Contents

List of Figures ix

List of Tables xi

1. System and Compiler Overview 1

IRIX 6.2 Overview 2

IRIX 6.2 Objectives 2

What is Not in the IRIX 6.2 Release 3

Compatibility and Supported ABIs 3

Known Compatibility Issues 7

Source Level System Interface 7

Graphics Environment Support 8

Inter-Process Communication 8

Device Driver Interface 9

Compiler System Components 9

Fortran 9

Fortran 32-Bit System 10

Fortran 64-Bit and N32 System 11

C 12

C 32-Bit System 12

C 64-Bit and N32 System 14

Fortran and C 64-Bit Components 15

Kuck and Associates Preprocessor (KAP) 15
Passing Options to KAP 17
KAP's Optimization Flags (-o, -r, -so) 17
The -o Flag 17
The -r Flag 18
The -so Flag 18
Opimization Levels Passed to KAP 19
Multiprocessing Support 20
MP Compatibility 20
MP Enhancements 20
MP Application Testing 22

2. Language Implementation Differences 23
Native 64-Bit Subprogram Interface for MIPS Architectures 23
Fortran Implementation Differences 29
New Fortran Features 29
Incompatibilities and Differences 29
C Implementation Differences 30
Structure and Union Layout Examples 31
Example 1: Structure Smaller Than a Word 31
Example 2: Structure With No Padding 31
Example 3: Structure With Internal Padding 32
Example 4: Structure With Internal and Tail Padding 32
Example 5: Union Allocation 33
Portability Issues 33

3. Source Code Porting 35
64-Bit Fortran Porting Guidelines 35
Examples of Fortran Portability Issues 36
Example 1: Changing Integer Variables 36
Example 2: Enlarging Tables 36
Example 3: Using **#if** Directives with Predefined Variables. 37
Example 4: Storing *%LOC* Return Values 37
Example 5: Modifying C Routines Called by Fortran 37
Example 6: Declaring Fortran Arguments as long ints 38
Example 7: Changing Argument Declarations in Fortran Subprograms 38
64-Bit C Porting Guidelines 39
Porting to the LP64 Model 39
Writing Code Portable to 64-Bit Platforms 41
Fundamental Types for C 42
Assembly Language Coding Guidelines 45
Overview and Predefined Variables 45
LP64 Model Implications for Assembly Language Code 46
Different Register Sizes 47
Using a Different Subrouting Linkage 47
Caller $gp (o32) vs. Callee Saved $gp (LP64) 48
Using More Floating Point Registers 54

4. Compilation Issues 55
Environment Variables 55
Command Line Switches 56
Fortran Switch Differences 56
C Switch Differences 57

Optimization Switches of the 64-Bit Compilers 58
General Optimizations 59
-O0 59
-O1 59
-O2
-O 59
-O3 59
Floating Point Optimizations 60
-OPT:roundoff=n 60
-OPT:IEEE_arithmetic=n 62
-OPT:div_split[=(ON|OFF)] 63
-OPT:fast_complex[=(ON|OFF)] 63
-OPT:fast_exp[=(ON|OFF)] 64
-OPT:fast_sqrt[=(ON|OFF)] 64
-OPT:fold_reassociate[=(ON|OFF)] 64
-OPT:IEEE_comparisons[=ON] 64
-OPT:recip[=(ON|OFF)] 64
-OPT:rsqrt[=(ON|OFF)] 64
-TARG:madd[=(ON|OFF)] 65
Miscellaneous Optimizations 65
-OPT:space 65
-OPT:alias=name 65
Loop Unrolling Optimizations 67
-OPT:unroll_times_max=n 67
-OPT:unroll_size=n 67
-OPT:unroll_bblimit=n 67

Global Code Motion 68
-GCM:aggressive_speculation[=(ON|OFF)] 68
-GCM:array_speculation[=(ON|OFF)] 68
-GCM:pointer_speculation[=(ON|OFF)] 69
-GCM:static_load_speculation[=(ON|OFF)] 69
-GCM:prepass[=(ON|OFF)]
-GCM:postpass[=(ON|OFF)] 69
Software Pipelining 70
-SWP:=(ON|OFF) 70
-SWP:back_substitution[=(ON|OFF)] 70
-SWP:backtracks_max=n 71
-SWP:body_ins_count_max=n 71
-SWP:fix_recurrences[=(ON|OFF)] 71
-SWP:if_conversion[=(ON|OFF)] 72
-SWP:interleave_reductions[=(ON|OFF)] 73
-SWP:trip_count_min=n 73
Target Environment Issues 74
-TENV:large_GOT[=(ON|OFF)]
-TENV:small_GOT[=(ON|OFF)] 74
-TENV:X=n 74
Data Alignment Options 75
-align8 / -align16 / -align32 / -align64 75
-TENV:align_aggregates=n 75
-TENV:misalignment=n 76
Compilation Messages 76
Important Warning Messages 78
warning(1411): destination type of cast is too small to hold all pointers: truncation possible 78
warning(1412): source type of cast is too small to hold all pointers: sign extension possible 78
Linking Your Application 79
Libraries 79

5. Runtime Issues 81
Runtime Differences 81
Reassociation of Operations by Optimizations 81
Algorithm Changes in Libraries 82
Hardware Changes 82
Extended MIPS Floating-Point Architecture 83
Performance Mode 84
Background 84
Performance Mode Definition 87
R8000 and R4400 Implementations 89
Full IEEE Compliance in Performance Mode 91
Application Bringup and Debugging 92

6. Performance Tuning for the R8000 95
Software Pipelining 95
Why Software Pipelining Delivers Better Performance 95
Looking at the Code Produced by Software Pipelining 99
Example 1: Output from Using the -S Compiler Switch 100
How to Diagnose What Went Wrong 102
KAP SWP Conflicts 102
Matrix Multiply – A Tuning Case Study 103
The Effect of Bellows Stalls 107
Example 2: Single Precision Loops 108
Example 3: Double Precision Loops 109
Use of the IVDEP Directive 112
Vector Intrinsic Functions 115
Manual vs. Automatic Invocation 116
Performance and Accuracy 116

7. Miscellaneous FAQ 117