
Index

A

a field of instructions, **45**, 52
ABI, see *SPARC Application Binary Interface (ABI)*
ABI software, 234
ACC, see *access permissions (ACC) field of RefMMU PTE*
access permissions (ACC) field of RefMMU PTE, **248**
access type (AT) field of RefMMU FSR register, **256**
accrued exception (*aexc*) field of FSR register, **38**, 36, 74, 299, 300
activation record, see *stack frame*
ADD instruction, **108**, 86, 199, 228
ADDcc instruction, **108**, 86, 107, 228
address
 physical, 242
address space, 3, 294
address space identifier (ASI), 10, 43, 45, 48, 60, 61, 248, 250, 251, 259 thru 267, 294
address, **83**
addressing conventions, 11, 47
addressing modes, 3
ADDX instruction, **108**, 228
ADDXcc instruction, **108**, 228
aexc, see *accrued exception (aexc) field of FSR register*
aggregate data values, see *data aggregates*
alignment, 11
 data (load/store), **46**
 floating-point registers, **299**, 37
 instructions, **46**
 stack pointer, **195**
`alloca()`, 194
alternate space instructions, 14, 48, 61
ancillary state registers (ASR's), **32**, 60, 61, 131, 133, 294
AND instruction, **106**, 228
ANDcc instruction, **106**, 86, 228
ANDN instruction, **106**, 86, 228
ANDNcc instruction, **106**, 228
annul bit, 32, 52
ANSI/IEEE Standard 754-1985, 34, 35, 37, 39, 40
application program, **6**, 9, 10, 36, 37, 40, 71, 72, 74, 241
arguments to a subroutine, 189
arithmetic
 tagged, 50, 197
arithmetic overflow, 29
ASI, see *address space identifier (ASI)*
asi field of instructions, **45**, 81
asr_reg, **82**

assembler
 synthetic instructions, 85 thru 86
assembly language syntax, **81**
AT, see *access type (AT) field of RefMMU FSR register*
atomic load-store instructions, 2, 46
 load-store unsigned byte, 101, 259
 swap *r* register with memory, 102

B

BA instruction, 120, 231
BCC instruction, 231
BCLR synthetic instruction, **86**
BCS instruction, 231
BE instruction, 231
Berkeley RISCs, 1
BG instruction, 231
BGE instruction, 231
BGU instruction, 231
bibliography, **7**
Bicc instruction, 50
Bicc instructions, **119**, 231
BL instruction, 231
BLE instruction, 231
BLEU instruction, 231
block-copy operation (Ref MMU), **266**, 263
block-fill operation (Ref MMU), **267**, 263
block-zero operation, see *block-fill operation*
blocking error exception, **77**, 78, 80
BN instruction, 120, 231
BNE instruction, 231
BNEG instruction, 231
bp_CP_cc, **154**, 180
bp_CP_exception, **154**
bp_CP_present, **154**, 159, 163, 165
bp_FPU_cc, **154**
bp_FPU_exception, **154**
bp_FPU_present, **154**, 159, 160, 163, 165
bp_IRL, **153**, 158
bp_reset_in, **153**, 156, 157, 158
BPOS instruction, 231
branch on coprocessor condition codes instructions, **123**
branch on floating-point condition codes instructions, **121**
branch on integer condition codes instructions, **119**
branches
 ccc-conditional, 124

branches, *continued*
 conditional, 52
 fcc-conditional, 122
 icc-conditional, 120
 unconditional, 52, 120, 122, 124
 BSET synthetic instruction, **86**
 BTOG synthetic instruction, **86**
 BTST synthetic instruction, **86**
 BVC instruction, 231
 BVS instruction, 231
 byte ordering, 11, 47

C

C condition code bit, *see carry (C) bit of icc field of PSR*
 C, *see cacheable (C) bit of RefMMU PTE*
 cache, 249
 misses and floating-point performance, 302
 non-consistent instruction cache, 62
 split Instruction & Data, 12, 49
 cacheable (C) bit of RefMMU PTE, **248**, 61
 CALL instruction, **125**, 1, 12, 24, 25, 32, 50, 54, 134, 190, 191, 196, 198, 202, 227
 CALL synthetic instruction, **85**
 carry (C) bit of *icc* field of PSR, **29**
 CB0 instruction, 231
 CB01 instruction, 231
 CB012 instruction, 231
 CB013 instruction, 231
 CB02 instruction, 231
 CB023 instruction, 231
 CB03 instruction, 231
 CB1 instruction, 231
 CB12 instruction, 231
 CB123 instruction, 231
 CB13 instruction, 231
 CB2 instruction, 231
 CB23 instruction, 231
 CB3 instruction, 231
 CBA instruction, 124, 231
 CBccc instruction, 50, 57, 79, 154
 CBccc instructions, **123**, 72, 231
 CBN instruction, 124, 231
ccc, *see coprocessor condition codes (ccc)*
 ccc-conditional branches, 124
cexc, *see current exception (cexc) field of FSR register*
 CLR synthetic instruction, **86**
 CMP synthetic instruction, **85**
 comparison instruction, 50, 110
 compliant SPARC implementation, **4**
 concurrency, 9
cond field of instructions, **45**
 condition codes, **28**
 conditional branches, 52, 120, 122, 124
 conforming SPARC implementation, **4**
 context number (in MMU), 246
 context register (Ref MMU), **254**, 250, 252
 context switching, 202, 203
 context table (in MMU), 246
 context table pointer register (Ref MMU), **254**, 252

contexts (in MMU), **246**, 241
 control register (Ref MMU), **253**, 252
 control registers, 28 *thru* 41
 control-transfer instructions (CTI's), 12, 50
 convert between floating-point formats instructions, **143**, 300
 convert floating point to integer instructions, **142**, 302
 convert integer to floating point instructions, **141**
 coprocessor, **10**
 coprocessor condition codes (*ccc*), 124, 149
 coprocessor deferred-trap queue (CQ), **41**, 70
 coprocessor instructions, 4, 13, 14, 49, 61, 94, 99, 123, 149, 238, 292
 coprocessor operate (CPop) instructions, **149**, 5, 13, 46, 57, 238, 292
 coprocessor state register (CSR), 41
 CP, *see coprocessor*
 cp_disabled exception, **79**, 41, 57
 cp_disabled trap, 10, 94, 124, 149
 cp_exception exception, **79**, 33, 71, 72, 99
 cp_exception trap, 149
 CPop instructions, *see coprocessor operate (CPop) instructions*, 79
 CPop1 instructions, **149**, 5, 41, 228, 238, 292
 CPop2 instructions, **149**, 5, 41, 228, 238, 292
 CQ, *see coprocessor deferred-trap queue (CQ)*

 CTI, *see control-transfer instructions (CTI's)*
 current exception (*cexc*) field of FSR register, **38**, 36, 299, 300
 current window pointer (CWP) field of PSR register, **29**, 5, 10, 14, 24, 30, 54, 69, 75, 77, 79, 117, 130, 134, 135, 154, 156, 161, 177, 181, 203
 CWP, *see current window pointer (CWP) field of PSR register*

D

data aggregate
 argument passed by value, 190
 automatic, 194
 examples of, 190
 return value from function, 194, 196
 data alignment, 195
 doubleword, **46**
 halfword, **46**
 word, **46**
 data formats, 17
 data types, 17
 data_access_error exception, **79**, 294
 data_access_exception exception, **80**, 293
 data_access_MMU_miss exception, **80**
 data_store_error exception, **78**, 294
 DCTI, *see delayed control-transfer instructions (CTI's)*
 DEC synthetic instruction, **86**
 DECccc synthetic instruction, **86**
 deferred traps, **70**, 14, 43, 73
 deferred-trap queue, 14, 70
 coprocessor (CQ), **41**, 70
 floating-point (FQ), **40**, 37, 70, 97, 293
 integer unit, **33**
 Dekker's algorithm, 271, 274
 delay instruction, **51**, 12, 32, 52, 240
 delay slot, *see delay instruction*

delayed control-transfer instructions (DCTI's), 12, 51, 55, 240
 diagnostic registers (Ref MMU), **255**
disp22 field of instructions, **45**
disp30 field of instructions, **45**
 divide instructions, **115**, 12, 134, 235, 236, 292
 divide-by-zero mask (DZM) bit of TEM field of FSR register, **39**
 division-by-zero accrued (*dza*) bit of *aexc* field of FSR register, **40**
 division-by-zero current (*dzc*) bit of *cexc* field of FSR register, **40**
 division_by_zero exception, **80**, 294
 doublewords in *r* registers, **25**
 dynamic linking, 239
dza, see *division-by-zero accrued (dza) bit of aexc field of FSR register*
dzc, see *division-by-zero current (dzc) bit of cexc field of FSR register*
 DZM, see *division-by-zero mask (DZM) bit of TEM field of FSR register*

E

E, see *enable (E) bit of RefMMU CR*
 EBE, see *external bus error (EBE) field of RefMMU FSR register*
 EC field of PSR register, 94, 149
 enable (E) bit of RefMMU CR, **254**
 enable coprocessor (EC) field of PSR register, 29, 41, 57, 79, 99
 enable floating-point (EF) field of PSR register, 29, 56, 79, 92, 98, 122, 140
 enable traps (ET) field of PSR register, **29**, 70, 73, 75, 77, 80, 127, 134
 entry type (ET) field of RefMMU PTD, **247**
 entry type (ET) field of RefMMU PTE, **248**
 error_mode, 74, 75, 77, 127, 154, 156, 294
 ET, see *enable trap (ET) field of PSR register*
 ET field of PSR register, 134, 135
 ET, see *entry type (ET) field of RefMMU PTD*
 ET, see *entry type (ET) field of RefMMU PTE*
 exception, **14**
 machine-check, see *exception, non-resumable machine-check*
 exceptions, also see *trap types*, 74
 blocking error, **77**, 78, 80
 cp_disabled, **79**, 41, 57
 cp_exception, **79**, 33, 71, 72, 99, 293
 data_access_error, **79**, 294
 data_access_exception, **80**, 293
 data_access_MMU_miss, **80**
 data_store_error, **78**, 294
 division_by_zero, **80**, 294
 fp_disabled, **79**, 56, 92, 98, 141, 142, 143, 145, 146, 147, 148
 fp_exception, **79**, 33, 34, 38, 71, 72, 92, 97, 140, 141, 142, 143, 145, 146, 147, 148, 292, 293, 299
 illegal_instruction, **78**, 25, 87, 91, 95, 96, 101, 102, 127, 131, 134, 137, 139, 292
 instruction_access_error, **78**, 294
 instruction_access_exception, **78**
 instruction_access_MMU_miss, **78**
 interrupt_level_n, **80**
 mem_address_not_aligned, **79**, 46, 91, 92, 93, 94, 96, 98, 99, 100, 102, 103, 126, 127, 128
 non-resumable machine-check, **69**, 15, 71, 72, 94, 96, 98, 101, 102, 153, 164, 167, 169, 171
 peremptory error, **77**, 78, 79

exceptions, *continued*
 privileged_instruction, **78**, 127
 r_register_access_error, **78**, 294
 tag_overflow, **80**, 50, 109, 111, 197
 trap_instruction, **80**
 unimplemented_FLUSH, **78**, 139, 294
 watchpoint_detected, **79**
 window_overflow, **79**, 189
 window_underflow, **79**, 127
 execute_mode, 156
 external bus error (EBE) field of RefMMU FSR register, **256**
 external reset request, 43, 74, 75, 77, 294

F

f registers, **33**, 74, 293, 299
 FABSS instruction, **144**, 230
 FADDd instruction, **146**, 230
 FADDq instruction, **146**, 230, 237
 FADDs instruction, **146**, 230
 FAR, see *fault address register (Ref MMU)*
 fault address register (Ref MMU), **258**, 246, 252, 256
 fault address valid (FAV) bit of RefMMU FSR register, **258**
 fault status register (Ref MMU), **256**, 246, 252, 258
 fault type (FT) field of RefMMU FSR register, **257**, 254
 FAV, see *fault address valid (FAV) bit of RefMMU FSR register*
 FBA instruction, 122, 231
 FBE instruction, 231
 FBfcc instruction, 50, 56, 79, 154
 FBfcc instructions, **121**, 72, 140, 231
 FBG instruction, 231
 FBGE instruction, 231
 FBL instruction, 231
 FBLE instruction, 231
 FBLG instruction, 231
 FBN instruction, 122, 231
 FBNE instruction, 231
 FBO instruction, 231
 FBU instruction, 231
 FBUE instruction, 231
 FBUG instruction, 231
 FBUE instruction, 231
 FBUL instruction, 231
 FBULE instruction, 231
fcc, see *floating-point condition codes (fcc) field of FSR register*
fcc field of FSR register, 92, 122, 148
 fcc-conditional branches, 122
 FCMP* instructions, 38, 140
 FCMPd instruction, **148**, 231, 300
 FCMPE* instructions, 38, 140
 FCMPEd instruction, **148**, 231, 300
 FCMPEq instruction, **148**, 231, 237, 300
 FCMPEs instruction, **148**, 231, 300
 FCMPq instruction, **148**, 231, 237, 300
 FCMPs instruction, **148**, 231, 300
 FDIVd instruction, **147**, 230
 FDIVq instruction, **147**, 230, 237
 FDIVs instruction, **147**, 230
 FdMULq instruction, **147**, 230, 237, 292
 FdTOi instruction, **142**, 230, 302

- FdTOq instruction, **143**, 230, 237, 300
 - FdTOs instruction, **143**, 230, 300
 - FIFO store buffer, 64
 - FiTOd instruction, **141**, 230
 - FiTOq instruction, **141**, 230, 237
 - FiTOs instruction, **141**, 230
 - floating-point add and subtract instructions, **146**
 - floating-point compare instructions, **148**, 38, 300
 - floating-point condition codes (*fcc*) field of FSR register, **38**, 36, 37, 74, 154, 300
 - floating-point deferred-trap queue (FQ), **40**, 37, 70, 97, 293
 - floating-point instructions
 - quad-precision, 237
 - floating-point move instructions, **144**
 - floating-point multiply and divide instructions, **147**
 - floating-point operate (FPop) instructions, **140**, 5, 13, 35, 46, 56, 72, 293
 - floating-point square root instructions, **145**, 235, 237
 - floating-point state
 - fp_exception, 293
 - fp_exception_pending, 293
 - fp_execute, 293
 - floating-point state register (FSR), **34**, 38, 299
 - floating-point trap type (*ftt*) field of FSR register, **35**, 35 *thru* 37, 38, 41, 79, 292, 293, 300
 - floating-point trap types
 - hardware_error, **37**, 36
 - IEEE_754_exception, **37**, 36, 38, 40, 299
 - invalid_fp_register, **37**, 34, 36, 92, 97, 98, 140, 141, 142, 143, 145, 146, 147, 148, 163, 166, 299
 - sequence_error, **37**, 36, 41, 92, 97, 98, 293, 299
 - unfinished_FPop, **37**, 36, 40, 299, 302
 - unimplemented_FPop, **37**, 36, 40, 292, 299
 - floating-point unit (FPU), **10**
 - flush (Ref MMU operation), **250**, 249, 263
 - context, **251**
 - entire, **251**
 - region, **251**
 - segment, **251**
 - FLUSH instruction, **138**, 12, 49, 61, 62, 64, 65, 66 *thru* 68, 78, 228, 269, 273, 276, 281, 283, 287, 292
 - flush instruction memory instruction, **138**, 61, 62, 64, 65, 66 *thru* 68, 269, 273, 276, 281, 283, 287, 292
 - flush match criteria (Ref MMU)
 - PTD, **251**
 - PTE, **251**
 - flushing register windows to memory, 191
 - FMOV's instruction, **144**, 36, 230, 237
 - FMULd instruction, **147**, 230
 - FMULq instruction, **147**, 230, 237
 - FMULs instruction, **147**, 230
 - FNEGs instruction, **144**, 230, 237
 - fp_disabled exception, **79**, 56, 92, 98, 141, 142, 143, 145, 146, 147, 148
 - fp_disabled trap, 10, 92, 122, 140
 - fp_exception exception, **79**, 33, 34, 38, 71, 72, 92, 97, 140, 141, 142, 143, 145, 146, 147, 148, 292, 293, 299
 - fp_exception floating-point state, 293
 - fp_exception_pending floating-point state, 293
 - fp_execute floating-point state, 293
 - FPop instructions, **140**, see *floating-point operate (FPop) instructions*, 35, 79, 293
 - FPop1 instructions, **140**, 5, 228
 - FPop2 instructions, **140**, 5, 148, 228
 - FPU, see *floating-point unit*
 - FPU control registers, 34 *thru* 41
 - FQ, see *floating-point deferred-trap queue (FQ)*
 - FqTO* instructions, 237
 - FqTOd instruction, **143**, 230, 237, 300
 - FqTOi instruction, **142**, 230, 237, 302
 - FqTOs instruction, **143**, 230, 237, 300
 - frame pointer register, 190
 - freg*, **82**
 - FsMULd instruction, **147**, 230, 235, 237, 292
 - FSQRT* instructions, 235, 237
 - FSQRTd instruction, **145**, 230, 292
 - FSQRTq instruction, **145**, 230, 237
 - FSQRTs instruction, **145**, 230, 292
 - FSR, see *floating-point state register (FSR)*
 - FSR, see *fault status register (Ref MMU)*
 - FsTOd instruction, **143**, 230, 300
 - FsTOi instruction, **142**, 230, 302
 - FsTOq instruction, **143**, 230, 237, 300
 - FSUBd instruction, **146**, 230
 - FSUBq instruction, **146**, 230, 237
 - FSUBs instruction, **146**, 230
 - FT, see *fault type (FT) field of RefMMU FSR register*
 - ftt*, see *floating-point trap type (ftt) field of FSR register*
 - function return value, 190
 - functions returning aggregate values, 196
- ## G
- global registers, 9, 24, 192
- ## H
- hardware_error floating-point trap type, **37**, 36
 - Harvard architecture, 12, 49
 - “hidden” parameter, 194
- ## I
- i* field of instructions, **45**
 - I/O, see *input/output (I/O)*
 - IBuf, 62
 - icc* field of PSR register, **28**, 106, 108, 109, 110, 111, 112, 113, 116, 120, 130, 134
 - icc*-conditional branches, 120
 - IEEE Standard 754-1985, see *ANSI/IEEE Standard 754-1985*
 - IEEE_754_exception floating-point trap type, **37**, 36, 38, 40, 299
 - ignored, **5**
 - illegal_instruction exception, **78**, 25, 87, 91, 95, 96, 101, 102, 127, 131, 134, 137, 139, 292
 - imm22* field of instructions, **45**
 - impl field of PSR register, **28**, 35, 289, 290
 - IMPL, see *implementation (IMPL) field of RefMMU CR*
 - implementation, **5**
 - implementation (IMPL) field of RefMMU CR, **253**
 - in* registers, 9, 24
 - INC synthetic instruction, **86**

INCCc synthetic instruction, **86**

inexact accrued (nxa) bit of *aexc* field of FSR register, **40**, 302

inexact current (nxc) bit of *cexc* field of FSR register, **40**, 301, 302

inexact mask (*NXM*) bit of TEM field of FSR register, **39**, 301

infinity, 21, 302

input/output (I/O), 2, 14, 49, 59, 60, 248

input/output (I/O) locations, **60**

instruction alignment, **46**

instruction fetch, **62**, 46, 66, 68, 79, 281

instruction fields

a, **45**

asi, **45**

cond, **45**

disp22, **45**

disp30, **45**

i, **45**

imm22, **45**

op, **44**

op2, **44**

op3, **45**

opf, **46**

rd, **45**, 6

rs1, **45**, 6

rs2, **46**, 6

simm13, **46**

instruction load, **62**, 281

instruction scheduling, **240**

instruction set architecture, **5**

instruction timing, 295

instruction_access_error exception, **78**, 294

instruction_access_exception exception, **78**

instruction_access_MMU_miss exception, **78**

instructions

ADD, **108**, 86, 199, 228

ADDcc, **108**, 86, 107, 228

ADDX, **108**, 228

ADDXcc, **108**, 228

AND, **106**, 228

ANDcc, **106**, 86, 228

ANDN, **106**, 86, 228

ANDNcc, **106**, 228

atomic load-store, 2, 46, 101, 102, 292

BA, 231

BCC, 231

BCS, 231

BE, 231

BG, 231

BGE, 231

BGU, 231

Bicc, **119**, 50, 231

BL, 231

BLE, 231

BLEU, 231

BN, 231

BNE, 231

BNEG, 231

BPOS, 231

branch on coprocessor condition codes, **123**

branch on floating-point condition codes, **121**

branch on integer condition codes, **119**

BVC, 231

BVS, 231

instructions, *continued*

CALL, **125**, 1, 12, 24, 25, 32, 50, 54, 134, 190, 191, 196, 198, 202, 227

CALL (synthetic), **85**

CB0, 231

CB01, 231

CB012, 231

CB013, 231

CB02, 231

CB023, 231

CB03, 231

CB1, 231

CB12, 231

CB123, 231

CB13, 231

CB2, 231

CB23, 231

CB3, 231

CBA, 231

CBccc, **123**, 50, 57, 72, 79, 154, 231

CBN, 231

comparison, 50, 110

control-transfer (CTI's), 12, 50

convert between floating-point formats, **143**, 300

convert floating point to integer, **142**, 302

convert integer to floating point, **141**

coprocessor, 4, 13, 94, 99, 123, 149, 238, 292

coprocessor operate (CPop), **149**, 13, 238, 292

CPop, 5, 46, 57, 79

CPop1, **149**, 5, 41, 228, 238, 292

CPop2, **149**, 5, 41, 228, 238, 292

delayed control-transfer (DCTI's), 12, 51, 55, 240

divide, **115**, 12, 134, 235, 236, 292

FABSS, **144**, 230

FADDd, **146**, 230

FADDq, **146**, 230, 237, 292

FADDs, **146**, 230

FBA, 231

FBE, 231

FBfcc, **121**, 50, 56, 72, 79, 140, 154, 231

FBG, 231

FBGE, 231

FBL, 231

FBLE, 231

FBLG, 231

FBN, 231

FBNE, 231

FBO, 231

FBU, 231

FBUE, 231

FBUG, 231

FBUGE, 231

FBUL, 231

FBULE, 231

FCMP*, 38, 140

FCMPd, **148**, 231, 300

FCMPE*, 38, 140

FCMPed, **148**, 231, 300

FCMPeq, **148**, 231, 237, 292, 300

FCMPes, **148**, 231, 300

FCMPq, **148**, 231, 237, 292, 300

FCMPs, **148**, 231, 300

FDIVd, **147**, 230

FDIVq, **147**, 230, 237, 292

instructions, *continued*

FDIVs, **147**, 230
 FdMULq, **147**, 230, 237, 292
 FdTOi, **142**, 230, 302
 FdTOq, **143**, 230, 237, 292, 300
 FdTOs, **143**, 230, 300
 FiTOd, **141**, 230
 FiTOq, **141**, 230, 237, 292
 FiTOs, **141**, 230
 floating-point add and subtract, **146**
 floating-point compare, **148**, 38, 300
 floating-point move, **144**
 floating-point multiply and divide, **147**
 floating-point operate (FPop), **140**, 13, 35, 72, 293
 floating-point square root, **145**, 235, 237
 FLUSH, **138**, 12, 49, 61, 62, 64, 65, 66 *thru* 68, 78, 228, 269, 273, 276, 281, 283, 287, 292
 flush instruction memory, **138**, 61, 62, 64, 65, 66 *thru* 68, 269, 273, 276, 281, 283, 287, 292
 FMOVs, **144**, 36, 230, 237
 FMULd, **147**, 230
 FMULq, **147**, 230, 237, 292
 FMULs, **147**, 230
 FNEGs, **144**, 230, 237
 FPop, **140**, 5, 35, 46, 56, 79, 293
 FPop1, **140**, 5, 228
 FPop2, **140**, 5, 148, 228
 FqTO*, 237
 FqTOd, **143**, 230, 237, 300
 FqTOi, **142**, 230, 237, 302
 FqTOs, **143**, 230, 237, 300
 FsMULd, **147**, 230, 235, 237, 292
 FSQRT*, 235, 237
 FSQRTd, **145**, 230
 FSQRTq, **145**, 230, 237, 292
 FSQRTs, **145**, 230
 FsTOd, **143**, 230, 300
 FsTOi, **142**, 230, 302
 FsTOq, **143**, 230, 237, 292, 300
 FSUBd, **146**, 230
 FSUBq, **146**, 230, 237, 292
 FSUBs, **146**, 230
 JMPL, **126**, 1, 12, 24, 32, 50, 54, 79, 85, 127, 190, 198, 202, 228
 jump and link, **126**, 12
 LD, **90**, 229
 LDA, **90**, 229
 LDC, **94**, 229
 LDCSR, **94**, 229
 LDD, **90**, 195, 229
 LDDA, **90**, 229
 LDDC, **94**, 229
 LDDF, **92**, 229
 LDF, **92**, 229
 LDFSR, **92**, 35, 36, 40, 229, 239, 290
 LDSB, **90**, 229
 LDSBA, **90**, 229
 LDSH, **90**, 229
 LDSHA, **90**, 229
 LDSTUB, **101**, 64, 66, 229, 259, 271
 LDSTUBA, **101**, 229, 259
 LDUB, **90**, 229
 LDUBA, **90**, 229
 LDUH, **90**, 229

instructions, *continued*

LDUHA, **90**, 229
 load, 269
 load coprocessor, **94**, 46
 load floating-point, **92**, 46
 load integer, **90**, 46
 load-store unsigned byte, **101**, 259
 logical, **106**
 MULScc, **112**, 228
 multiply, **113**, 12, 234, 236, 292
 multiply step, **112**, 134
 NOP, **105**, 52, 104, 199
 OR, **106**, 85, 86, 228
 ORcc, **106**, 85, 228
 ORN, **106**, 228
 ORNcc, **106**, 228
 RDASR, **131**, 14, 32, 40, 49, 61, 86, 136, 239
 RDPSR, **131**, 28, 86, 135, 228
 RDTBR, **131**, 86, 228
 RDWIM, **131**, 30, 86, 228
 RDY, **131**, 32, 86, 134, 228
 read state register, **131**, 13, 239
 RESTORE, **117**, 1, 24, 30, 54, 79, 134, 191, 198, 199, 203, 228
 RESTORE (synthetic), **85**
 RETT, **127**, 24, 30, 50, 75, 79, 126, 134, 228
 return from trap (RETT), *see instructions, RETT*
 SAVE, **117**, 1, 24, 30, 54, 79, 134, 190, 191, 198, 199, 203, 228
 SAVE (synthetic), **85**
 SDIV, **115**, 228, 235, 236, 292
 SDIV*, 134, 205, 221
 SDIVcc, **115**, 228, 235, 236, 292
 SETHI, **104**, 85, 105, 192
 shift, **107**, 12
 SLL, **107**, 228
 SMUL, **113**, 228, 234, 236, 292
 SMUL*, 205, 206
 SMULcc, **113**, 228, 234, 236, 292
 SRA, **107**, 228
 SRL, **107**, 228
 ST, **95**, 86, 229, 259
 STA, **95**, 229, 250, 259
 STB, **95**, 86, 229, 259
 STBA, **95**, 229, 259
 STBAR, **136**, 13, 60, 65, 66, 68, 131, 138, 228, 235, 239, 294
 STC, **99**, 229, 259
 STCSR, **99**, 229, 259
 STD, **95**, 195, 229, 259
 STDA, **95**, 229, 259
 STDC, **99**, 229, 259
 STDCQ, **99**, 229, 259
 STDF, **97**, 229, 259
 STDFQ, **97**, 37, 38, 40, 229, 259, 292, 293
 STF, **97**, 229, 259
 STFSR, **97**, 35, 40, 229, 259, 291, 293
 STH, **95**, 86, 229, 259
 STHA, **95**, 229, 259
 store, 269
 store coprocessor, **99**, 46, 259
 store floating point, **97**, 46, 259
 store integer, **95**, 2, 46, 49, 259
 SUB, **110**, 85, 86, 228
 SUBcc, **110**, 50, 85, 86, 228

instructions, *continued*

- subtract, **110**
- SUBX, **110**, 228
- SUBXcc, **110**, 228
- SWAP, **102**, 64, 66, 229, 235, 259, 271, 292
- swap *r* register with memory, **102**, 259, 292
- SWAPA, **102**, 229, 259, 292
- synthetic, 85
- TA, 231
- TADDcc, **109**, 228
- TADDccTV, **109**, 80, 228
- tagged add, **109**
- tagged subtract, **111**
- TCC, 231
- TCS, 231
- TE, 231
- TG, 231
- TGE, 231
- TGU, 231
- Ticc, **129**, 14, 50, 55, 74, 80, 228, 231
- TL, 231
- TLE, 231
- TLEU, 231
- TN, 231
- TNE, 231
- TNEG, 231
- TPOS, 231
- trap on integer condition codes, **129**
- TSUBcc, **111**, 228
- TSUBccTV, **111**, 80, 228
- TVC, 231
- TVS, 231
- UDIV, **115**, 228, 235, 236, 292
- UDIV*, 134, 205, 221
- UDIVcc, **115**, 228, 235, 236, 292
- UMUL, **113**, 228, 234, 236, 292
- UMUL*, 205, 206
- UMULcc, **113**, 228, 234, 236, 292
- UNIMP, **137**, 196
- WRASR, **133**, 14, 32, 40, 49, 61, 86, 239, 294
- write state register, **133**, 239
- WRPSR, **133**, 28, 86, 228, 289
- WRTBR, **133**, 86, 228
- WRWIM, **133**, 30, 86, 228
- WRY, **133**, 32, 86, 228, 294
- XNOR, **106**, 85, 228
- XNORcc, **106**, 228
- XOR, **106**, 86, 228
- XORcc, **106**, 228

instructions (synthetic)

- BCLR, **86**
- BSET, **86**
- BTOG, **86**
- BTST, **86**
- CALL, **85**
- CLR, **86**
- CMP, **85**
- DEC, **86**
- DECcc, **86**
- INC, **86**
- INCcc, **86**
- JMP, **85**
- MOV, **86**
- NEG, **85**

instructions (synthetic), *continued*

- NOT, **85**
- RESTORE, **85**
- RET, **85**, 199, 202
- RETL, **85**, 199
- SAVE, **85**
- SET, **85**
- TST, **85**

integer condition codes, *see* *icc field of PSR register*integer divide instructions, *see* *divide instructions*integer multiply instructions, *see* *multiply instructions*integer unit (IU), **9**integer unit deferred-trap queue, **33**interrupt request, **14**

interrupt request level (bp_IRL), 73, 74, 80

interrupt_level_n exception, **80**interrupting traps, **70**, 15, 43, 73

invalid (NV) exception, 148

invalid accrued (*nva*) bit of *aexc* field of FSR register, **39**invalid current (*nvc*) bit of *cexc* field of FSR register, **39**invalid mask (*NVM*) bit of TEM field of FSR register, **39**invalid_fp_register floating-point trap type, **37**, 34, 36, 92, 97, 98, 140, 141, 142, 143, 145, 146, 147, 148, 163, 166, 299IRL, *see* *interrupt request level (IRL) field of PSR register*ISA, *see* *instruction set architecture*IU, *see* *integer unit***J**JMP synthetic instruction, **85**JMPL instruction, **126**, 1, 12, 24, 32, 50, 54, 79, 85, 127, 190, 198, 202, 228jump and link instruction, **126**, 12**L**L, *see* *level (L) field of RefMMU FSR register*LD instruction, **90**, 229LDA instruction, **90**, 229LDC instruction, **94**, 229LDCSR instruction, **94**, 229LDD instruction, **90**, 195, 229LDDA instruction, **90**, 229LDDC instruction, **94**, 229LDDF instruction, **92**, 229LDF instruction, **92**, 229LDFSR instruction, **92**, 35, 36, 40, 229, 239, 290LDSB instruction, **90**, 229LDSBA instruction, **90**, 229LDSH instruction, **90**, 229LDSHA instruction, **90**, 229LDSTUB instruction, **101**, 66, 229, 259, 271LDSTUBA instruction, **101**, 229, 259LDUB instruction, **90**, 229LDUBA instruction, **90**, 229LDUH instruction, **90**, 229LDUHA instruction, **90**, 229leaf procedure, **198**, 54, 194leaf procedure (optimized), **198**, 199level (L) field of RefMMU FSR register, **256**load coprocessor instructions, **94**

load floating-point instructions, **92**
 load instructions, 46, 240, 269
 load integer instructions, **90**
 load/store alignment, **46**, 11
 load/store alternate instructions, 11
 load/store instructions, 11
 loads from alternate space, 14, 48, 61
local registers, 9, 24, 201
local registers 1 and 2, 25, 55, 69, 77, 130
 logical instructions, **106**

M

M, see *modified (M) bit of RefMMU PTE*
 machine-check exception, see *non-resumable machine-check exception*
 main memory, see *real memory*
 Manual
 fonts, 4
 mem_address_not_aligned, 91, 92, 93, 94, 96, 98, 99, 100, 102, 103, 126, 128
 mem_address_not_aligned exception, **79**, 46, 127
 memory
 main, see *real memory*
 real, **60**, 14
 split Instruction & Data, 12, 49
 memory management unit (MMU), 2, 59, 66, 78, 80, 203, 241
 memory model, 59 *thru* 68
 N-Port, 61
 barrier synchronization, 278
 basic definitions, 60
 code patching, 275
 Dekker's algorithm, 271, 274
 fetch and add, 277
 FLUSH, **66**
 IBuf, 62
 Idealized Processor (IP), 62
 issuing order, 61, 62, 274
 memory order, 61
 mode control, 66
 mutex (mutual exclusion) locks, 270
 notation, 281
 operations, 269
 overview, 13, 59
 partial store ordering (PSO), 2, 13, 65, 136, 253, 269
 portability and recommended programming style, 269
 process switch sequence, 273
 processors and processes, 269
 producer-consumer relationship, 272
 program order, **62**
 programming with, 269 *thru* 279
 programs, 60
 single-writer-multiple-readers lock, 13
 specification, 281 *thru* 288
 spin lock, 271
 store buffer, 61
 Strong Consistency, 2, 13, 59, 60, 136, 239, 270, 274, 294
 total store ordering (TSO), 2, 13, 64, 136, 253, 269
 write lock, 13
 memory protection, 242
 memory stack layout, 194, 234
 MMU, see *memory management unit (MMU)*

mode
 supervisor, **6**, 6, 9, 48, 77, 135
 user, 2, 6, 9, 48, 191
 modified (M) bit of RefMMU PTE, **248**, 267
 MOV synthetic instruction, **86**
 MULScc (multiply step) instruction, **112**, 134, 228
 multiply instructions, **113**, 12, 234, 236, 292
 multiply step instruction, see *MULScc instruction*
 multiply/divide register, see *Y register*
 multiprocessor synchronization instructions, 4, 101, 102
 multiprocessor system, 2, 4, 13, 59, 66, 96, 98, 99, 101, 102, 138, 139
 multithreaded software, 14, 60, 65, 66
 mutex (mutual exclusion) locks, 270

N

N condition code bit, see *negative (N) bit of icc field of PSR*
 NaN (not-a-number), 300, 302
 quiet, **21**, 38, 148, 300
 signaling, **21**, 38, 143, 148, 300
 NEG synthetic instruction, **85**
 negative (N) bit of *icc* field of PSR, **28**
 negative infinity, 21, 302
 next program counter (nPC), **32**, 6, 14, 25, 43, 55, 73, 77, 130
 NF, see *no fault (NF) field of RefMMU CR*
 no fault (NF) field of RefMMU CR, **253**
 non-consistent instruction cache, 62
 non-resumable machine-check exception, **69**, 15, 71, 72, 94, 96, 98, 101, 102, 153, 164, 167, 169, 171
 nonstandard floating-point (NS) field of FSR register, **35**, 239, 293, 302
 NOP instruction, **105**, 52, 104, 120, 122, 124, 130, 199
 NOT synthetic instruction, **85**
 nPC, see *next program counter (nPC)*
ns, see *nonstandard floating-point (NS) field of FSR register*
 NV (invalid) exception, 148
nva, see *invalid accrued (nva) bit of aexc field of FSR register*
nvc, see *invalid current (nvc) bit of cexc field of FSR register*
NVM, see *invalid mask (NVM) bit of TEM field of FSR register*
 NWINDOWS, **24**, 9, 29, 77, 117, 127, 155, 161, 177, 181, 183, 198, 294
nxa, see *inexact accrued (nxa) bit of aexc field of FSR register*
nxc, see *inexact current (nxc) bit of cexc field of FSR register*
NXM, see *inexact mask (NXM) bit of TEM field of FSR register*

O

ofa, see *overflow accrued (ofa) bit of aexc field of FSR register*
ofc, see *overflow current (ofc) bit of cexc field of FSR register*
OFM, see *overflow mask (OFM) bit of TEM field of FSR register*
op field of instructions, **44**
op2 field of instructions, **44**
op3 field of instructions, **45**
 opcodes, 226
opf field of instructions, **46**
 optimized leaf procedure, see *leaf procedure (optimized)*
 OR instruction, **106**, 85, 86, 228
 ORcc instruction, **106**, 85, 228
 ORN instruction, **106**, 228

ORNcc instruction, **106**, 228
out register number 7, 25, 54, 125
out registers, 9, 24
 overflow (V) bit of *icc* field of PSR, **29**, 50, 197
 overflow accrued (*ofa*) bit of *aexc* field of FSR register, **39**
 overflow current (*ofc*) bit of *cexc* field of FSR register, **39**
 overflow mask (*OFM*) bit of TEM field of FSR register, **39**
 overwrite (OW) bit of RefMMU FSR register, **258**, 256
 OW, see *overwrite (OW) bit of RefMMU FSR register*

P

page descriptor cache (PDC), **244**, 78, 80, 249, 262, 263, 264
 page offset, 242
 page size, 241, 242
 page table descriptor (PTD), **246**
 page table entry (PTE), **247**, 243, 248, 249, 267
 page table pointer (PTP) field of RefMMU PTD, **247**
 page tables, 242
 parameters to a subroutine, 189
 partial store ordering (PSO) field of RefMMU CR, **253**, 59
 partial store ordering (PSO) memory model, 2, 13, 59, 65, 136, 138, 239, 253, 269, 285, 294
 pb_block_ldst_byte, **154**, 169, 171
 pb_block_ldst_word, **154**, 169, 171
 pb_error, **154**, 157
 PC, see *program counter (PC)*
 PDC, see *page descriptor cache (PDC)*
 peremptory error exception, **77**, 78, 79
 physical address, 242
 physical address pass-through ASI, **267**, 60, 61, 248
 physical addresses, 241, 242
 physical page number (PPN)
 Physical Page Number (PPN), 242
 physical page number (PPN) field of RefMMU PTE, **247**
 PIL, see *processor interrupt level (PIL) field of PSR register*
 positive infinity, 21, 302
 power-up reset, 75
 PPN, see *physical page number (PPN) field of RefMMU PTE*
 precise flush, **250**
 precise traps, **69**, 14, 43, 73
 previous supervisor (PS) field of PSR register, **29**, 77, 127, 135
 privileged, **6**
 privileged_instruction exception, **78**, 127
 probe (Ref MMU operation), **251**, 249, 263, 264
 process switch sequence, 273
 Processor, **6**, 9
 processor interrupt level (PIL) field of PSR register, **29**, 70, 73, 80, 134, 135
 processor state register (PSR), **28**, 10, 131, 133, 134
 processor states
 error_mode, 74, 75, 77, 127, 156, 294
 execute_mode, 156
 reset_mode, 156
 producer-consumer relationship, 272
 program counter (PC), **32**, 14, 25, 43, 55, 73, 77, 125, 130
 program counters — PC and nPC, **32**
 PS, see *previous supervisor (PS) field of PSR register*
 PSO, see *Partial Store Ordering (PSO) memory model*

PSO, see *partial store ordering (PSO) field of RefMMU CR*
 PSR, see *processor state register (PSR)*
 PTD, see *page table descriptor (PTD)*
 PTE, see *page table entry (PTE)*
 PTP, see *page table pointer (PTP) field of RefMMU PTD*

Q

qne, see *queue not empty (qne) field of FSR register*
 quad-precision floating-point instructions, 237
 queue not empty (*qne*) bit of FSR register, **38**, 37, 97, 293, 300
 quiet NaN (not-a-number), **21**, 38, 148, 300

R

R, see *referenced (R) bit of RefMMU PTE*
r register 15, 25, 54, 125
r registers, **23**
r registers 17 and 18, 25, 55, 69, 77, 130
r_register_access_error exception, **78**, 294
rd field of instructions, **45**, 6
RD, see *rounding direction (RD) field of FSR register*
 RDASR instruction, **131**, 14, 32, 40, 49, 61, 86, 136, 239
 RDPSR instruction, **131**, 28, 86, 135, 228
 RDTBR instruction, **131**, 86, 228
 RDWIM instruction, **131**, 30, 86, 228
 RDY instruction, **131**, 32, 86, 134, 228
 read state register instructions, **131**, 13
 real memory, **60**, 14
 real-time software, 202, 203
 Reference MMU, 2, 241 *thru* 259
 Reference MMU operation
 flush, 249, 250, 263
 probe, 249, 251, 263, 264
 Reference MMU reset, **259**, 254
 referenced (R) bit of RefMMU PTE, **248**, 267
 references, **7**
reg, **81**
reg_or_imm, **83**
regaddr, **83**
 register allocation within a window, 201
 register sets, 24
 register usage, 189, 234
 register window usage models, 202
 register windows, 9, 24, 189, 202, 294
 registers
 ancillary state registers (ASR's), 32, 60, 61, 131, 133, 294
 context register (Ref MMU), **254**, 250, 252
 context table pointer register (Ref MMU), **254**, 252
 control, 28, 41
 control register (Ref MMU), **253**, 252
 coprocessor, 10, 60, 61
 coprocessor state register (CSR), 41
 diagnostic registers (Ref MMU), **255**
 f registers, **33**, 10, 74, 192, 293, 299
 fault address register (Ref MMU), **258**, 246, 252, 256
 fault status register (Ref MMU), **256**, 246, 252, 258
 floating-point, 10, 192
 floating-point state register (FSR), **34**, 38, 299
 frame pointer, 190
 global, 9, 24, 192
 in, 9, 24, 189

registers, *continued*

- input/output (I/O), **61**, 14, 60, 248
- local*, 9, 24, 191, 201
- locals* 1 and 2, 25, 55, 69, 77, 130
- out*, 9, 24, 189
- out* number 7, 25, 54, 125
- processor state register (PSR), **28**, 10, 131, 133
- r* register 15, 25, 54, 125
- r* registers, **23**
- r* registers 17 and 18, 25, 55, 69, 77, 130
- stack pointer, 189, 191
- trap base register (TBR), **31**, 6, 43, 51, 55, 69, 74, 77, 127, 130, 131, 133
- window invalid mask (WIM), **30**, 10, 24, 54, 79, 131, 133
- Y*, **32**, 112, 131, 133, 134

reserved, **6**

reset

- power-up, 75
- Reference MMU, **259**, 254
- reset trap, **77**, 75, 77, 294

reset_mode, 156

RESTORE instruction, **117**, 1, 24, 30, 54, 79, 134, 191, 198, 199, 203, 228RESTORE synthetic instruction, **85**RET synthetic instruction, **85**, 199, 202RETL synthetic instruction, **85**, 199RETT instruction, **127**, 24, 30, 50, 75, 79, 126, 134, 228

return address, 191, 198

return from trap (RETT) instruction, see *RETT instruction*rounding direction (RD) field of FSR register, **34**, 141, 142, 143, 145*rs1* field of instructions, **45**, 6*rs2* field of instructions, **46**, 6

S

S, see *supervisor (S) field of PSR register*SAVE instruction, **117**, 1, 24, 30, 54, 79, 134, 190, 191, 198, 199, 203, 228SAVE synthetic instruction, **85**SC, see *system control (SC) field of RefMMU CR*

SCDs, 2

SDIV instruction, **115**, 228, 235, 236, 292

SDIV* instructions, 134, 205, 221

SDIVcc instruction, **115**, 228, 235, 236, 292

self-modifying code, 12, 49, 139, 239

sequence_error floating-point trap type, **37**, 36, 41, 92, 97, 98, 293, 299SET synthetic instruction, **85**SETHI instruction, **104**, 85, 105, 192

shared memory, 13, 59, 60, 239, 269, 270, 271, 276

shift instructions, **107**, 12, 50

sign-extended 32-bit constant, 46

signal handler, see *trap handler*signaling NaN (not-a-number), **21**, 38, 143, 148, 300

signals

- bp_CP_cc, **154**, 180
- bp_CP_exception, **154**
- bp_CP_present, **154**, 159, 163, 165
- bp_FPU_cc, **154**
- bp_FPU_exception, **154**

signals, *continued*

- bp_FPU_present, **154**, 159, 160, 163, 165
- bp_IRL, **153**, 73, 74, 80, 158
- bp_reset_in, **153**, 156, 157, 158
- pb_block_ldst_byte, **154**, 169, 171
- pb_block_ldst_word, **154**, 169, 171
- pb_error, **154**, 157

simm13 field of instructions, **46**

single-writer-multiple-readers lock, 13

SLL instruction, **107**, 228SMUL instruction, **113**, 228, 234, 236, 292

SMUL* instructions, 205, 206

SMULcc instruction, **113**, 228, 234, 236, 292software trap, **74**, 130*software_trap#*, **83**SPARC, **1**, 3, 9

SPARC ABI compliance, 192, 193, 198, 199

SPARC Application Binary Interface (ABI), **231**, 2, 192, 193, 198, 199, 205, 206, 231 *thru* 240

SPARC Compliance Definitions, 2

spin lock, 271

split I & D caches, 12, 49

SRA instruction, **107**, 228SRL instruction, **107**, 228ST instruction, **95**, 86, 229, 259STA instruction, **95**, 229, 250, 259

stack frame, 194, 195, 234

stack pointer alignment, **195**

stack pointer register, 189, 191

STB instruction, **95**, 86, 229, 259STBA instruction, **95**, 229, 259STBAR instruction, **136**, 13, 60, 65, 66, 68, 131, 138, 228, 235, 239, 294STC instruction, **99**, 229, 259STCSR instruction, **99**, 229, 259STD instruction, **95**, 195, 229, 259STDA instruction, **95**, 229, 259STDC instruction, **99**, 229, 259STDCQ instruction, **99**, 229, 259STDF instruction, **97**, 229, 259STDFQ instruction, **97**, 37, 38, 40, 229, 259, 292, 293STF instruction, **97**, 229, 259STFSR instruction, **97**, 35, 40, 229, 259, 291, 293STH instruction, **95**, 86, 229, 259STHA instruction, **95**, 229, 259store coprocessor instructions, **99**, 259store floating-point instructions, **97**, 259

store instructions, 46, 269

store integer instructions, **95**, 2, 49, 259

stores to alternate space, 14, 48, 61

Strong Consistency memory model, 2, 13, 59, 60, 136, 239, 270, 274, 294

Strong Ordering, see *Strong Consistency memory model*SUB instruction, **110**, 85, 86, 228SUBcc instruction, **110**, 50, 85, 86, 228subtract instructions, **110**SUBX instruction, **110**, 228SUBXcc instruction, **110**, 228supervisor (S) field of PSR register, **29**, 75, 77, 91, 96, 101, 102,

126, 127, 135
 supervisor mode, **6**, 6, 9, 29, 48, 77, 135
 supervisor software, **6**, 2, 14, 15, 24, 29, 30, 31, 35, 36, 37, 38, 41, 69, 77, 189
 supervisor-mode trap handler, 69, 74, 77
 swap *r* register with memory instructions, **102**, 259, 292
 SWAP instruction, **102**, 66, 229, 235, 259, 271, 292
 SWAPA instruction, **102**, 229, 259, 292
 synthetic instructions in assembler, 85 *thru* 86
 system control (SC) field of RefMMU CR, **253**

T

TA instruction, 231
 TADDcc instruction, **109**, 228
 TADDccTV instruction, **109**, 80, 228
 tag_overflow exception, **80**, 50, 109, 111, 197
 tagged add instructions, **109**
 tagged arithmetic, 50, 197
 tagged subtract instructions, **111**
 task switching, *see context switching*
 TBA, *see trap base address (TBA) field of TBR register*
 TBR, *see trap base register (TBR)*
 TCC instruction, 231
 TCS instruction, 231
 TE instruction, 231
 TEM, *see trap enable mask (TEM) field of FSR register*
 TEM, *see trap enable mask (TEM) field of PSR register*
 TG instruction, 231
 TGE instruction, 231
 TGU instruction, 231
 threads, *see multithreaded software*
 Ticc instruction, **129**, 14, 50, 55, 74, 80, 228, 231
 TL instruction, 231
 TLB, *see page descriptor cache (PDC)*
 TLE instruction, 231
 TLEU instruction, 231
 TN instruction, 231
 TNE instruction, 231
 TNEG instruction, 231
 total store ordering (TSO) memory model, 2, 13, 59, 64, 65, 136, 239, 253, 269, 283, 294
 TPOS instruction, 231
 Translation Lookaside Buffer (TLB), *see page descriptor cache (PDC)*
 trap, **14**, 6, 25, 69 *thru* 80, 189
 trap base address (TBA) field of TBR register, **31**, 69, 74, 135
 trap base register (TBR), **31**, 6, 43, 51, 55, 69, 74, 77, 127, 130, 131, 133
 trap categories
 deferred, **70**, 14, 43, 73
 interrupting, **70**, 15, 43, 73
 precise, **69**, 14, 43, 73
 trap enable mask (TEM) field of FSR register, **35**, 39, 79
 trap enable mask (TEM) field of PSR register, 73, 74
 trap handler
 supervisor-mode, 69, 74, 77
 user-mode, 36, 71, 72, 73, 301
 trap on integer condition codes instructions, **129**

trap priorities, 75
 trap type (*tt*) field of TBR register, **31**, 51, 55, 74, 75, 127, 294
 trap types, *also see exceptions*
 trap_instruction exception, **80**
 trap_instruction trap, 130
 traps, *also see exceptions*
 cp_disabled, 10
 fp_disabled, 10
 reset, **77**, 75, 77, 294
 “software”, 130
 trap_instruction, 130
 TSO, *see Total Store Ordering (TSO) memory model*
 TST synthetic instruction, **85**
 TSUBcc instruction, **111**, 228
 TSUBccTV instruction, **111**, 80, 228
tt, *see trap type (tt) field of TBR register*
 TVC instruction, 231
 TVS instruction, 231
 type field of RefMMU flush/probe address, **250**

U

UDIV instruction, **115**, 228, 235, 236, 292
 UDIV* instructions, 134, 205, 221
 UDIVcc instruction, **115**, 228, 235, 236, 292
ufa, *see underflow accrued (ufa) bit of aexc field of FSR register*
ufc, *see underflow current (ufc) bit of cexc field of FSR register*
UFM, *see underflow mask (UFM) bit of TEM field of FSR register*
 UMUL instruction, **113**, 228, 234, 236, 292
 UMUL* instructions, 205, 206
 UMULcc instruction, **113**, 228, 234, 236, 292
 unconditional branches, 52, 120, 122, 124
 underflow accrued (*ufa*) bit of *aexc* field of FSR register, **39**, 301
 underflow current (*ufc*) bit of *cexc* field of FSR register, **39**, 301
 underflow mask (*UFM*) bit of TEM field of FSR register, **39**, 301
 unfinished_FPop floating-point trap type, **37**, 36, 40, 299, 302
 UNIMP instruction, **137**, 196
 unimplemented_FLUSH exception, **78**, 139, 294
 unimplemented_FPop floating-point trap type, **37**, 36, 40, 292, 299
 unused, **6**
 user application program, *see application program*
 user code, *see user mode*
 user mode, **6**, 2, 9, 29, 35, 38, 48, 191
 user stack frame, 195
 user-mode trap handler, 36, 71, 72, 73, 301

V

V condition code bit, *see overflow (V) bit of icc field of PSR*
varargs, *see variable-length argument list*
 variable-length argument list, 190
 IMPL, *see version (ver) field of RefMMU CR*
ver, *see version (ver) field of FSR register*
ver, *see version (ver) field of PSR register*
 version (*ver*) field of FSR register, **35**, 289, 290 *thru* 291
 version (*ver*) field of PSR register, **28**, 35, 289, 290
 version (VER) field of RefMMU CR, **253**
 virtual addresses, 241, 242, 243
 virtual flush/probe address (VFPA) field of RefMMU flush/probe address, **250**

W

watchpoint register, 79
watchpoint_detected exception, **79**
WIM, see *window invalid mask (WIM) register*
window invalid mask (WIM) register, **30**, 10, 24, 54, 79, 117, 131, 133, 134, 203
window_overflow exception, **79**, 189
window_underflow exception, **79**, 127
windows, see *register windows*
WRASR instruction, **133**, 14, 32, 40, 49, 61, 86, 239, 294
write lock, 13
write state register instructions, **133**
WRPSR instruction, **133**, 28, 86, 228, 289
WRTBR instruction, **133**, 86, 228
WRWIM instruction, **133**, 30, 86, 228
WRY instruction, **133**, 32, 86, 228, 294

X

XNOR instruction, **106**, 85, 228
XNORcc instruction, **106**, 228
XOR instruction, **106**, 86, 228
XORcc instruction, **106**, 228

Y

Y register, **32**, 112, 131, 133, 134

Z

Z condition code bit, see *zero (Z) bit of icc field of PSR*
zero (Z) bit of icc field of PSR, **29**