PM0214
Programming manual

STM32 Cortex®-M4 MCUs and MPUs programming manual

Introduction

This programming manual provides information for application and system-level software developers. It gives a full description of the STM32 Cortex®-M4 processor programming model, instruction set and core peripherals. The applicable products are listed in the table below.

The Cortex®-M4 processor used in STM32F3 Series, STM32F4 Series, STM32G4 Series, STM32H745/755 and STM32H747/757 Lines, STM32L4 Series, STM32L4+ Series, STM32WB Series, STM32WL Series and STM32MP1 Series, is a high performance 32-bit processor designed for the microcontroller and microprocessor market. It offers significant benefits to developers, including:

- Outstanding processing performance combined with fast interrupt handling
- Enhanced system debug with extensive breakpoint and trace capabilities
- Efficient processor core, system and memories
- Ultra-low power consumption with integrated sleep modes
- Platform security

Table 1. Applicable products

Type	Product Series and Lines
Microcontrollers	STM32F3 Series, STM32F4 Series, STM32G4 Series, STM32L4 Series, STM32L4+ Series, STM32WB Series, STM32WL Series STM32H745/755 and STM32H747/757 Lines
Microprocessors	STM32MP1 Series

Reference documents

Available from STMicroelectronics web site *www.st.com*:

- Datasheets of STM32F3 Series, STM32F4 Series, STM32G4 Series, STM32H745/755 and STM32H747/757 Lines, STM32L4 Series, STM32L4+ Series, STM32MP1 Series, STM32WB Series and STM32WL Series
- Reference manuals of STM32F3 Series, STM32F4 Series, STM32G4 Series, STM32H745/755 and STM32H747/757 Lines, STM32L4 Series, STM32L4+ Series, STM32MP1 Series, STM32WB Series and STM32WL Series

Contents

1 About this document 12
1.1 Typographical conventions 12
1.2 List of abbreviations for registers 12
1.3 About the STM32 Cortex-M4 processor and core peripherals 13
1.3.1 System level interface 14
1.3.2 Integrated configurable debug 14
1.3.3 Cortex-M4 processor features and benefits summary 15
1.3.4 Cortex-M4 core peripherals 16

2 The Cortex-M4 processor 17
2.1 Programmers model 17
2.1.1 Processor mode and privilege levels for software execution 17
2.1.2 Stacks 17
2.1.3 Core registers 18
2.1.4 Exceptions and interrupts 26
2.1.5 Data types 26
2.1.6 The Cortex microcontroller software interface standard (CMSIS) 26
2.2 Memory model 28
2.2.1 Memory regions, types and attributes 29
2.2.2 Memory system ordering of memory accesses 29
2.2.3 Behavior of memory accesses 30
2.2.4 Software ordering of memory accesses 31
2.2.5 Bit-banding 32
2.2.6 Memory endianness 34
2.2.7 Synchronization primitives 34
2.2.8 Programming hints for the synchronization primitives 36
2.3 Exception model 37
2.3.1 Exception states 37
2.3.2 Exception types 37
2.3.3 Exception handlers 39
2.3.4 Vector table 40
2.3.5 Exception priorities 41
2.3.6 Interrupt priority grouping 41
2.3.7 Exception entry and return 42

2.4 Fault handling . . . 44
2.4.1 Fault types . . . 45
2.4.2 Fault escalation and hard faults . . . 46
2.4.3 Fault status registers and fault address registers . . . 47
2.4.4 Lockup . . . 47
2.5 Power management . . . 47
2.5.1 Entering sleep mode . . . 48
2.5.2 Wakeup from sleep mode . . . 48
2.5.3 External event input / extended interrupt and event input . . . 49
2.5.4 Power management programming hints . . . 49

3 The STM32 Cortex-M4 instruction set . . . 50
3.1 Instruction set summary . . . 50
3.2 CMSIS intrinsic functions . . . 58
3.3 About the instruction descriptions . . . 60
3.3.1 Operands . . . 60
3.3.2 Restrictions when using PC or SP . . . 60
3.3.3 Flexible second operand . . . 60
3.3.4 Shift operations . . . 62
3.3.5 Address alignment . . . 65
3.3.6 PC-relative expressions . . . 65
3.3.7 Conditional execution . . . 65
3.3.8 Instruction width selection . . . 68
3.4 Memory access instructions . . . 69
3.4.1 ADR . . . 70
3.4.2 LDR and STR, immediate offset . . . 71
3.4.3 LDR and STR, register offset . . . 73
3.4.4 LDR and STR, unprivileged . . . 74
3.4.5 LDR, PC-relative . . . 75
3.4.6 LDM and STM . . . 76
3.4.7 PUSH and POP . . . 78
3.4.8 LDREX and STREX . . . 79
3.4.9 CLREX . . . 80
3.5 General data processing instructions . . . 81
3.5.1 ADD, ADC, SUB, SBC, and RSB . . . 83
3.5.2 AND, ORR, EOR, BIC, and ORN . . . 85

3.5.3 ASR, LSL, LSR, ROR, and RRX . . . 86
3.5.4 CLZ . . . 87
3.5.5 CMP and CMN . . . 88
3.5.6 MOV and MVN . . . 89
3.5.7 MOVT . . . 91
3.5.8 REV, REV16, REVSH, and RBIT . . . 92
3.5.9 SADD16 and SADD8 . . . 93
3.5.10 SHADD16 and SHADD8 . . . 94
3.5.11 SHASX and SHSAX . . . 95
3.5.12 SHSUB16 and SHSUB8 . . . 96
3.5.13 SSUB16 and SSUB8 . . . 97
3.5.14 SASX and SSAX . . . 98
3.5.15 TST and TEQ . . . 99
3.5.16 UADD16 and UADD8 . . . 100
3.5.17 UASX and USAX . . . 101
3.5.18 UHADD16 and UHADD8 . . . 102
3.5.19 UHASX and UHSAX . . . 103
3.5.20 UHSUB16 and UHSUB8 . . . 104
3.5.21 SEL . . . 105
3.5.22 USAD8 . . . 106
3.5.23 USADA8 . . . 107
3.5.24 USUB16 and USUB8 . . . 108
3.6 Multiply and divide instructions . . . 109
3.6.1 MUL, MLA, and MLS . . . 110
3.6.2 UMULL, UMAAL and UMLAL . . . 111
3.6.3 SMLA and SMLAW . . . 112
3.6.4 SMLAD . . . 114
3.6.5 SMLAL and SMLALD . . . 115
3.6.6 SMLSD and SMLSLD . . . 117
3.6.7 SMMLA and SMMLS . . . 119
3.6.8 SMMUL . . . 120
3.6.9 SMUAD and SMUSD . . . 121
3.6.10 SMUL and SMULW . . . 122
3.6.11 UMULL, UMLAL, SMULL, and SMLAL . . . 123
3.6.12 SDIV and UDIV . . . 124
3.7 Saturating instructions . . . 125
3.7.1 SSAT and USAT . . . 126

3.7.2 SSAT16 and USAT16 . . . 127
3.7.3 QADD and QSUB . . . 128
3.7.4 QASX and QSAX . . . 129
3.7.5 QDADD and QDSUB . . . 130
3.7.6 UQASX and UQSAX . . . 131
3.7.7 UQADD and UQSUB . . . 132
3.8 Packing and unpacking instructions . . . 134
3.8.1 PKHBT and PKHTB . . . 135
3.8.2 SXT and UXT . . . 136
3.8.3 SXTA and UXTA . . . 137
3.9 Bitfield instructions . . . 138
3.9.1 BFC and BFI . . . 139
3.9.2 SBFX and UBFX . . . 140
3.9.3 SXT and UXT . . . 141
3.9.4 Branch and control instructions . . . 142
3.9.5 B, BL, BX, and BLX . . . 142
3.9.6 CBZ and CBNZ . . . 144
3.9.7 IT . . . 145
3.9.8 TBB and TBH . . . 147
3.10 Floating-point instructions . . . 149
3.10.1 VABS . . . 151
3.10.2 VADD . . . 152
3.10.3 VCMP, VCMPE . . . 153
3.10.4 VCVT, VCVTR between floating-point and integer . . . 154
3.10.5 VCVT between floating-point and fixed-point . . . 155
3.10.6 VCVTB, VCVTT . . . 156
3.10.7 VDIV . . . 157
3.10.8 VFMA, VFMS . . . 158
3.10.9 VFNMA, VFNMS . . . 159
3.10.10 VLDM . . . 160
3.10.11 VLDR . . . 161
3.10.12 VLMA, VLMS . . . 162
3.10.13 VMOV immediate . . . 163
3.10.14 VMOV register . . . 164
3.10.15 VMOV scalar to Arm core register . . . 165
3.10.16 VMOV Arm core register to single precision . . . 166
3.10.17 VMOV two Arm core registers to two single precision . . . 167

3.10.18 VMOV Arm Core register to scalar . . . 168
3.10.19 VMRS . . . 169
3.10.20 VMSR . . . 170
3.10.21 VMUL . . . 171
3.10.22 VNEG . . . 172
3.10.23 VNMLA, VNMLS, VNMUL . . . 173
3.10.24 VPOP . . . 174
3.10.25 VPUSH . . . 175
3.10.26 VSQRT . . . 176
3.10.27 VSTM . . . 177
3.10.28 VSTR . . . 178
3.10.29 VSUB . . . 179
3.11 Miscellaneous instructions . . . 180
3.11.1 BKPT . . . 181
3.11.2 CPS . . . 182
3.11.3 DMB . . . 183
3.11.4 DSB . . . 184
3.11.5 ISB . . . 185
3.11.6 MRS . . . 186
3.11.7 MSR . . . 187
3.11.8 NOP . . . 188
3.11.9 SEV . . . 189
3.11.10 SVC . . . 190
3.11.11 WFE . . . 191
3.11.12 WFI . . . 192

4 Core peripherals . . . 193
4.1 About the STM32 Cortex-M4 core peripherals . . . 193
4.2 Memory protection unit (MPU) . . . 193
4.2.1 MPU access permission attributes . . . 195
4.2.2 MPU mismatch . . . 196
4.2.3 Updating an MPU region . . . 196
4.2.4 MPU design hints and tips . . . 199
4.2.5 MPU type register (MPU_TYPER) . . . 200
4.2.6 MPU control register (MPU_CTRL) . . . 201
4.2.7 MPU region number register (MPU_RNR) . . . 202
4.2.8 MPU region base address register (MPU_RBAR) . . . 203

4.2.9 MPU region attribute and size register (MPU_RASR) 204
4.2.10 MPU register map 206
4.3 Nested vectored interrupt controller (NVIC) 208
4.3.1 Accessing the Cortex-M4 NVIC registers using CMSIS 209
4.3.2 Interrupt set-enable register x (NVIC_ISERx) 210
4.3.3 Interrupt clear-enable register x (NVIC_ICERx) 211
4.3.4 Interrupt set-pending register x (NVIC_ISPRx) 212
4.3.5 Interrupt clear-pending register x (NVIC_ICPRx) 213
4.3.6 Interrupt active bit register x (NVIC_IABRx) 214
4.3.7 Interrupt priority register x (NVIC_IPRx) 215
4.3.8 Software trigger interrupt register (NVIC_STIR) 216
4.3.9 Level-sensitive and pulse interrupts 217
4.3.10 NVIC design hints and tips 218
4.3.11 NVIC register map 219
4.4 System control block (SCB) 221
4.4.1 Auxiliary control register (ACTLR) 222
4.4.2 CPUID base register (CPUID) 224
4.4.3 Interrupt control and state register (ICSR) 225
4.4.4 Vector table offset register (VTOR) 227
4.4.5 Application interrupt and reset control register (AIRCR) 228
4.4.6 System control register (SCR) 230
4.4.7 Configuration and control register (CCR) 231
4.4.8 System handler priority registers (SHPRx) 233
4.4.9 System handler control and state register (SHCSR) 235
4.4.10 Configurable fault status register (CFSR; UFSR+BFSR+MMFSR) . . . 237
4.4.11 Usage fault status register (UFSR) 238
4.4.12 Bus fault status register (BFSR) 239
4.4.13 Memory management fault address register (MMFSR) 240
4.4.14 Hard fault status register (HFSR) 241
4.4.15 Memory management fault address register (MMFAR) 242
4.4.16 Bus fault address register (BFAR) 242
4.4.17 Auxiliary fault status register (AFSR) 243
4.4.18 System control block design hints and tips 243
4.4.19 SCB register map 244
4.5 SysTick timer (STK) 246
4.5.1 SysTick control and status register (STK_CTRL) 247
4.5.2 SysTick reload value register (STK_LOAD) 248

4.5.3 SysTick current value register (STK_VAL) 249
4.5.4 SysTick calibration value register (STK_CALIB) 250
4.5.5 SysTick design hints and tips 250
4.5.6 SysTick register map 251
4.6 Floating point unit (FPU) 252
4.6.1 Coprocessor access control register (CPACR) 253
4.6.2 Floating-point context control register (FPCCR) 253
4.6.3 Floating-point context address register (FPCAR) 255
4.6.4 Floating-point status control register (FPSCR) 255
4.6.5 Floating-point default status control register (FPDSCR) 257
4.6.6 Enabling the FPU 257
4.6.7 Enabling and clearing FPU exception interrupts 258

5 Revision history 260

List of tables

Table 1. Applicable products . . . 1
Table 2. Summary of processor mode, execution privilege level, and stack usage . . . 18
Table 3. Core register set summary . . . 18
Table 4. PSR register combinations . . . 20
Table 5. APSR bit definitions . . . 21
Table 6. IPSR bit definitions . . . 22
Table 7. EPSR bit definitions . . . 23
Table 8. PRIMASK register bit definitions . . . 24
Table 9. FAULTMASK register bit definitions . . . 24
Table 10. BASEPRI register bit assignment . . . 25
Table 11. CONTROL register bit definitions . . . 25
Table 12. Ordering of memory accesses . . . 29
Table 13. Memory access behavior . . . 30
Table 14. SRAM memory bit-banding regions . . . 32
Table 15. Peripheral memory bit-banding regions . . . 32
Table 16. CMSIS functions for exclusive access instructions . . . 36
Table 17. Properties of the different exception types . . . 38
Table 18. Exception return behavior . . . 44
Table 19. Faults . . . 45
Table 20. Fault status and fault address registers . . . 47
Table 21. Cortex-M4 instructions . . . 50
Table 22. CMSIS intrinsic functions to generate some Cortex-M4 instructions . . . 59
Table 23. CMSIS intrinsic functions to access the special registers . . . 59
Table 24. Condition code suffixes . . . 67
Table 25. Memory access instructions . . . 69
Table 26. Immediate, pre-indexed and post-indexed offset ranges . . . 72
Table 27. *label*-PC offset ranges . . . 75
Table 28. Data processing instructions . . . 81
Table 29. Multiply and divide instructions . . . 109
Table 30. Saturating instructions . . . 125
Table 31. Packing and unpacking instructions . . . 134
Table 32. Instructions that operate on adjacent sets of bits . . . 138
Table 33. Branch and control instructions . . . 142
Table 34. Branch ranges . . . 143
Table 35. Floating-point instructions . . . 149
Table 36. Miscellaneous instructions . . . 180
Table 37. STM32 core peripheral register regions . . . 193
Table 38. Memory attributes summary . . . 194
Table 39. TEX, C, B, and S encoding . . . 195
Table 40. Cache policy for memory attribute encoding . . . 195
Table 41. AP encoding . . . 196
Table 42. Memory region attributes for STM32 . . . 199
Table 43. Example SIZE field values . . . 205
Table 44. MPU register map and reset values . . . 206
Table 45. NVIC register summary . . . 208
Table 46. CMSIS access NVIC functions . . . 209
Table 47. NVIC_IPRx bit assignment . . . 215
Table 48. CMSIS functions for NVIC control . . . 218

Table 49. NVIC register map and reset values . . . 219
Table 50. Summary of the system control block registers . . . 221
Table 51. Priority grouping . . . 229
Table 52. System fault handler priority fields . . . 233
Table 53. SCB register map and reset values . . . 244
Table 54. System timer registers summary . . . 246
Table 55. SysTick register map and reset values . . . 251
Table 56. Cortex-M4F floating-point system registers . . . 252
Table 57. Effect of a Floating-point comparison on the condition flags . . . 256
Table 58. Document revision history . . . 260

List of figures

Figure 1. STM32 Cortex-M4 implementation . . . 13
Figure 2. Processor core registers . . . 18
Figure 3. APSR, IPSR and EPSR bit assignment . . . 20
Figure 4. PSR bit assignment . . . 20
Figure 5. PRIMASK bit assignment . . . 24
Figure 6. FAULTMASK bit assignment . . . 24
Figure 7. BASEPRI bit assignment . . . 25
Figure 8. Memory map . . . 28
Figure 9. Bit-band mapping . . . 33
Figure 10. Little-endian example . . . 34
Figure 11. Vector table . . . 40
Figure 12. Cortex-M4 stack frame layout . . . 43
Figure 13. ASR #3 . . . 62
Figure 14. LSR #3 . . . 63
Figure 15. LSL #3 . . . 63
Figure 16. ROR #3 . . . 64
Figure 17. RRX #3 . . . 64
Figure 18. Subregion example . . . 198
Figure 19. Mapping of IP[N] fields in NVIC_IPRx registers . . . 215
Figure 20. CFSR subregisters . . . 237