CONTENTS

Page

FOREWORD 12

Clause

1 Introduction 19
1.1 Document structure 19
1.2 SCI overview 20
1.2.1 Scope and directions 20
1.2.2 The SCI approach 21
1.2.3 System configurations 22
1.2.4 Initial physical models 23
1.2.5 SCI node model 24
1.2.6 Architectural parameters 25
1.2.7 A common CSR architecture 25
1.2.8 Structure of the specification 26
1.3 Interconnect topologies 26
1.3.1 Bridged systems 26
1.3.2 Scalable systems 27
1.3.3 Interconnected systems 27
1.3.4 Backplane rings 27
1.3.5 Interconnected rings 28
1.3.6 Rectangular grid interconnects 29
1.3.7 Butterfly switches 30
1.3.8 Vendor-dependent switches 31
1.4 Transactions 31
1.4.1 Packet formats 32
1.4.2 Input and output queues 33
1.4.3 Request and response queues 34
1.4.4 Switch queues 36
1.4.5 Subactions 36
1.4.6 Remote transactions (through agents) 39
1.4.7 Move transactions 41
1.4.8 Broadcast moves 42
1.4.9 Broadcast passing by agents 43
1.4.10 Transaction types 44
1.4.11 Message passing 45
1.4.12 Global clocks 45
1.4.13 Allocation protocols 46
1.4.14 Queue allocation 47
1.5 Cache coherence 49
1.5.1 Interconnect constraints 49
1.5.2 Distributed directories 49
1.5.3 Standard optimizations 50
1.5.4 Future extensions 50
1.5.5 TLB purges 53

Clause Page
1.6 Reliability, availability, and support (RAS) ... 54
1.6.1 RAS overview ... 54
1.6.2 Autoconfiguration ... 54
1.6.3 Control and status registers ... 54
1.6.4 Transmission-error detection and isolation ... 55
1.6.5 Error containment ... 55
1.6.6 Hardware fault retry (ringlet-local, physical layer option) ... 56
1.6.7 Software fault recovery (end-to-end) ... 56
1.6.8 System debugging ... 57
1.6.9 Alternate routing ... 57
1.6.10 Online replacement ... 57
2 References, glossary, and notation ... 58
2.1 Normative references ... 58
2.2 Conformance levels ... 58
2.3 Terms and definitions ... 59
2.4 Bit and byte ordering ... 66
2.5 Numerical values ... 68
2.6 C code ... 68
3 Logical protocols and formats ... 68
3.1 Packet formats ... 68
3.1.1 Packet types ... 68
3.2 Send and echo packet formats ... 69
3.2.1 Request-send packet format ... 69
3.2.2 Request-echo packet format ... 72
3.2.3 Response-send packet ... 74
3.2.4 Standard status codes ... 76
3.2.5 Response-echo packet format ... 78
3.2.6 Interconnect-affected fields ... 79
3.2.7 Init packets ... 80
3.2.8 Cyclic redundancy code (CRC) ... 81
3.2.9 Parallel 16-bit CRC calculations ... 82
3.2.10 CRC stomping ... 84
3.2.11 Idle symbols ... 85
3.3 Logical packet encodings ... 86
3.3.1 Flag coding ... 86
3.4 Transaction types ... 89
3.4.1 Transaction commands ... 89
3.4.2 Lock subcommands ... 92
3.4.3 Unaligned DMA transfers ... 94
3.4.4 Aligned block-transfer hints ... 95
3.4.5 Move transactions ... 97
3.4.6 Global time synchronization ... 98
3.5 Elastic buffers ... 99
3.5.1 Elasticity models ... 99
3.5.2 Idle-symbol insertions ... 100
3.5.3 Idle-symbol deletions ... 101

Clause Page
3.6 Bandwidth allocation ... 101
3.6.1 Fair bandwidth allocation ... 102
3.6.2 Setting ringlet priority ... 104
3.6.3 Bandwidth partitioning ... 106
3.6.4 Types of transmission protocols ... 108
3.6.5 Pass-transmission protocol ... 108
3.6.6 Low-transmission protocol ... 111
3.6.7 Idle insertions ... 114
3.6.8 High-transmission protocol ... 114
3.7 Queue allocation ... 116
3.7.1 Queue reservations ... 116
3.7.2 Multiple active sends ... 118
3.7.3 Unfair reservations ... 119
3.7.4 Queue-selection protocols ... 119
3.7.5 Re-send priorities ... 119
3.8 Transaction errors ... 120
3.8.1 Requester timeouts (response-expected packets) ... 120
3.8.2 Time-of-death timeout (optional, all nodes) ... 120
3.8.3 Responder-processing errors ... 122
3.9 Transmission errors ... 123
3.9.1 Error isolation ... 123
3.9.2 Scrubber maintenance ... 125
3.9.3 Producer-detected errors ... 126
3.9.4 Consumer-detected errors ... 128
3.10 Address initialization ... 129
3.10.1 Transaction addressing ... 129
3.10.2 Reset types ... 131
3.10.3 Unique node identifiers ... 132
3.10.4 Ringlet initialization ... 133
3.10.5 Simple-subset ringlet resets ... 135
3.10.6 Ringlet resets ... 135
3.10.7 Ringlet clears (optional) ... 137
3.10.8 Inserting initialization packets ... 138
3.10.9 Address initialization ... 139
3.11 Packet encoding ... 140
3.11.1 Common encoding features (L18) ... 140
3.11.2 Parallel encoding with 18 signals (P18) ... 141
3.11.3 Serial encoding with 20-bit symbols (S20) ... 141
3.12 SCI-specific control and status registers ... 144
3.12.1 SCI transaction sets ... 144
3.12.2 SCI resets ... 145
3.12.3 SCI-dependent fields within standard CSRs ... 145
3.12.4 SCI-dependent CSRs ... 148
3.12.5 SCI-dependent ROM ... 151
3.12.6 Interrupt register formats ... 155
3.12.7 Interleaved logical addressing ... 157

Clause Page
4 Cache-coherence protocols ... 158
4.1 Introduction ... 158
4.1.1 Objectives ... 158
4.1.2 SCI transaction components ... 158
4.1.3 Physical addressing ... 159
4.1.4 Coherence directory overview ... 159
4.1.5 Memory and cache tags ... 160
4.1.6 Instruction-execution model ... 161
4.1.7 Coherence document structure ... 162
4.2 Coherence update sequences ... 163
4.2.1 List prepend ... 163
4.2.2 List-entry deletion ... 165
4.2.3 Update actions ... 167
4.2.4 Cache-line locks ... 167
4.2.5 Stable sharing lists ... 168
4.3 Minimal-set coherence protocols ... 171
4.3.1 Sharing-list updates ... 171
4.3.2 Cache fetching ... 171
4.3.3 Cache rollouts ... 173
4.3.4 Instruction-execution model ... 174
4.4 Typical-set coherence protocols ... 175
4.4.1 Sharing-list updates ... 175
4.4.2 Read-only fetch ... 175
4.4.3 Read-write fetch ... 177
4.4.4 Data modifications ... 178
4.4.5 Mid and head deletions ... 179
4.4.6 DMA reads and writes ... 181
4.4.7 Instruction-execution model ... 183
4.5 Full-set coherence protocols ... 184
4.5.1 Full-set option summary ... 184
4.5.2 CLEAN-list creation ... 184
4.5.3 Sharing-list additions ... 185
4.5.4 Cache washing ... 187
4.5.5 Cache flushing ... 189
4.5.6 Cache cleansing ... 191
4.5.7 Pairwise sharing ... 192
4.5.8 Pairwise-sharing faults ... 196
4.5.9 QOLB sharing ... 197
4.5.10 Cache-access properties ... 200
4.5.11 Instruction-execution model ... 201
4.6 C-code naming conventions ... 202
4.7 Coherent read and write transactions ... 203
4.7.1 Extended mread transactions ... 204
4.7.2 Cache cread and cwrite64 transactions ... 205
4.7.3 Smaller tag sizes ... 206

Clause Page
5 C-code structure ... 207
5.1 Node structure ... 207
5.1.1 Signals within a node ... 207
5.1.2 Packet transfers among node components ... 208
5.1.3 Transfer-cloud components ... 208
5.2 A node's linc component ... 210
5.2.1 A linc's subcomponents ... 210
5.2.2 A linc's elastic buffer ... 212
5.2.3 Other linc components ... 213
5.3 Other node components ... 213
5.3.1 A node's core component ... 213
5.3.2 A node's memory component ... 213
5.3.3 A node's exec component ... 214
5.3.4 A node's proc component ... 215
6 Physical layers ... 216
6.1 Type 1 module ... 217
6.1.1 Module characteristics ... 217
6.1.2 Module compatibility considerations ... 217
6.1.3 Module size ... 218
6.1.4 Warpage, bowing, and deflection ... 224
6.1.5 Cooling ... 225
6.1.6 Connector ... 226
6.1.7 Power and ground connection ... 227
6.1.8 Pin allocation for backplane parallel 18-signal encoding ... 229
6.1.9 Slot-identification signals ... 231
6.2 Type 18-DE-500 signals and power control ... 232
6.2.1 SCI differential signals ... 233
6.2.2 Status lines ... 233
6.2.3 Serial Bus signals ... 233
6.2.4 Signal levels and skew ... 233
6.2.5 Power-conversion control ... 236
6.3 Type 18-DE-500 module extender cable ... 238
6.4 Type 18-DE-500 cable-link ... 240
6.5 Serial interconnection ... 242
6.5.1 Serial interface Type 1-SE-1250, single-ended electrical ... 243
6.5.2 Optical interface, fiber-optic signal type 1-FO-1250 ... 249
6.5.3 Test methods ... 252

Annex A (informative) Ringlet initialization ... 254
Annex B (informative) SCI design models ... 257
B.1 Fast counters ... 257
B.2 Translation-lookaside-buffer coherence ... 257
B.3 Coherent lock models ... 261
B.4 Coherence-performance models ... 263

Bibliography ... 265

Page

Figure 1 – Physical-layer alternatives 23
Figure 2 – SCI node model 24
Figure 3 – 64-bit-fixed addressing 25
Figure 4 – Bridged systems 26
Figure 5 – Backplane rings 28
Figure 6 – Interconnected rings 29
Figure 7 – 2-D processor grids 29
Figure 8 – Butterfly ringlets 30
Figure 9 – Switch interface 31
Figure 10 – Subactions 32
Figure 11 – Send-packet format, simplified 32
Figure 12 – Responder queues 34
Figure 13 – Logical requester/responder queues 35
Figure 14 – Paired request and response queues 35
Figure 15 – Basic SCI bridge, paired request and response queues 36
Figure 16 – Local transaction components 37
Figure 17 – Local transaction components (busied by responder) 38
Figure 18 – Remote transaction components 40
Figure 19 – Remote move-transaction components 41
Figure 20 – Broadcast starts 43
Figure 21 – Broadcast resumes 43
Figure 22 – Transaction formats 44
Figure 23 – Bandwidth partitioning 46
Figure 24 – Resource bottlenecks 47
Figure 25 – Queue allocation avoids starvation 48
Figure 26 – Distributed cache tags 49
Figure 27 – Request combining 52
Figure 28 – Binary tree 52
Figure 29 – TLB purging 53
Figure 30 – Hardware fault-retry sequence 56
Figure 31 – Software fault-retry on coherent data 57
Figure 32 – Big-endian packet notation 67
Figure 33 – Big-endian register notation 67
Figure 34 – Send- and echo-packet formats 69
Figure 35 – Request-packet format 70
Figure 36 – Request-packet symbols 70
Figure 37 – Request-echo packet format 72
Figure 38 – Response-packet format 74
Figure 39 – Response-packet symbols 75
Figure 40 – Response-echo packet format 78
Figure 41 – Initialization-packet format 80
Figure 42 – Initialization-packet format example (*companyId*-based *uniqueId* value) 81
Figure 43 – Serialized implementation of 16-bit CRC 82
Figure 44 – Parallel CRC check 84
Figure 45 – Remote transaction components (local request-send damaged) 85
Figure 46 – Logical idle-symbol encoding 85
Figure 47 – Flag framing convention 86
Figure 48 – Logical send- and init-packet framing convention 87
Figure 49 – Logical echo-packet framing convention 87
Figure 50 – Logical sync-packet framing convention 88
Figure 51 – Logical *abort*-packet framing convention 88
Figure 52 – Selected-byte reads and writes 91
Figure 53 – Simplified lock model 92
Figure 54 – Selected-byte locks (quadlet access) 93
Figure 55 – Selected-byte locks (octlet access) 94
Figure 56 – Expected DMA read transfers 94
Figure 57 – Expected DMA write transfers 95
Figure 58 – DMA block-transfer model 96
Figure 59 – Time-sync on SCI 98
Figure 60 – Elasticity model 99

Page

Figure 61 – Input-synchronizer model 100
Figure 62 – Idle-symbol insertion 100
Figure 63 – Idle-symbol deletion 101
Figure 64 – Fair bandwidth allocation 103
Figure 65 – Increasing ringlet priority 105
Figure 66 – Restoring ringlet priority 105
Figure 67 – Idle-symbol creation, fair-only node 106
Figure 68 – Idle-symbol creation, unfair-capable node 107
Figure 69 – Idle consumption, fair-only node 107
Figure 70 – Idle consumption, unfair-capable node 108
Figure 71 – Pass-transmission model (fair-only node) 109
Figure 72 – Pass-transmission enabled 109
Figure 73 – Pass-transmission active 110
Figure 74 – Pass-transmission recovery 110
Figure 75 – Low/high-transmission model 111
Figure 76 – Low-transmission enabled 111
Figure 77 – Low-transmission active 112
Figure 78 – Low/high-transmission recovery 113
Figure 79 – Low/high-transmission debt repayment 113
Figure 80 – Low/high-transmission idle insertion 114
Figure 81 – High-transmission enabled 115
Figure 82 – Consumer send-packet queue reservations 116
Figure 83 – A/B age labels 118
Figure 84 – Response timeouts (request and no response) 120
Figure 85 – Time-of-death discards 121
Figure 85 – Packet life-cycle intervals 121
Figure 87 – Time-of-death generation model 122
Figure 88 – Responder's address-error processing 122
Figure 89 – Response timeouts (request and no response) 123
Figure 90 – Error-logging registers 124
Figure 91 – Scrubber maintenance functions 125
Figure 92 – Detecting lost low-go bits 126
Figure 93 – Producer's address-error processing 127
Figure 94 – Producer's echo-timeout processing 127
Figure 95 – Producer fatal-error recovery (optional) 128
Figure 96 – Consumer error recovery 129
Figure 97 – SCI (64-bit fixed) addressing 129
Figure 98 – Forms of node resets 132
Figure 99 – Receiver synchronization and scrubber selection 134
Figure 100 – Reset-closure generates idle symbols 134
Figure 100 – Idle-closure injects go-bits in idles 134
Figure 101 – Initialization states 136
Figure 103 – Initialization states (clear option) 137
Figure 104 – Output symbol sequence during initialization 138
Figure 105 – Insert-multiplexer model 139
Figure 106 – NodeIds after ringlet initialization and monarch selection 139
Figure 107 – NodeIds after emperor selection, final address assignments 140
Figure 108 – Flag framing convention 141
Figure 109 – S20 symbol encoding 142
Figure 110 – S20 symbol decoding 143
Figure 111 – S20 sync-packet encoding 143
Figure 112 – NODE_IDS register 145
Figure 113 – STATE_CLEAR fields 146
Figure 114 – SPLIT_TIMEOUT register-pair format 147
Figure 115 – ARGUMENT register-pair format 147
Figure 115 – CLOCK_STROBE_THROUGH format (offset 112) 148
Figure 117 – ERROR_COUNT register (offset 384) 149
Figure 118 – SYNC_INTERVAL register (offset 512) 149
Figure 119 – SAVE_ID register (offset 520) 150
Figure 120 – SLOT_ID register (offset 524) 150

Figure 121 – SCI ROM format (bus_info_block) 151
Figure 122 – ROM format, CsrOptions 152
Figure 123 – ROM format, LincOptions 153
Figure 124 – ROM format, MemoryOptions 154
Figure 125 – ROM format, CacheOptions 155
Figure 126 – DIRECTED_TARGET format 156
Figure 127 – Logical-to-physical address translation 157
Figure 128 – SCI transaction components 159
Figure 129 – Distributed sharing-list directory 160
Figure 130 – SCI coherence tags (64-byte line, 64K nodes) 161
Figure 131 – Prepend to ONLYP_DIRTY (pairwise capable) 163
Figure 132 – Memory *mread* and cache-extended *cread* components 164
Figure 133 – Deletion of head (and exclusive) entry 165
Figure 134 – Cache *cwrite64* and memory-extended *mread* components 166
Figure 135 – ONLY_DIRTY list creation (minimal set) 171
Figure 136 – GONE list additions (minimal set) 172
Figure 137 – FRESH list additions (minimal set) 172
Figure 138 – Only-entry deletions 173
Figure 139 – Tail-entry deletions 174
Figure 140 – FRESH list creation 175
Figure 141 – FRESH addition to FRESH list 176
Figure 142 – FRESH addition to DIRTY list 176
Figure 143 – DIRTY addition to FRESH list 177
Figure 144 – DIRTY addition to DIRTY list 177
Figure 145 – Head purging others 178
Figure 146 – ONLY_FRESH list conversion 179
Figure 147 – HEAD_FRESH list conversion 179
Figure 148 – Mid-entry deletions 180
Figure 149 – Head-entry deletions 180
Figure 150 – Robust ONLY_DIRTY deletions 181
Figure 151 – Checked DMA reads 181
Figure 152 – Checked DMA write (memory FRESH) 182
Figure 153 – Checked DMA write (memory GONE) 183
Figure 154 – CLEAN list creation 184
Figure 155 – FRESH addition to CLEAN/DIRTY list 185
Figure 156 – CLEAN addition to FRESH list 186
Figure 157 – CLEAN addition to CLEAN/DIRTY list 186
Figure 158 – Washing DIRTY sharing lists (prepend conflict) 188
Figure 159 – Flushing a FRESH list 190
Figure 160 – Flushing a GONE list 191
Figure 161 – Cleansing DIRTY sharing lists (prepend conflict) 192
Figure 162 – Pairwise-sharing transitions 193
Figure 163 – Prepending to pairwise list (HEAD_EXCL) 194
Figure 164 – Prepending to pairwise list (HEAD_STALE0) 195
Figure 165 – Two stale copies, head is valid 196
Figure 166 – Two stale copies, tail is valid 197
Figure 167 – Enqolb prepending to QOLB-locked list 198
Figure 168 – Deqolb tail-deletion on QOLB sharing list 199
Figure 169 – QOLB usage 199
Figure 170 – Basic mread/mwrite request 204
Figure 171 – Memory-access response 204
Figure 172 – Extended coherent memory read request 205
Figure 173 – Cache cread and cwrite64 requests 206
Figure 174 – Cache cread and cwrite64 responses 206
Figure 175 – Linc and component signals 207
Figure 176 – Linc and component queues 208
Figure 177 – One node's transfer-cloud model 209
Figure 178 – The linc packet queues 210
Figure 179 – Node interface structure 211
Figure 180 – Elasticity model 212

Page

Figure 181 – A memory component's packet queues 214
Figure 182 – An exec component's packet queues 215
Figure 183 – A proc component's packet queues 215
Figure 184 – Type 1 module and a typical subrack 217
Figure 185 – Module board 219
Figure 186 – Module injector/ejector and top and bottom shielding 220
Figure 187 – Front panel arrangement, module shielding and clearances 221
Figure 188 – Top view of subrack 222
Figure 189 – Front view of subrack, left end 223
Figure 190 – Front view of subrack, top left detail 224
Figure 191 – Module power and ESD connections 228
Figure 192 – Backplane power pinout 230
Figure 193 – Backplane signal pinout 230
Figure 194 – Slot-position backplane wiring 232
Figure 195 – ECL signal voltage limits 234
Figure 196 – Basic timing 235
Figure 197 – SCI power-distribution model 236
Figure 198 – SCI power-control signal timing 237
Figure 199 – Type 18-DE-500 module extender cable 238
Figure 200 – Arrangement of module extender cable and connector 238
Figure 201 – Arrangement of module extender power cable and connector 239
Figure 202 – Cable-link and module signal connections contrasted 240
Figure 203 – Pinout of outgoing cable-link connector 240
Figure 204 – Pinout of incoming cable-link connector 241
Figure 205 – Generic eye mask 244
Figure 206 – Line driver with transformer isolation 247
Figure 207 – Line driver with capacitive coupling 248
Figure 208 – Receiver with transformer isolation and cable equalization 248
Figure 209 – Receiver with capacitive isolation and cable equalization 249
Figure A.1 – Simple reset 255
Figure A.2 – Simple reset states 256
Figure B.1 – Simple thru-counter implementation 257
Figure B.2 – Direct-register TLB-purge interlock 259
Figure B.3 – Coherent-TLB-purge interlock 260
Figure B.4 – Enqueuing messages 263
Figure B.5 – Dequeuing messages 263

Table 1 – Packet types 68
Table 2 – Phase field for send packets 71
Table 3 – Phase field for nonbusied echoes 73
Table 4 – Phase field for busied echoes 73
Table 5 – *status.sStat* status summary codes 76
Table 6 – Serial CRC-16 implementation 82
Table 7 – Parallel implementation of 16-bit CRC 83
Table 8 – Response-expected-subaction commands (read, write, and lock) 89
Table 9 – Responseless-subaction commands (move) 90
Table 10 – Event- and response-subaction commands 90
Table 11 – Subcommand values for Lock4 and Lock8 92
Table 12 – Noncoherent block-transfer hints 97
Table 13 – Defined SCI nodeId addresses 130
Table 14 – Additional SCI transaction types 144
Table 15 – Initial nodeId values 146
Table 16 – Never-implemented CSR registers 147
Table 17 – Physical standard description 151
Table 18 – Interleave-control bits 158
Table 19 – Memory and cache update actions 167
Table 20 – Stable and semistable memory-tag states 168
Table 21 – Stable cache-tag states 169
Table 22 – Stable sharing lists 170

Page

Table 23 – MinimalExecute Routines 174
Table 24 – TypicalExecute Routines 184
Table 25 – Readable cache states 200
Table 26 – FullExecute Routines 202
Table 27 – Coherent transaction summary 203
Table 28 – Module-connector part numbers 226
Table 29 – Backplane-fixed-connector part numbers 226
Table 30 – Power-connection summary 227
Table 31 – Main characteristics of ECL signals for SCI 235
Table 32 – Cable module-like connector part number 239
Table 33 – Cable backplane-like connector part numbers 239
Table 34 – Device cable-link connector (right-angle pins) 242
Table 35 – Device cable-link connector (straight pins) 242
Table 36 – Cable cable-link connector (sockets) 242
Table 37 – Electrical signals at ETX 244
Table 38 – Electrical eye at ETX 245
Table 39 – Electrical signals at ERX 245
Table 40 – Electrical eye at ERX 245
Table 41 – Estimated maximum cable lengths 246
Table 42 – Optical eye at OTX 250
Table 43 – Optical eye at ORX 250
Table 44 – General optical requirements 250
Table 45 – Maximum laser spectral width 251
Table 46 – Typical connector properties 252
Table 47 – Loss budget 252