Посвящение

Эта книга посвящается моей семье,
а также кошкам и собакам по соседству,
развлекавшим меня все это время

Содержание

От издательства 15

Предисловие 16

Благодарности 17

Глава 1. Введение 18
1.1. Микроконтроллеры и процессоры 18
1.2. Классификация процессоров 20
1.3. Процессоры Cortex-M23 и Cortex-M33 и архитектура Armv8-M 21
1.4. Характеристики процессоров Cortex-M23 и Cortex-M33 23
1.5. Зачем нужны два разных процессора? 25
1.6. Области применения процессоров Cortex-M23 и Cortex-M33 26
1.7. Технические характеристики 27
1.8. Сравнение с предыдущими поколениями процессоров Cortex-M 28
1.9. Преимущества процессоров Cortex-M23 и Cortex-M33 31
1.10. Общие сведения о программировании микроконтроллеров 34

Глава 2. Знакомство с программированием Cortex-M 37
2.1. Общие сведения 37
2.1.1. Комплекты для разработки 38
2.1.2. Плата для разработки 38
2.1.3. Адаптер для отладки 39
2.1.4. Ресурсы 40
2.2. Некоторые основные понятия 40
2.2.1. Сброс 40
2.2.2. Тактовые сигналы 41
2.2.3. Уровень напряжения 42
2.2.4. Входы и выходы 42
2.2.5. Введение в потоки встроенного программного обеспечения 45
2.3. Введение в программирование Arm Cortex-M 49
2.3.1. Программирование на C – типы данных 49
2.3.2. Доступ к периферийным устройствам в C 50
2.3.3. Что находится внутри образа программы? 54
2.3.4. Данные в статическом ОЗУ 57
2.3.5. Что происходит при запуске микроконтроллера? 59
2.3.6. Понимание собственной аппаратной платформы 61
2.4. Процесс разработки программного обеспечения 62
2.5. Стандарт программного интерфейса микроконтроллеров Cortex (CMSIS) 66

2.5.1. Введение в CMSIS 66
2.5.2. Что стандартизировано в CMSIS-CORE? 69
2.5.3. Использование CMSIS-CORE 70
2.5.4. Преимущества CMSIS 72
2.6. Дополнительная информация по разработке программного обеспечения 73

Глава 3. Обзор технических характеристик процессоров Cortex-M23 и Cortex-M33 75

3.1. Цели проектирования процессоров Cortex-M23 и Cortex-M33 75
3.2. Структурные схемы процессоров 77
3.2.1. Процессор Cortex-M23 77
3.2.2. Cortex-M33 78
3.3. Процессор 79
3.4. Набор команд 80
3.5. Карта памяти 82
3.6. Шинные интерфейсы 84
3.7. Защита памяти 84
3.8. Обработка прерываний и исключений 86
3.9. Функции с низким энергопотреблением 87
3.10. Функции поддержки операционной системы 88
3.11. Модуль с плавающей запятой 88
3.12. Интерфейс сопроцессора и пользовательские команды Arm 89
3.13. Поддержка отладки и трассировки 90
3.14. Поддержка проектирования многоядерных систем 90
3.15. Основные улучшения в процессорах Cortex-M23 и Cortex-M33 91
3.15.1. Сравнение процессоров Cortex-M0+ и Cortex-M23 91
3.15.2. Сравнение процессоров Cortex-M3/M4 и Cortex-M33 93
3.16. Совместимость с другими процессорами Cortex-M 95
3.17. Параметры настройки процессора 97
3.18. Введение в TrustZone 98
3.18.1. Обзор требований к безопасности 98
3.18.2. Эволюция безопасности во встраиваемых системах 99
3.18.3. Технология TrustZone для Armv8-M 103
3.19. Почему TrustZone обеспечивает лучшую безопасность? 107
3.20. Защита ресурсов встроенного ПО с помощью памяти, предназначенной только для выполнения (XOM) 111

Глава 4. Архитектура 113

4.1. Знакомство с архитектурой Armv8-M 113
4.1.1. Обзор 113
4.1.2. Развитие архитектуры Armv8-M 114
4.2. Модель программирования 116
4.2.1. Режимы и состояния процессора 116
4.2.2. Регистры 120
4.2.3. Поведение APSR (регистров флагов состояния АЛУ) 139

4.2.4. Влияние TrustZone на модель программирования 144
4.3. Система памяти 144
4.3.1. Карта памяти 146
4.3.2. Разделение адресных пространств внутри TrustZone 148
4.3.3. Пространство управления системой (SCS) и блок управления системой (SCB) 150
4.3.4. Стековая память 150
4.3.5. Настройка и доступ к указателям стека и регистрам границы стека 155
4.3.6. Модуль защиты памяти (MPU) 157
4.4. Исключения и прерывания 158
4.4.1. Что такое исключения 158
4.4.2. TrustZone и исключения 161
4.4.3. Контроллер вложенных векторных прерываний (NVIC) 161
4.4.4. Управление прерываниями с помощью CMSIS-CORE 164
4.4.5. Таблицы векторов 164
4.4.6. Обработка отказов 165
4.5. Отладка 167
4.6. Сброс и последовательность сброса 170

Глава 5. Набор команд 174
5.1. Основные сведения 174
5.1.1. Информация о данной главе 174
5.1.2. Общие сведения о наборе команд в процессорах Arm Cortex®-M 175
5.2. Особенности набора команд в различных процессорах Cortex-M 176
5.3. Понимание синтаксиса языка ассемблера 180
5.4. Использование суффикса в команде 184
5.5. Унифицированный язык ассемблера (UAL) 186
5.6. Набор команд – пересылка данных внутри процессоров 187
5.6.1. Обзор 187
5.6.2. Пересылка данных между регистрами 188
5.6.3. Создание непосредственных данных 189
5.6.4. Команды доступа к специальным регистрам 190
5.6.5. Доступ к регистрам с плавающей запятой 192
5.6.6. Создание непосредственных данных с плавающей запятой 193
5.6.7. Пересылка данных между регистром и регистром сопроцессора 194
5.7. Набор команд – доступ к памяти 194
5.7.1. Обзор 194
5.7.2. Единичный доступ к памяти 195
5.7.3. Загрузка/сохранение относительно SP 197
5.7.4. Предындексный и постиндексный режимы адресации 199
5.7.5. Дополнительный сдвиг в регистре смещения (Barrel shifter) 201
5.7.6. Чтение литеральных данных 202
5.7.7. Команды групповой загрузки/сохранения 203
5.7.8. Команды PUSH/POP 206
5.7.9. Команды доступа для непривилегированных пользователей 207

5.7.10. Команды доступа к памяти FPU 208
5.7.11. Монопольный доступ 211
5.7.12. Команды загрузки–получения и сохранения–освобождения данных 215
5.8. Набор команд – арифметические операции 220
5.9. Набор команд – логические операции 224
5.10. Набор команд – операции сдвига и циклического сдвига 225
5.11. Набор команд – преобразование данных (расширение и изменение порядка) 228
5.12. Набор команд – обработка битовых полей 230
5.13. Набор команд – операции насыщения 232
5.14. Набор команд – управление потоком выполнения программы 235
5.14.1. Обзор 235
5.14.2. Команды перехода 235
5.14.3. Вызов функции 236
5.14.4. Условный переход 238
5.14.5. Команды сравнения и перехода (CBZ, CBNZ) 240
5.14.6. Условное выполнение с использованием блока IT команды 241
5.14.7. Команды табличного перехода TBB и TBH 243
5.15. Набор команд – расширение DSP 246
5.15.1. Обзор 246
5.15.2. Концепция SIMD 247
5.15.3. SIMD и арифметические команды насыщения 248
5.15.4. Команды умножения и MAC 251
5.15.5. Команды упаковки и распаковки 255
5.16. Набор команд – команды поддержки операций с плавающей запятой 256
5.16.1. Обзор поддержки операций с плавающей запятой в процессорах Armv8-M 256
5.16.2. Включение FPU 256
5.16.3. Команды вычислений с плавающей запятой 257
5.17. Набор команд – команды, связанные с исключениями 262
5.18. Набор команд – команды, относящиеся к спящему режиму 265
5.19. Набор команд – команды барьера (блокирования) памяти 267
5.20. Набор команд – команды поддержки TrustZone 269
5.21. Набор команд – команды поддержки сопроцессора и пользовательские команды Arm 271
5.22. Набор команд – другие функции 276
5.23. Доступ к специальным регистрам с помощью CMSIS-CORE 277

Глава 6. Система памяти 280

6.1. Общий обзор системы памяти 280
6.1.1. Что входит в систему памяти? 280
6.1.2. Особенности системы памяти 281
6.1.3. Основные изменения в Cortex-M23/M33 по сравнению с предыдущими процессорами Cortex-M 282

6.2. Карта памяти 283
6.3. Типы и атрибуты памяти 287
6.3.1. Классификация типов памяти 287
6.3.2. Обзор атрибутов памяти 287
6.3.3. Атрибуты памяти для карты памяти по умолчанию 291
6.4. Управление правами доступа 292
6.4.1. Обзор управления правами доступа 292
6.4.2. Механизмы управления доступом 293
6.4.3. Различия между SAU/IDAU и MPU 294
6.4.4. Права доступа по умолчанию 295
6.5. Порядок байтов в памяти 297
6.6. Поддержка выравнивания данных и доступа к ним без выравнивания ... 299
6.7. Поддержка монопольного доступа 302
6.8. Команды упорядочения памяти и барьеров памяти 306
6.9. Поддержка состояния ожидания и ответов на ошибки шины 308
6.10. Порт однотактного ввода-вывода – только для Cortex-M23 310
6.11. Системы памяти в микроконтроллерах 311
6.11.1. Требования к памяти 311
6.11.2. Дизайн шинных систем 314
6.11.3. Управление безопасностью 316
6.12. Обсуждение настройки программного обеспечения 319
6.12.1. Управление питанием на уровне шины 319
6.12.2. Расширение безопасности TrustZone 319
6.12.3. Использование команд групповой загрузки и сохранения 319

Глава 7. Поддержка TrustZone в системе памяти 321

7.1. Обзор 321
7.1.1. Об этой главе 321
7.1.2. Атрибуты безопасности памяти 321
7.2. SAU и IDAU 323
7.3. Объединенные и не объединенные в банк регистры 325
7.3.1. Обзор 325
7.3.2. Незащищенный псевдоним пространства управления системой (SCS) 327
7.4. Команды тестирования целевых объектов (TT) и идентификационные номера областей 328
7.4.1. Зачем нужны команды тестирования? 328
7.4.2. Команды TT 330
7.4.3. Идентификационные номера областей 331
7.5. Контроллер защиты памяти и контроллер защиты периферийных устройств 334
7.6. Периферийные устройства, отвечающие требованиям безопасности 338

Глава 8. Исключения и прерывания – обзор архитектуры 340

8.1. Обзор исключений и прерываний 340

8.1.1. Необходимость в исключениях и прерываниях....340
8.1.2. Основные принципы работы с прерываниями от периферийных устройств....341
8.1.3. Введение в NVIC....342
8.2. Типы исключений....344
8.3. Обзор управления прерываниями и исключениями....347
8.4. Анализ последовательности исключения....349
8.4.1. Обзор....349
8.4.2. Принятие запроса исключения....350
8.4.3. Последовательность вхождения в исключение....350
8.4.4. Выполнение обработчика исключения....351
8.4.5. Последовательность возврата исключения....352
8.5. Определения уровней приоритета исключений....353
8.5.1. Обзор уровней приоритета исключений и прерываний....353
8.5.2. Группирование приоритетов в Armv8-M Mainline....356
8.5.3. Определение приоритета защищенных исключений и прерываний....357
8.5.4. Объединенные в банк регистры маскирования прерываний....360
8.6. Таблица векторов и регистр смещения таблицы векторов (VTOR)....360
8.7. Вхождение в прерывание и действия в состоянии ожидания....363
8.8. Целевые состояния исключений и прерываний в системах TrustZone....368
8.9. Стековые кадры....371
8.9.1. Обзор загрузки в стек и извлечения из стека....371
8.9.2. Интерфейс функций языка C....372
8.9.3. Обработчик исключений на языке C....372
8.9.4. Форматы стековых кадров....374
8.9.5. Какой указатель стека используется для загрузки и извлечения?....381
8.10. EXC_RETURN....382
8.11. Классификация синхронных и асинхронных исключений....388

Глава 9. Управление исключениями и прерываниями....389

9.1. Обзор управления исключениями и прерываниями....389
9.1.1. Доступ к функциям управления исключениями....389
9.1.2. Базовое управление прерываниями в CMSIS-CORE....390
9.1.3. Дополнительные функции управления прерываниями в CMSIS-CORE....393
9.1.4. Управление системными исключениями....395
9.2. Подробные сведения о регистрах NVIC для управления прерываниями....395
9.2.1. Краткое описание....395
9.2.2. Регистры разрешения прерываний....396
9.2.3. Регистры установки и отмены ожидания обработки прерывания....398
9.2.4. Активное состояние....399
9.2.5. Регистр(ы) назначения прерыванию состояния безопасности....400
9.2.6. Уровень приоритета....401
9.2.7. Регистр программной генерации прерывания (только Armv8-M Mainline)....402

9.2.8. Регистр типа контроллера прерываний 403
9.2.9. Улучшения функциональности NVIC 404
9.3. Подробная информация о регистрах SCB для управления системными исключениями 404
9.3.1. Краткое описание 404
9.3.2. Регистр управления и состояния прерываний (SCB->ICSR) 406
9.3.3. Регистры уровней приоритета обработчиков системных исключений (SCB->SHP[n]) 407
9.3.4. Регистр управления прерываниями и сбросом (SCB->AIRCR) 409
9.3.5. Регистр состояния и управления обработчиков системных исключений (SCB->SHCSR) 411
9.4. Подробная информация о специальных регистрах для маскирования исключений или прерываний 414
9.4.1. Обзор регистров маскирования прерываний 414
9.4.2 Регистр PRIMASK 416
9.4.3. Регистр FAULTMASK 417
9.4.4. Регистр BASEPRI 419
9.5. Определение таблицы векторов в программировании 422
9.5.1. Таблица векторов в стартовом коде 422
9.5.2. Перемещение таблицы векторов 423
9.6. Оптимизация задержки обработки прерывания и исключений 425
9.6.1. Что такое задержка обработки прерывания 425
9.6.2. Прерывания при выполнении многотактных команд 427
9.6.3. Поддержка прямого перехода в цепочке исключений (Tail chaining) 428
9.6.4. Исключения позднего прибытия (Late arrival) 429
9.6.5. Вытеснение при извлечении 429
9.6.6. Отложенная загрузка в стек (Lazy stacking) 430
9.7. Советы и подсказки 431

Глава 10. Функции снижения энергопотребления и управления системой 434

10.1. Вопросы по низкому энергопотреблению 434
10.1.1. Почему важно низкое энергопотребление? 434
10.1.2. Что означает низкое энергопотребление? И как его измерить? 435
10.2. Характеристики процессоров Cortex-M23 и Cortex-M33 с низким энергопотреблением 437
10.2.1. Характеристики процессоров с низким энергопотреблением 437
10.2.2. Режимы сна 438
10.2.3. Регистр управления системой (SCB->SCR) 439
10.2.4. Переход в спящий режим 440
10.2.5. Функция перехода в спящий режим при выходе (Sleep-on-Exit) 442
10.2.6. Отправка события в состоянии ожидания обработки прерывания (SEVONPEND) 443
10.2.7. Продление сна / задержка пробуждения 444
10.2.8. Контроллер пробуждающих прерываний (WIC) 444

10.2.9. Интерфейс обмена событиями 447
10.2.10. Влияние TrustZone на поддержку спящего режима 451
10.3. Подробнее о командах WFI, WFE и SEV 452
10.3.1. Использование команд WFI, WFE и SEV при программировании 452
10.3.2. Когда использовать команду WFI 452
10.3.3. Когда использовать команду WFE 454
10.3.4. Условия пробуждения 456
10.4. Разработка приложений с низким энергопотреблением 459
10.4.1. Начало работы 459
10.4.2. Снижение активной мощности 459
10.4.3. Сокращение числа активных циклов 462
10.4.4. Снижение тока в спящем режиме 462
10.5. Блок управления системой (SCB) и функции управления системой 463
10.5.1. Регистры в блоке управления системой 463
10.5.2. Базовый регистр идентификатора процессора 466
10.5.3. AIRCR – генерация программного сброса (SYSRESETREQ) 467
10.5.4. AIRCR – очистка всех состояний прерываний (VECTCLRACTIVE) 468
10.5.5. CCR – регистр управления конфигурирацией (SCB->CCR, 0xE000ED14) 469
10.6. Дополнительный регистр управления 472
10.6.1. Обзор дополнительного регистра управления 472
10.6.2. Дополнительный регистр управления для процессора Cortex-M23 472
10.6.3. Дополнительный регистр управления для процессора Cortex-M33 473
10.7. Другие регистры в блоке управления системой 474

Глава 11. Функции поддержки операционной системы 475

11.1. Обзор функций поддержки операционной системы 475
11.2. Таймер SysTick 477
11.2.1. Назначение таймера SysTick 477
11.2.2. Операции с таймером SysTick 478
11.2.3. Использование таймера SysTick 481
11.2.4. Расширение TrustZone и таймер SysTick 484
11.2.5. Другие соображения 486
11.3. Объединенные в банк указатели стека 486
11.3.1. Преимущества объединенных в банк указателей стека 486
11.3.2. Операции с отдельными указателями стека 487
11.3.3. Использование объединенных в банк указателей стека в устройствах без ОС 490
11.4. Проверка границы стека 492
11.4.1. Общие сведения 492
11.4.2. Доступ к регистрам границы стека 494
11.4.3. Защита указателя (указателей) основного стека 495
11.5. Исключения SVCall и PendSV 497
11.5.1. Обзор исключений SVCall и PendSV 497

11.5.2. Исключение SVCall 498
11.5.3. PendSV 502
11.6. Непривилегированный уровень выполнения и модуль защиты памяти (MPU) 507
11.7. Монопольный доступ 509
11.8. Как должна работать ОСРВ в среде TrustZone? 511
11.9. Основные принципы работы ОСРВ в процессорах Cortex-M 513
11.9.1. Запуск простой операционной системы 513
11.9.2. Переключение контекста 515
11.9.3. Переключение контекста в действии 516

Список литературы 526

Сокращения 528

Предметный указатель 531

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте *www.dmkpress.com*, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу *dmkpress@gmail.com*; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу *http://dmkpress.com/authors/publish_book/* или напишите в издательство по адресу *dmkpress@gmail.com*.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу *dmkpress@gmail.com*. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты *dmkpress@gmail.com*.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Предисловие

Прошло немало времени с момента выхода последней книги из серии «Полное руководство по процессорам Arm Cortex-M». В настоящее время на рынок выходят продукты на базе Cortex-M23 и Cortex-M33, и многие из них предлагают сложные функции безопасности – определение «современные микроконтроллеры», похоже, сделало большой шаг вперед. Хотя механизмы этих функций безопасности разработаны для небольших электронных устройств с низким энергопотреблением, многие из этих технологий безопасности, в принципе, схожи с технологиями высокопроизводительных вычислительных систем.

Хотя безопасность – очень обширная тема, ключевым аспектом безопасности миллиардов подключенных устройств является «корень доверия», такой как секретные ключи и механизмы безопасной загрузки, которые нуждаются в защите. Технология Arm TrustZone, поддерживаемая процессорами Cortex-M23 и Cortex-M33, позволяет защитить эти критически важные для безопасности элементы в защищенной среде обработки (безопасный мир). В то же время приложения, работающие в обычной среде обработки (небезопасный мир), могут быть легко разработаны и способны использовать функции безопасности, предоставляемые защищенным встроенным ПО. Вместе с другими проектами компании ARM, такими как Platform Security Architecture (PSA) и Trusted Firmware-M, разработчики программного обеспечения могут легко создавать программное обеспечение для защищенных устройств интернета вещей (IoT).

Взаимодействие между защищенными и незащищенными программными компонентами вносит новый уровень сложности в архитектуру. В то время как разработка незащищенного программного обеспечения по-прежнему очень похожа на ту, что требовалась для предыдущих процессоров Cortex-M, разработчики защищенного программного обеспечения, создающие защищенное встроенное ПО, теперь должны быть знакомы с рядом новых архитектурных особенностей, которые были внедрены в архитектуре Armv8-M.

В отличие от предыдущих изданий «Полного руководства по процессору ArmCortex-M», эта книга посвящена архитектуре процессоров Arm Cortex-M23 и Cortex-M33. Чтобы разработчики программного обеспечения могли создавать безопасные решения на основе архитектуры Armv8-M, необходимо, как указано в этой книге, предоставить подробное объяснение архитектурных особенностей, которые в настоящее время недоступны нигде. В результате объем информации, посвященной примерам на уровне приложений и рекомендациям по использованию инструментов разработки, сокращен, чтобы книга оставалась доступной для читателей. Поскольку примеры прикладного уровня и инструкции по применению различных инструментов доступны у других производителей, такие материалы не рассматриваются в этой книге.

Надеюсь, вы найдете эту книгу полезной и достойной прочтения.

Джозеф Ю

Благодарности

Я хотел бы поблагодарить своих друзей, которые оказали мне огромную поддержку во время работы над книгой, особенно Ивана, который в течение 5 месяцев помогал мне ее вычитывать. (Он не знал, насколько плохо я пишу, когда согласился помочь … извини!)

Я хотел бы поблагодарить отдел маркетинга компании ARM за поддержку в этом проекте, а также Томаса Грокатта (Thomas Grocutt) из ARM Research за предоставление технической информации по безопасному программному обеспечению, Санджива Сарпала (Sanjeev Sarpal) из Advanced Solutions Nederland B. V. за поддержку в разработке цифровых фильтров и Пола Бекманна (Paul Beckmann) из DSP Concepts, который внес свой вклад в подготовку глав, посвященных цифровой обработке сигналов.

И конечно, спасибо всем читателям, оставлявшим отзывы о моих предыдущих книгах, что было большим подспорьем при подготовке к написанию этой книги. И наконец, спасибо сотрудникам издательского дома Elsevier, чей вклад позволил издать эту книгу.

С уважением,
Джозеф Ю

Глава 1

Введение

1.1. Микроконтроллеры и процессоры

Процессоры используются в большинстве электронных продуктов: телефонах, телевизорах, пультах дистанционного управления, бытовой технике, электронных игрушках, компьютерах и аксессуарах к ним, транспорте, системах охраны зданий, банковских картах и т. д. Во многих случаях эти процессоры размещаются внутри микросхем, называемых микроконтроллерами, которые предназначены для обслуживания широкого спектра приложений. Микроконтроллеры являются программируемыми устройствами, требующими от разработчиков создания программного обеспечения, работающего на этих микросхемах. Довольно часто мы называем эти продукты встраиваемыми системами (embedded system), поскольку микроконтроллеры внутри них обычно хорошо спрятаны.

Для того чтобы микроконтроллеры могли взаимодействовать с внешней средой, они содержат ряд функциональных блоков, называемых периферийными устройствами. Например, аналого-цифровой преобразователь (АЦП) позволяет измерять напряжение внешних сигналов от датчиков, а последовательный периферийный интерфейс (SPI) позволяет управлять внешним модулем ЖК-дисплея. Разные микроконтроллеры имеют разную периферию, а микроконтроллеры разных производителей могут иметь схожую периферию, но с разными программными моделями и функциями.

Чтобы собирать и обрабатывать данные с периферийных устройств и управлять различными интерфейсами, нам нужны процессоры в микроконтроллерах и программное обеспечение, которое работает на этих процессорах. Помимо процессора, внутри микроконтроллера есть много других компонентов. На рис. 1.1 показаны общие компоненты микроконтроллера.

На схеме много сокращений и аббревиатур. Их описание приведено в табл. 1.1.

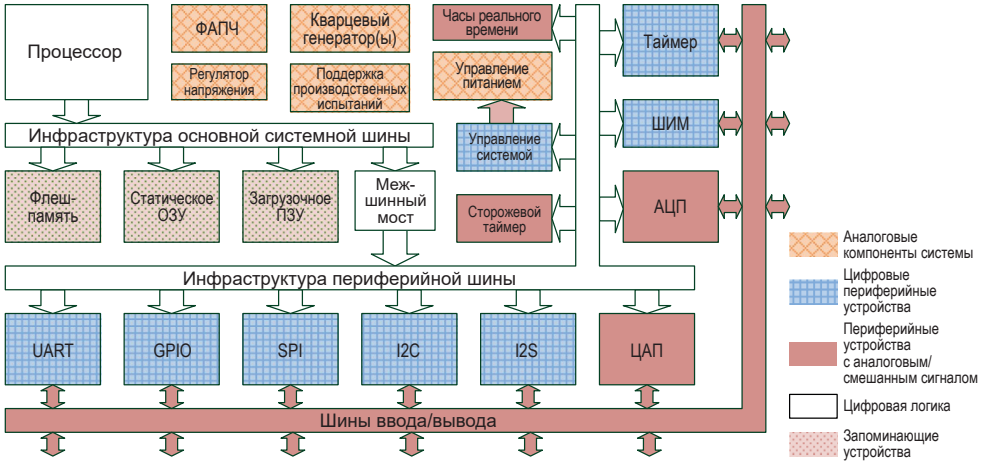


Рис. 1.1 ❖ Упрощенная структурная схема микроконтроллера

Таблица 1.1. Типовые компоненты микроконтроллера

Boot ROM	Read-Only Memory, постоянное (энергонезависимое) запоминающее устройство, ПЗУ. В данном случае хранит процедуру начальной загрузки (boot)
Flash memory	Специальный тип ПЗУ, который можно многократно перепрограммировать, как правило, служит для хранения программного кода
SRAM	Static Random Access Memory, статическое оперативное запоминающее устройство (СОЗУ), память с произвольным доступом (энергозависимая). Служит для хранения данных и констант
PLL	Phase Lock Loop, контур фазовой автоподстройки частоты (ФАПЧ): устройство для генерации программируемой тактовой частоты на основе опорного тактового сигнала
RTC	Real Time Clock, часы реального времени. Малопотребляющий таймер для подсчета секунд, а в некоторых случаях также имеет функции минут, часов и календаря
GPIO	General Purpose Input/Output, интерфейс ввода/вывода общего назначения. Периферийное устройство с параллельным интерфейсом передачи данных для управления внешними устройствами и считывания состояния внешних сигналов.
UART	Universal Asynchronous Receiver/Transmitter, универсальный асинхронный приемопередатчик. Периферийное устройство для передачи данных по простому последовательному протоколу.
I2C	Inter-Integrated Circuit, интерфейс I2C. Периферийное устройство для передачи данных по последовательному протоколу. В отличие от UART, требуется тактовый сигнал, что может обеспечить более высокую скорость передачи
SPI	Serial Peripheral Interface, последовательный периферийный интерфейс. Трехпроводный последовательный интерфейс связи для внешних периферийных устройств
I2S	Integrated Interchip Sound, интерфейс I2S. Последовательный интерфейс связи, предназначен для передачи аудиоданных
PWM	Pulse Width Modulator, широтно-импульсный модулятор, ШИМ. Периферийное устройство для формирования выходного сигнала с программируемой скважностью
ADC	Analog to Digital Converter, аналогово-цифровой преобразователь. Устройство для преобразования информации об уровне аналогового сигнала в цифровую форму
DAC	Digital to Analog Converter, цифроаналоговый преобразователь. Устройство для преобразования значений цифровых данных в аналоговые сигналы
Watchdog timer	Сторожевой таймер, WDT. Программируемый таймер, отслеживающий выполнение программы процессором. Если WDT включен, запущенная программа должна обновлять счетчик сторожевого таймера до истечения заданного промежутка времени. Если программа «зависает», таймер отключается, что может быть использовано для запуска сброса или аварийного прерывания

Некоторые сложные микроконтроллерные устройства могут содержать гораздо больше компонентов. Во многих случаях также используются контроллеры прямого доступа к памяти (DMA), ускорители шифрования данных и сложные интерфейсы, такие как USB и Ethernet. Некоторые микроконтроллеры тоже могут содержать более одного процессора.

Различные микроконтроллерные устройства могут иметь разные процессоры, объем памяти, периферийные устройства, корпуса и т. д. Таким образом, даже два микроконтроллера с одним и тем же процессором внутри могут иметь разное распределение памяти и периферийные регистры. В результате для выполнения одной и той же прикладной функции программный код, необходимый для различных микроконтроллерных продуктов, может быть совершенно разным.

1.2. Классификация процессоров

Существует множество различных типов процессоров и способов их классификации. Простой метод классификации основан на ширине канала передачи данных (например, канала передачи данных между АЛУ и банком регистров). С этой точки зрения существуют 8-битные, 16-битные, 32-битные и 64-битные процессоры. Другой метод классификации основан на их приложениях. Например, компания ARM классифицирует свои процессорные продукты по следующим категориям.

Прикладные процессоры. К ним относятся устройства, используемые в качестве основного процессора в компьютерах, серверах, планшетах, мобильных телефонах или телевизорах Smart TV. Обычно эти процессоры поддерживают полнофункциональные операционные системы, такие как Linux, Android или Windows, и имеют пользовательский интерфейс, позволяющий пользователям управлять устройством. Как правило, такие процессоры работают на высокой тактовой частоте и обеспечивают очень высокую производительность.

Процессоры реального времени. Процессоры реального времени часто используются в системах, требующих высокой производительности, но не требующих полнофункциональной операционной системы, и могут быть надежно спрятаны внутри продукта. Во многих случаях в этих процессорах для планирования задач и обмена сообщениями между задачами используются операционные системы реального времени (ОСРВ). Процессоры реального времени находят применение в модемах телефонов, работающих в основной полосе частот, в качестве специализированных микроконтроллеров в автомобильных системах или в контроллерах жестких дисков и твердотельных накопителей (SSD).

Процессоры для микроконтроллеров. Процессоры для микроконтроллеров используются в большинстве продуктов для микроконтроллеров (хотя вместо них в некоторых микроконтроллерах могут использоваться прикладные процессоры или процессоры реального времени). Конструкция этих процес-

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru