an Assembler 500