

МИКРОКОНТРОЛЛЕРНЫЕ СИСТЕМЫ НА БАЗЕ ЯДРА ARM 7

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к самостоятельной работе и к выполнению расчётно-графической работы
по дисциплине “Микроконтроллерные системы автоматики и управления”
для студентов направления подготовки
6.050102 – “Компьютерная инженерия”

Обсуждено и рекомендовано
на заседании кафедры
информационных и компьютерных систем
Протокол № 2
от 27.06.2014 г.

Мікроконтролерні системи на базі ядра ARM 7. Методичні вказівки до самостійної роботи та до виконання розрахунково-графічної роботи з дисципліни "Мікроконтролерні системи автоматики та управління" для студентів напряму підготовки 6.050102 – "Комп'ютерна інженерія"/ Укл.: А.І. Роговенко, О.В. Красножон, А.В. Красножон – Чернігів: ЧДТУ, 2014. – 87 с. Рос. мовою.

Составители: РОГОВЕНКО АНДРЕЙ ИВАНОВИЧ, старший преподаватель кафедры информационных и компьютерных систем
КРАСНОЖОН АЛЕКСЕЙ ВАСИЛЬЕВИЧ, ассистент кафедры информационных и компьютерных систем
КРАСНОЖОН АНДРЕЙ ВАСИЛЬЕВИЧ, кандидат технических наук, старший преподаватель кафедры электрических систем и сетей

Ответственный за выпуск: КАЗИМИР ВЛАДИМИР ВИКТОРОВИЧ, заведующий кафедрой информационных и компьютерных систем, доктор технических наук, профессор

Рецензент: ВЕРВЕЙКО АЛЕКСАНДР ИВАНОВИЧ, кандидат технических наук, доцент, доцент кафедры информационных и компьютерных систем Черниговского национального технологического университета

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 САМОСТОЯТЕЛЬНАЯ РАБОТА.....	6
1.1 Цель и задачи дисциплины	6
1.2 Содержание дисциплины.....	7
1.2.1 Введение	7
1.2.2 Интерфейсы микроконтроллерных систем	7
1.2.3 Последовательный интерфейс SPI.....	8
1.2.4 Последовательный интерфейс I ² C	8
1.2.5 Последовательная шина 1-Wire	8
1.2.6 Основы теории автоматического управления и регулирования.....	8
1.2.7 Фундаментальные принципы управления	8
1.2.8 Понятие о передаточной функции системы автоматического управления	8
1.2.9 Параметры и характеристики систем автоматического управления	9
1.2.10 Классификация систем автоматического управления.....	9
1.2.11 Средства автоматизации для построения систем автоматического управления	9
1.2.12 Внутренняя организация промышленных микроконтроллерных систем автоматического управления	9
1.2.13 Развитие микроконтроллерных систем автоматического управления в промышленности	9
1.2.14 Возникновение и развитие промышленных контроллеров	9
1.2.15 Классификация промышленных контроллеров	10
1.2.16 Эволюция развития средств автоматизированных систем управления технологическими процессами	10
1.3 Распределение объема самостоятельной работы	10
1.4 Вопросы для сдачи зачёта по дисциплине	10
2 ХАРАКТЕРНЫЕ ОСОБЕННОСТИ МИКРОКОНТРОЛЛЕРОВ ARM7	13
2.1 Типовые характеристики микроконтроллеров с ядром ARM7.....	13
2.2 Особенности архитектурного построения ядра ARM7	13
2.3 Микроконтроллерное ядро ARM7TDMI-S.....	14
2.4 Режимы работы и регистровые модели микроконтроллерного ядра ARM7TDMI-S в состоянии ARM	16
2.5 Режимы работы и регистровые модели микроконтроллерного ядра ARM7TDMI-S в состоянии THUMB.....	20
2.6 Время реакции на прерывание	20
2.7 Организация системной памяти микроконтроллеров ARM7	21
2.8 Структура микроконтроллера ARM7	24
3 СИСТЕМНЫЕ УСТРОЙСТВА МИКРОКОНТРОЛЛЕРОВ СЕМЕЙСТВА ARM7.....	26
3.1 Блок управления системой	26

3.1.1	Генератор, стабилизированный кварцевым резонатором.....	27
3.1.2	Входы внешних прерываний.....	27
3.1.3	Система фазовой автоподстройки частоты.....	33
3.1.4	Узел управления потребляемой мощностью	35
3.1.5	Узел сброса микроконтроллера	39
3.1.6	Делитель частоты для шины периферийных устройств VPB.....	41
3.1.7	Таймер “пробуждения” микроконтроллера	42
3.2	Контроллер векторов прерываний VIC	44
3.2.1	Регистры контроллера векторов прерываний	45
3.2.2	Ложные прерывания	51
3.2.3	Рекомендации по программированию VIC	51
3.3	Блок коммутации выводов.....	52
3.4	Управление линиями ввода/вывода общего назначения.....	55
3.4.1	Регистры управления вводом/выводом общего назначения.....	56
3.4.2	Рекомендации по использованию линий ввода/вывода общего назначения	58
4	РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА	59
4.1	Цель и порядок выполнения расчётно-графической работы.....	59
4.2	Требования к отчёту о выполнении расчётно-графической работы	60
4.3	Требования и рекомендации к выполнению расчётно-графической работы.....	61
4.4	Варианты заданий.....	61
4.5	Примеры генерации сигналов про помощи ЦАП	85
4.5.1	Пример генерации прямоугольного сигнала	85
4.5.2	Пример генерации пилообразного сигнала	86
	РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА.....	88

ВВЕДЕНИЕ

В последнее время однокристальные микроконтроллеры находят самое разнообразное применение во многих областях человеческой деятельности, причём, границы их применения расширяются с каждым днём. С появлением первых микроконтроллеров их сложность постоянно возрастает за счёт разработки более совершенных архитектур, новых решений в области полупроводниковых технологий, добавления новых инструкций. Для решения задач с высоким уровнем сложности применяются микроконтроллерные системы.

Значительных успехов в области создания 16/32-разрядных микроконтроллерных ядер достигла британская компания ARM (Advanced RISC Machines). Благодаря высокой производительности и выгодному соотношению цена/качество эти микроконтроллеры приобрели широкую популярность в различных областях человеческой деятельности. Фактически, микроконтроллеры с ядром ARM выделились в отдельный класс микроконтроллеров для встраиваемых приложений.

Компания ARM не имеет собственных мощностей полупроводникового производства, поэтому микроконтроллерные ядра, поставляемые ею в различных видах (описанные на языках VHDL или Verilog, в электрических схемах или топологической макроячейке), реализуются в дальнейшем такими крупными производителями, как Atmel, Philips, Samsung, Sony, Texas Instruments и т.д.

Согласно рекомендациям производителя, микроконтроллеры с ядром ARM предназначены для использования в автомобилестроении, медицине, промышленной и бытовой электронике (в том числе и с батарейным питанием), компьютерных сетях и системах. Это возможно благодаря высоким показателям надёжности, относительно низкой мощности потребления и наличию различных режимов функционирования.

Микроконтроллеры с ядром ARM различных версий поддерживаются многими программными продуктами – отладчики, компиляторы C/C++, эмуляторы, операционные системы реального времени, драйверы низкого уровня, высокоуровневые приложения – различных всемирно известных производителей: Keil Software, Microsoft, Philips и другие.

Производитель определяет ядро ARM версии 7 как универсальное ядро 32-разрядного RISC-микроконтроллера с малым энергопотреблением, предназначенное для использования в различных заказных и специализированных интегральных схемах.

Настоящие методические указания предназначены для ознакомления и начального изучения внутренней организации микроконтроллеров с ядром ARM7 производства компании NXP (подразделения компании Philips), особенностей их функционирования, а также для выполнения расчётно-графической работы по дисциплине “Микроконтроллерные системы автоматики и управления”. Они не охватывают всех доступных возможностей и нюансов функционирования этого семейства микроконтроллеров, поэтому при необходимости следует обращаться к другим литературным источникам и ресурсам.

1 САМОСТОЯТЕЛЬНАЯ РАБОТА

1.1 Цель и задачи дисциплины

Дисциплина "Микроконтроллерные системы автоматики и управления" входит в состав цикла дисциплин самостоятельного выбора учебного заведения из раздела профессиональной и практической подготовки образовательно-профессиональной программы высшего образования по направлению 6.050102 – "Компьютерная инженерия" и рассматривается кафедрой как базовая для изучения дисциплин "Системное программное обеспечение", "Проектирование специализированных компьютерных систем", "Программное обеспечение специализированных компьютерных систем". Необходимыми предпосылками для преподавания дисциплины "Микроконтроллерные системы автоматики и управления" является усвоение студентами материала дисциплин "Дискретная математика", "Программирование", "Теория электрических цепей", "Компьютерная электроника", "Компьютерная схемотехника", "Большие интегральные схемы микропроцессоров и программируемой логики", "Программирование на языке C++".

Целью изложения дисциплины "Микроконтроллерные системы автоматики и управления" является формирование у студентов научного уровня инженерного мышления будущего специалиста, содержащего в себе теоретическую базу, которая необходима при усвоении прикладных вопросов разработки и эксплуатации аппаратных средств микроконтроллерных систем управления. В результате изучения дисциплины студенты должны знать:

- основные понятия и определения микроконтроллерной техники;
- архитектурные и структурные элементы микроконтроллерных систем и средства их построения;
- характеристики современных аппаратных интерфейсов микроконтроллерных систем и способы их применения;
- основные понятия и определения теории автоматического управления и регулирования;
- законы осуществления управления;
- основные характеристики и параметры систем автоматического управления;
- формулы расчета передаточных функций систем управления с большим количеством элементарных звеньев и разными типами обратной связи;
- основные направления развития и использования микроконтроллерных систем для автоматизации процессов управления в промышленности;
- характерные особенности промышленных систем автоматического управления.

В результате освоения учебной дисциплиной студенты должны уметь:

- реализовать с помощью языка программирования алгоритм управления объектом в соответствии с приведённым текстовым описанием и загрузить программу в микроконтроллер;

- наглядно оценивать правильность реализации указанного алгоритма управления, находить и исправлять допущенные ошибки;
- определять необходимое количество и типы интерфейсов связи при разработке микроконтроллерной системы и организовывать обмен данными между различными устройствами в составе системы;
- составлять в общем виде выражение для передаточной функции сложной системы, если известны передаточные функции её звеньев;
- обращаться и использовать необходимое в курсе дисциплины программное и аппаратное обеспечение.

Значение дисциплины для реализации требований квалификационной характеристики специалиста и изучения последующих дисциплин заключается в том, что её изучение способствует формированию алгоритмического мышления будущего специалиста, формирования у него практических навыков написания программ для микроконтроллеров, создаёт базу, необходимую при изучении других дисциплин.

В связи с интенсивным развитием средств микроконтроллерной техники, а также интерфейсов связи основное внимание уделено изучению современных аппаратных интерфейсов микроконтроллерных систем, а также приобретению знаний, навыков и практического опыта в программировании микроконтроллеров с применением современных программных средств – интегрированных сред разработки.

Преподавание дисциплины "Микроконтроллерные системы автоматики и управления" обусловлено необходимостью формирования у студентов четкой системы представлений о целостном комплексе проблем, которые должны быть решены в процессе разработки узлов промышленных систем автоматического управления с применением современных микроконтроллеров.

Дисциплина является базовой для изучения таких дисциплин, как "Системное программное обеспечение", "Проектирование специализированных компьютерных систем", "Программное обеспечение специализированных компьютерных систем".

1.2 Содержание дисциплины

1.2.1 Введение

Предмет и цели изучения дисциплины "Микроконтроллерные системы автоматики и управления. Основные задачи, решаемые в процессе изложения дисциплины. Структура учебного курса. Учебно-методическая литература по дисциплине. Основные понятия и определения микроконтроллерной техники.

1.2.2 Интерфейсы микроконтроллерных систем

Понятие об аппаратных интерфейсах, виды их совместимости. Классификация интерфейсов микроконтроллерных систем по различным признакам. Характеристики интерфейсов. Назначение интерфейсов. Примеры использования интерфейсов.

1.2.3 Последовательный интерфейс SPI

Общая характеристика интерфейса. Внутреннее устройство. Режимы функционирования и их конфигурация. Протокол обмена. Особенности подключения устройств.

1.2.4 Последовательный интерфейс I²C

Общая характеристика интерфейса. Внутреннее устройство. Режимы функционирования и их конфигурация. Протокол обмена. Особенности подключения устройств.

1.2.5 Последовательная шина 1-Wire

Общая характеристика интерфейса. Внутреннее устройство. Режимы функционирования и их конфигурация. Протокол обмена. Особенности подключения устройств.

1.2.6 Основы теории автоматического управления и регулирования

Цели и необходимость изучения теории автоматического управления и регулирования. История возникновения теории автоматического управления и регулирования. Примеры применения теории автоматического управления и регулирования в повседневной деятельности человека. Основные определения и понятия. Основная задача автоматического управления. Управляющая величина. Структура системы автоматического управления. Понятия о входных и выходных величинах.

1.2.7 Фундаментальные принципы управления

Принцип управления по возмущению. Принцип управления по задающему воздействию. Принцип управления по отклонению. Комбинированный принцип управления. Функциональные схемы систем автоматического управления с разными принципами управления. Преимущества и недостатки каждого из принципов управления. Цели и границы применимости каждого из рассмотренных принципов управления.

1.2.8 Понятие о передаточной функции системы автоматического управления

Понятие об элементарных звеньях системы автоматического управления. Виды элементарных звеньев и их характеристика. Определение и общий вид передаточной функции системы автоматического управления. Передаточная функция системы автоматического управления с разными типами соединения звеньев. Понятие об обратной связи, её разновидности. Передаточная функция системы автоматического управления с положительной и отрицательной обратными связями.

1.2.9 Параметры и характеристики систем автоматического управления

Понятие о характеристиках систем автоматического управления, их разновидности. Статические характеристики систем автоматического управления. Динамические характеристики систем автоматического управления. Характеристика типовых воздействий для исследования систем автоматического управления. Понятие о параметрах систем автоматического управления и их характеристика.

1.2.10 Классификация систем автоматического управления

Классификация по алгоритму функционирования. Классификация по свойствам в установившемся режиме. Классификация по характеру изменения внутренних величин. Классификация по способам настройки. Классификация по математическим свойствам. Некоторые другие классификации.

1.2.11 Средства автоматизации для построения систем автоматического управления

Разработка систем автоматического управления на основании микроконтроллеров, персональных компьютеров, специализированных средств автоматизации. Характеристика каждого из способов построения. Основные преимущества и недостатки.

1.2.12 Внутренняя организация промышленных микроконтроллерных систем автоматического управления

Архитектура и структура микроконтроллерных систем. Конструктивные стандарты (форм-факторы), применяемые для построения микроконтроллерных систем. Классификация и использование программного обеспечения для функционирования микроконтроллерных систем.

1.2.13 Развитие микроконтроллерных систем автоматического управления в промышленности

Внутреннее устройство и особенности использования промышленных компьютеров, встраиваемых компьютеров и промышленных контроллеров для решения задач управления промышленными процессами.

1.2.14 Возникновение и развитие промышленных контроллеров

Цели и результаты использования микроконтроллерных систем. Специфика использования микроконтроллерных систем в промышленности. Влияние негативных факторов промышленности на функционирование микроконтроллерных систем. Характеристика промышленных контроллеров.

1.2.15 Классификация промышленных контроллеров

Классификация промышленных контроллеров по функциональному назначению. Классификация по количеству входов/выходов. Классификация по пространственному размещению структурных элементов. Классификация по конструктивному исполнению. Некоторые другие классификации.

1.2.16 Эволюция развития средств автоматизированных систем управления технологическими процессами

Исторические этапы развития средств автоматизированных систем управления технологическим процессом. Характеристика централизованных и распределённых систем автоматизации. Характеристика программно-технических комплексов на базе промышленных контроллеров.

1.3 Распределение объема самостоятельной работы

Распределение объема самостоятельной работы указано в таблице 1.1.

Таблица 1.1 – Распределение объема самостоятельной работы по видам работ

Вид работы	Объем, часов
1 Изучение терминологии и основных определений дисциплины	4
2 Подготовка к лабораторным работам	6
3 Изучение архитектуры и структуры микроконтроллеров с ядром ARM 7	10
4 Ознакомление с техническими возможностями лабораторного оборудования дисциплины	3
5. Подготовка к сдаче модульных контролей	12
6 Выполнение расчетно-графической работы	7
Всего за курс	42

1.4 Вопросы для сдачи зачёта по дисциплине

1 Дайте определения понятий: микроконтроллер, система, управление, система управления, реальное время, автоматизированное и автоматическое управление, архитектура и структура микроконтроллерной системы.

2 Понятие об интерфейсе МКС. Классификации интерфейсов по различным признакам (с обязательным указанием примеров).

3 Понятие об интерфейсах МКС, виды их совместимости.

4 Понятие об интерфейсах МКС. Основные характеристики SPI-интерфейса. Внутренняя организация SPI (сигнальные линии, регистры). Схема простейшего подключения устройств по SPI.

5 Понятие об интерфейсах МКС. Основные характеристики SPI-интерфейса. Протокол обмена данными. Схема простейшего подключения устройств по SPI.

6 Понятие об интерфейсах МКС. Основные характеристики SPI-интерфейса. Конфигурация режимов обмена данными. Конфигурация выводов модуля интерфейса SPI.

7 Понятие об интерфейсах МКС. Основные характеристики SPI-интерфейса. Временные диаграммы обмена данными в каждом из существующих режимов.

8 Понятие об интерфейсах МКС. Основные характеристики SPI-интерфейса. Способы подключения нескольких ведомых устройств, их преимущества и недостатки.

9 Общая характеристика синхронной шины I²C. Схема подключения устройств к шине.

10 Общая характеристика синхронной шины I²C. Предназначение и генерация старт- и стоп-битов. Выборка данных.

11 Протокол обмена данными по шине I²C.

12 Общая характеристика шины 1-Wire.

13 Генерация и предназначение сигналов шины 1-Wire.

14 Команды ПЗУ и функциональные команды шины 1-Wire.

15 Понятие об интерфейсах МКС. Типичная последовательность сеанса обмена данными между устройствами по шине 1-Wire. Команды ПЗУ для шины

1-Wire.

16 Дайте определения понятий: система, управление, регулирование, основная задача управления, канал воздействия, выходная величина, объект управления, элементарное звено, передаточная функция, обратная связь, характеристика, параметр, коэффициент статизма, адаптивность, устойчивость.

17 Дайте определения всех типов воздействий, которые существуют в системах управления.

18 Назовите фундаментальные принципы управления. Дайте характеристику принципа управления по возмущению.

19 Назовите фундаментальные принципы управления. Дайте характеристику принципа управления по задающему воздействию. Принцип комбинированного управления.

20 Назовите фундаментальные принципы управления. Дайте характеристику принципа управления по отклонению.

21 Понятие об элементарных звеньях системы управления. Виды соединения элементарных звеньев.

22 Характеристика передаточной функции системы управления.

23 Обратная связь в системах управления.

24 Временные характеристики САУ. Типовые воздействия для исследования САУ.

25 Частотные и логарифмические частотные характеристики САУ.

26 Дайте характеристику параметров САУ.

27 Классификации САУ: по алгоритму функционирования; по математическим свойствам.

28 Классификации САУ: по свойствам в установившемся режиме; в зависимости от способов настройки.

29 Дайте определения понятий: микроконтроллер, система, управление, время реакции на событие, промышленный контроллер, устройство связи с объектом.

30 Преимущества создания систем управления на базе ПК.

31 Недостатки создания систем управления на базе ПК.

32 Характерные особенности промышленных компьютеров.

33 Характерные особенности встраиваемых компьютеров.

34 Структура промышленного контроллера.

35 Классификация входов/выходов промышленного контроллера по способу функционирования.

36 Программное обеспечение промышленных контроллеров, программный рабочий цикл, время реакции на событие.

37 Краткая характеристика направлений развития промышленных контроллеров, характерные особенности ПЛК.

38 Классификации промышленных контроллеров по различным признакам.

39 Классификация промышленных контроллеров по признаку "мощность", по пространственному расположению компонентов.

40 Классификация промышленных контроллеров по признаку "мощность", по внутренней архитектуре.

41 Классификация промышленных контроллеров по функциональному назначению, характеристика классов.

42 Общая характеристика этапов развития средств АСУ ТП.

43 Структура централизованной системы автоматизации.

44 Структура распределённой системы автоматизации.

45 Программно-технические комплексы на базе промышленных контроллеров.

46 Характеристики программно-технических комплексов.

2 ХАРАКТЕРНЫЕ ОСОБЕННОСТИ МИКРОКОНТРОЛЛЕРОВ ARM7

2.1 Типовые характеристики микроконтроллеров с ядром ARM7

Микроконтроллеры семейства LPC2000 имеют следующие типовые характеристики:

- высокопроизводительное 32/16-битное ядро ARM7TDMI-S;
- встроенный тактовый генератор 1...30 МГц со схемой ФАПЧ;
- 8/16/32/64 Кбайт памяти типа SRAM (статическое ОЗУ);
- 64/128/256/512 Кбайт памяти типа Flash со 128-битной шиной и обращением без цикла ожидания (ресурс не менее 10000 циклов стирания/записи в рабочем температурном диапазоне);
- стандартный интерфейс отладки JTAG;
- ISP (In-System Programming) внутрисистемное программирование Flash-памяти;
- IAP (In-Application Programming) режим побайтового программирования в составе целевой программы (пользовательского приложения);
- работа с командами длиной 32 бита (в стандартном режиме ARM) и 16 бит (в режиме THUMB);
- многоканальный модуль ШИМ;
- интерфейсы UART, скоростной I²C (до 400 Кбит/с), SPI, USB;
- интерфейсы CAN с приемными фильтрами;
- два 32-битных таймера с каналами сравнения и захвата;
- микросхема часов реального времени и сторожевой таймер WDT;
- многоканальный 10-битный АЦП;
- 10-битный модуль ЦАП;
- встроенный отладочный интерфейс ICE, позволяющий задавать точки останова;
- раздельное питание ядра и встроенной периферии (1.65...1.95 В и 3.0...3.6 В соответственно);
- возможность отключения неиспользуемой периферии;
- два режима экономии потребляемой мощности;
- индивидуальное управление питанием встроенной периферии;
- линии ввода/вывода общего назначения GPIO допускают подачу на них напряжения ВЫСОКОГО уровня 5 В (при этом номинальная величина напряжения ВЫСОКОГО уровня на этих линиях составляет 3.0 ... 3.6 В).

2.2 Особенности архитектурного построения ядра ARM7

Все микроконтроллеры семейства LPC2000, которые производятся компанией NXP (дочерним предприятием компании Philips), имеют процессорное ядро ARM7TDMI-S с поддержкой эмуляции, шину ARM7 контроллера встроенной памяти, шину AMBA AHB (Advanced Microcontroller Bus Architecture – Advanced High-performance Bus) контроллера прерываний и шину VPB (VLSI Peripheral Bus) встроенных периферийных устройств.

Под периферийные устройства, подключенные к шине АНВ, в общем адресном пространстве программной модели ARM размером 4 Гбайт отведено адресное пространство размером 2 Мбайт. Каждому периферийному устройству АНВ выделено адресное пространство размером 16 Кбайт в пределах адресного пространства периферии АНВ.

Под периферийные устройства, подключенные к шине VPB, также отведено адресное пространство размером 2 Мбайт, которое начинается с адреса, находящегося на отметке 3.5 Гбайт. Каждому периферийному устройству VPB также выделено адресное пространство размером 16 Кбайт в пределах адресного пространства периферии VPB.

Сопряжение периферийных устройств АНВ с периферийными устройствами VPB осуществляется посредством интерфейса шины VPB к шине АНВ. Для управления связью периферийных устройств с выводами микроконтроллера предназначен блок коммутации выводов (PIN CONNECT BLOCK), который должен быть корректно сконфигурирован пользовательским программным обеспечением и должен соответствовать определенным прикладным требованиям для использования периферийных функций и выводов.

2.3 Микроконтроллерное ядро ARM7TDMI-S

Буквенные обозначения “TDMI-S” в названии ядра ARM7TDMI-S, согласно спецификациям ARM, имеют следующую расшифровку:

T – указывает на наличие в составе ядра, помимо базовой архитектуры ARM, дополнительной архитектуры THUMB, что позволяет пользователю выбирать для своего приложения одну из двух систем команд: ARM или THUMB;

D – ядро содержит в своем составе дополнительные отладочные модули;

M – ядро содержит в своем составе дополнительный аппаратный умножитель, позволяющий выполнять команды умножения так называемой “длинной формы” (с 64-битным результатом);

I – ядро содержит в своем составе встроенную логику отладки (ICE);

S – ядро является полностью синтезируемым, т.е. оно может быть описано на языках описания цифровых устройств VHDL или Verilog.

ARM7TDMI-S – универсальный 32-битный микропроцессор, который потребляет сравнительно небольшую мощность и при этом обеспечивает высокую эффективность. Архитектура ARM основана на использовании сокращенного набора команд процессора (RISC – Reduced Instruction Set Computer) и реализует намного более простой механизм декодирования команд, нежели у микроконтроллеров с архитектурой, основанной на использовании полного набора команд процессора (CISC – Complex Instruction Set Computer). Эта простота обеспечивает высокую производительность при выполнении каждой команды, а также малое время реакции на прерывания в реальном времени. В состав ядра включены отладочные интерфейсы JTAG и ETM, а также встроенная логическая схема отладки ICE. Упрощенная архитектура микроконтроллерного ядра ARM7TDMI-S приведена на рисунке 2.1.

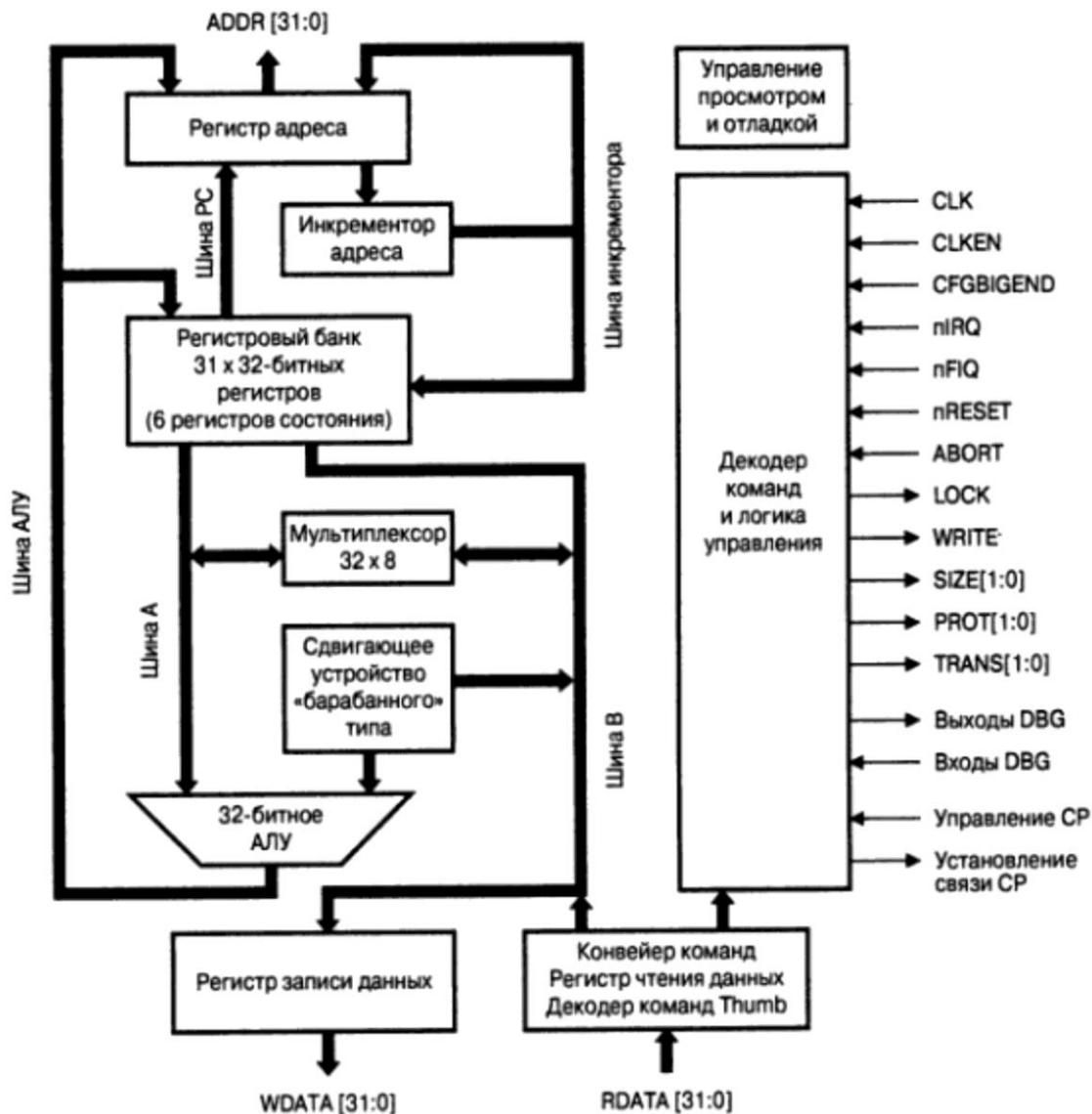


Рисунок 2.1 – Упрощенная архитектура ядра ARM7TDMI-S

Для того, чтобы обеспечить непрерывность выполнения программ и обращения к памяти, в ядре ARM7TDMI-S, как и во многих других ядрах, реализован конвейер команд. Трехступенчатый конвейер ARM7TDMI-S работает следующим образом: когда первая инструкция выполняется, следующая за ней в программе (вторая) инструкция декодируется, а третья считывается из памяти. Наличие конвейера позволяет последовательно выполнять поступающие команды, затрачивая на обработку каждой из них по одному такту.

Ядро ARM7TDMI-S, помимо стандартного набора команд, называемого ARM, также использует уникальный набор команд, известный как THUMB. Этот набор команд в большей степени, нежели набор ARM, подходит для крупномасштабных приложений, реализация которых предполагает ограничение объема используемой памяти или размера программного кода. Количество команд, входящих в набор THUMB, значительно меньше, по сравнению с системой команд набора ARM. Поэтому, процессорное ядро ARM7TDMI-S имеет две системы команд:

- стандартная 32-битная система команд ARM;

– 16-битная система команд THUMB.

При использовании набора команд THUMB с 16-битной длиной команды, в ряде случаев удается достичь двойной плотности относительно стандартного кода ARM; при этом сохраняется большинство функций кода ARM. Фактически код THUMB оперирует тем же самым набором 32-битных регистров, что и код ARM. По некоторым оценкам, код THUMB “ужимается” до 65% размера функционально эквивалентного ему кода ARM, а также обеспечивает до 160% производительности по сравнению с 16-битной процессорной системой, архитектура которой отлична от ARM.

Процессорное ядро ARM7TDMI-S использует следующие типы данных:

- слово (32 бита);
- полуслово (16 бит);
- байт (8 бит).

Процессорное ядро ARM7TDMI-S рассматривает память как линейный массив байтов, пронумерованных с нуля в возрастающем порядке, т.е.:

- байты 0...3 составляют первое слово памяти;
- байты 4...7 составляют второе слово памяти;
- байты 8... 11 составляют третье слово памяти и т. д.

Ядро ARM7TDMI-S использует следующие форматы хранения данных (программ) в памяти:

- big-endian;
- little-endian.

В соответствии с форматом big-endian, процессор хранит старший байт слова данных (программ) в байте памяти с наименьшим номером, а младший байт слова – в байте памяти с наибольшим номером.

В соответствии с форматом little-endian, процессор хранит младший байт слова данных (программ) в байте памяти с наименьшим номером, а старший байт слова – в байте памяти с наибольшим номером.

В исходном коде пользовательской программы с помощью специальных директив можно указать, какой из форматов хранения данных использовать.

2.4 Режимы работы и регистровые модели микроконтроллерного ядра ARM7TDMI-S в состоянии ARM

В состоянии ARM микроконтроллерное ядро может функционировать в одном из следующих режимов:

User – выполнение программ пользователя.

Supervisor – работа под управлением операционной системы, которая оперирует с данными, недоступными программам пользователя.

System – режим выполнения системных программ, при котором операционная система работает с данными пользователя.

IRQ – режим обработки прерываний, в который ядро переходит при поступлении на вход IRQ запроса прерывания низшего уровня.

FIQ (Fast IRQ) – режим быстрой реакции на прерывания, в который ядро переходит при поступлении на вход FIQ запроса прерывания высшего уровня.

Abort – режим, в который ядро переходит при ошибке обращения к памяти (обращение по несуществующему адресу, попытка записи в ПЗУ и т.п.). Такие ошибки фиксируются контроллером прерываний, который и выдает ядру запрос на переход в режим Abort.

Undefined – режим, в который ядро переходит при выборке неправильного кода команды.

Все режимы функционирования, кроме режима User, называются привилегированными. Каждому из них соответствует определенная регистровая модель, содержащая набор 32-битных регистров, доступных в этом режиме. Все возможные регистровые модели для состояний ядра ARM приведены ниже в таблице 2.1. В каждом режиме процессору доступны 16 регистров общего назначения R0...R15 и регистр состояния CPSR. Все регистры общего назначения имеют определенные стандартом ARM дополнительные имена, которые указаны в таблице в скобках. При вызове подпрограмм и организации передачи данных между ними, этот стандарт рекомендует следующее использование регистров общего назначения:

- регистры a1 ... a4 используются для передачи параметров подпрограммам;
- регистры v1 ... v8 используются для хранения локальных переменных;
- регистр IP служит для хранения промежуточных данных между вызовами процедур.

Регистры, имена которых выделены в таблице жирным шрифтом, являются банкированными. Это значит, что они используются только в том режиме, на который указывают их суффиксы (**_fiq**, **_svc**, **_abt**, **__irq**, **__und**) вместо соответствующих им регистров без суффиксов. При переходе в указанный режим из других режимов, текущее содержимое регистров без суффиксов автоматически переписывается в соответствующие банкированные регистры, а при возврате в прежний режим вновь используются его собственные регистры. Таким образом, суммарное количество физических 32-битных регистров общего назначения, доступное процессору во всех режимах состояния ARM, равно 31, а суммарное количество доступных процессору физических регистров состояния равно 6 (регистру CPSR соответствуют банкированные регистры с обобщенным форматом имени **SPSR_xxx**).

В базовом режиме User, в котором ядро находится большую часть рабочего времени, используются 32-битные регистры R0...R15 и регистр состояния CPSR.

Регистр R15 выполняет функции программного счетчика (PC). В состоянии ARM биты [1] и [0] R15 всегда равны 0, т. е. адреса команд всегда кратны четырем, а в состоянии THUMB бит [0] R15 всегда равен 0, т. е. адреса команд всегда четные.

Регистр R14 (LR) используется при вызовах подпрограмм как регистр связи, в котором сохраняется текущее содержимое PC для возврата в основную программу.

Регистр R13 (SP) рекомендуется использовать в качестве указателя на вершину стека.

Таблица 2.1 – Регистровые модели ядра в различных режимах состояния ARM

User, System	FIQ	Supervisor	Abort	IRQ	Undefined
R0 (al)	R0	R0	R0	R0	R0
R1 (a2)	R1	R1	R1	R1	R1
R2 (a3)	R2	R2	R2	R2	R2
R3 (a4)	R3	R3	R3	R3	R3
R4 (v1)	R4	R4	R4	R4	R4
R5 (v2)	R5	R5	R5	R5	R5
R6 (v3)	R6	R6	R6	R6	R6
R7 (v4)	R7	R7	R7	R7	R7
R8 (v5)	R8_fiq	R8	R8	R8	R8
R9 (v6)	R9_fiq	R9	R9	R9	R9
R10 (v7)	R10_fiq	R10	R10	R10	R10
R11 (v8)	R11_fiq	R11	R11	R11	R11
R12 (IP)	R12_fiq	R12	R12	R12	R12
R13 (SP)	R13_fiq	R13_svc	R13_abt	R13_irq	R13_und
R14 (LR)	R14_fiq	R14_svc	R14_abt	R14_irq	R14_und
R15 (PC)	R15	R15	R15	R15	R15
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

Битовая структура регистра состояния CPSR изображена на рисунке 2.2. Как видно из рисунка, он содержит следующие биты:

M4 ... M0 – определяют режимы работы ядра, как указано в таблице 2.2;

T – задает набор исполняемых команд: ARM (при T = 0) или THUMB (при T = 1);

I, F – маскируют (запрещают) обработку прерываний IRQ и FIQ соответственно;

N, Z, C, V – являются признаками: отрицательного результата (N), нуля (Z), переноса (C) и переполнения (V); их значения могут устанавливаться в соответствии с результатом выполнения очередной операции.

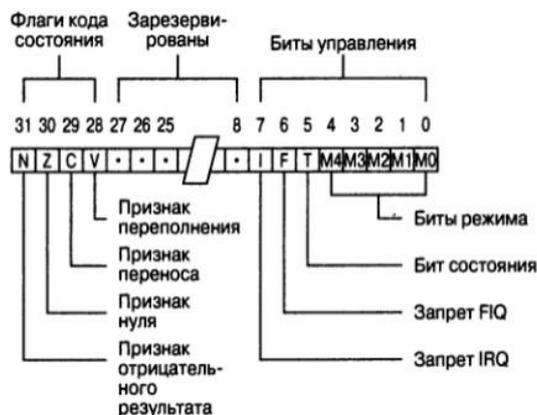


Рисунок 2.2 – Битовая структура регистра состояния CPSR

Регистр CPSR доступен для записи во всех режимах, кроме режима User.

После включения питания микроконтроллерное ядро начинает работу в режиме Supervisor. Переход в базовый режим User реализуется путем записи в регистр CPSR содержимого, в котором биты M4...M0 = 10000. Обратное переключение из режима User в режим Supervisor может быть произведено только при поступлении команды программного прерывания, мнемоническое имя которой – SWI. Таким образом, обеспечивается доступ пользователя к ресурсам операционной системы. Чтобы из режима Supervisor получить доступ ко всем регистрам режима User, следует выполнить переход в режим System. Для этого в регистр CPSR записывается значение битов M4...M0 = 11111. Переход в режимы IRQ, FIQ, Abort, Undefined происходит автоматически при поступлении соответствующих запросов прерывания или при наступлении соответствующих событий.

Таблица 2.2 – Конфигурирование режимов работы микроконтроллерного ядра

Комбинация битов M4 ... M0	Режим
10000	User
10001	FIQ
10010	IRQ
10011	Supervisor
10111	Abort
11011	Undefined
11111	System

При переходе в другой режим текущее содержимое регистра CPSR (слово состояния) автоматически переписывается в регистр SPSR_xxx, соответствующий новому режиму, а также автоматически сохраняется в теневом регистре. При возврате в исходный режим содержимое CPSR автоматически восстанавливается из сохраненной копии. Кроме того, для каждого режима имеются отдельные банкированные регистры R13_xxx (SP) и R14_xxx (LR), поэтому при переключении режимов сохраняются текущие значения указателя стека и регистра связи. Таким образом, при возврате к исходному режиму обеспечивается корректное обращение к вершине стека и возможность возврата в основную программу, если перед сменой режима производился вызов подпрограммы.

При переходе в режим FIQ (после поступления запроса прерывания на входе FIQ), помимо вышеперечисленных регистров, сохраняется текущее содержимое регистров R8...R12, вместо которых в данном случае используются банкированные регистры R8_flq ... R12_flq. Следовательно, при обработке быстрого прерывания нет необходимости сохранять содержимое этих регистров в стеке, что, обеспечивает более быструю передачу управления программному блоку обработчика прерываний FIQ и возврат из него.

2.5 Режимы работы и регистровые модели микроконтроллерного ядра ARM7TDMI-S в состоянии THUMB

Переход ядра из состояния ARM в состояние THUMB и обратно производится с помощью команды BX. При выполнении программы, встроенный декодер преобразует команды, записанные в системе THUMB, в команды ARM. Вследствие компактности используемых форматов, система команд THUMB имеет ряд особенностей и ограничений по сравнению с системой ARM. Набор регистров в состоянии THUMB сокращен по сравнению с набором регистров, доступных в состоянии ARM. Все возможные регистровые модели для состояния THUMB приведены в таблице 2.3.

Таблица 2.3 – Регистровые модели микроконтроллерного ядра в различных режимах состояния THUMB

User, System	FIQ	Supervisor	Abort	IRQ	Undefined
R0 (a1)	R0	R0	R0	R0	R0
R1 (a2)	R1	R1	R1	R1	R1
R2 (a3)	R2	R2	R2	R2	R2
R3 (a4)	R3	R3	R3	R3	R3
R4 (v1)	R4	R4	R4	R4	R4
R5 (v2)	R5	R5	R5	R5	R5
R6 (v3)	R6	R6	R6	R6	R6
R7 (v4)	R7	R7	R7	R7	R7
SP	SP_fiq	SP_svc	SP_abt	SP_irq	SP_und
LR	LR_fiq	LR_svc	LR_abt	LR_irq	LR_und
PC	PC	PC	PC	PC	PC
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

В коде операции большинства команд THUMB под номер регистра отведено только 3 бита, поэтому прямое обращение возможно только к восьми регистрам: R0 ... R7. Регистры R8 ... R12 доступны только через специальные команды загрузки. Регистры SP и LR выполняют фиксированные функции указателя вершины стека и регистра связи, соответственно. В состоянии THUMB имеются специальные команды, ориентированные на работу с этими регистрами: загрузка в стек проходит только через регистр SP, адрес возврата из подпрограммы сохраняется только в регистре LR.

2.6 Время реакции на прерывание

Время реакции на прерывание в микроконтроллерной системе критично, как правило, только для быстрых прерываний FIQ. Значение этого времени лежит в некотором возможном диапазоне. Когда прерывания FIQ разрешены, максимальное время реакции на него для самого “худшего” случая состоит из:

T_{syncmax} – наиболее длительное время запроса, которое требуется для реакции синхронизатора, оно составляет 2 машинных цикла (т.е. 2 такта кварцевого резонатора);

T_{ldm} – время, которое требуется для завершения выполнения самой длинной команды – LDM, которая загружает все регистры микроконтроллера, включая регистр PC, T_{ldm} составляет 20 машинных циклов в системе с нулевым временем ожидания;

T_{exc} – время выполнения входа в обработчик исключений Data Abort, оно составляет 3 машинных цикла;

T_{fiq} – время для выполнения входа в обработчик прерывания FIQ, оно составляет 2 машинных цикла.

Таким образом, полное время ожидания для самого “худшего” случая составляет 27 машинных циклов, что составляет 0.675 мкс в системе с тактовой частотой кварцевого резонатора в 40 МГц.

При вычислении максимального времени реакции на прерывание IRQ необходимо учитывать тот факт, что обработка прерываний FIQ, имеющих приоритет гораздо выше, чем у IRQ, может задержать процесс входа в обработчик прерывания IRQ.

Минимальное время реакции на прерывание для FIQ или IRQ – это кратчайшее время запроса, которое может потребоваться на реакцию синхронизатора T_{syncmin} , сложенное со временем T_{fiq} , что в сумме составляет 4 машинных цикла.

2.7 Организация системной памяти микроконтроллеров ARM7

Встроенная память микроконтроллеров семейства LPC2000 построена по технологии Flash и разделена на несколько областей. На рисунке 2.3 изображена карта всей системной памяти микроконтроллеров LPC2141/42/44/46/48, доступной пользовательской программе после сброса.

На рисунке 2.4 изображена карта периферийной памяти микроконтроллеров семейства LPC2000. К периферийной памяти относятся периферийные области АНВ и VPВ, размером по 2 Мбайт. Каждая из них разделена на 128 областей периферийных устройств. Таким образом, каждому периферийному устройству отведена область памяти размером 16 Кбайт, что упрощает дешифрацию адреса для соответствующего периферийного устройства.

4.0Г	АНВ-периферия	0xFFFF FFFF
3.75Г	VPB-периферия	0xF000 0000
3.5Г		0xE000 0000
3.0Г	Резервное пространство адресов	0xC000 0000
2.0Г	Загрузочный блок (переотображен на встроенную Flash-память)	0x8000 0000
	Резервное пространство адресов	0x7FFF E000 0x7FFF CFFF
	8К встроенной RAM USB DMA (LPC2146/2148)	0x4000 2000 0x4000 1FFF
	Резервное пространство адресов	0x7FD0 0000 0x7FCF FFFF
	32К встроенной SRAM (LPC2146/2148)	0x4000 8000 0x4000 7FFF
	16К встроенной SRAM (LPC2142/2144)	0x4000 4000 0x4000 3FFF
	8К встроенной SRAM (LPC2141)	0x4000 2000 0x4000 1FFF
1.0Г	Резервное пространство адресов	0x4000 0000 0x3FFF FFFF
	512К встроенной энергонезависимой памяти (LPC2148)	0x0008 0000 0x0007 FFFF
	256К встроенной энергонезависимой памяти (LPC2146)	0x0004 0000 0x0003 FFFF
	128К встроенной энергонезависимой памяти (LPC2144)	0x0002 0000 0x0001 FFFF
	64К встроенной энергонезависимой памяти (LPC2142)	0x0001 0000 0x0000 FFFF
	32К встроенной энергонезависимой памяти (LPC2141)	0x0000 8000 0x0000 7FFF
0.0Г		0x0000 0000

Рисунок 2.3 – Карта всей системной памяти микроконтроллеров LPC2141/42/44/46/48

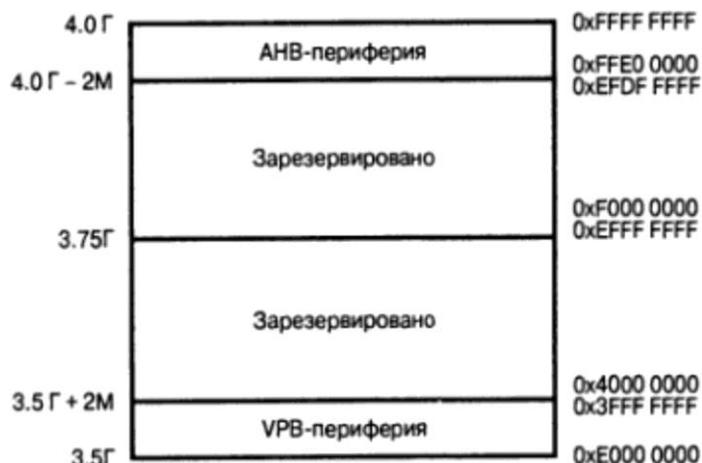


Рисунок 2.4 – Карта периферийной памяти

Карта периферийной памяти области АНВ микроконтроллеров LPC2000 изображена на рисунке 2.5.

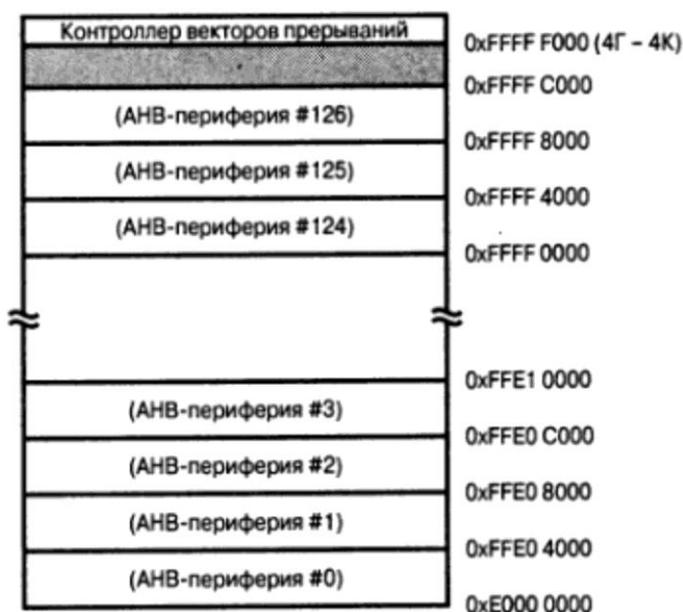


Рисунок 2.5 – Карта периферийной памяти области АНВ устройств

Базовые адреса VPB-устройств микроконтроллеров LPC2141/42/44/46/48 приведены в таблице 2.4.

Концепция организации встроенной памяти микроконтроллеров семейства LPC2000 состоит в следующем: каждая область памяти микроконтроллера имеет свое “естественное”, раз и навсегда заданное местоположение в карте памяти. Это фиксированное местоположение представляет собой адресный интервал, для которого пишется программный код, “привязанный” к выбранной области. Объем каждой такой области памяти всегда постоянен. Таким образом, пользователю не нужно писать программный код так, чтобы он мог работать сразу в нескольких различных областях памяти.

Таблица 2.4 – Базовые адреса VPB-устройств для LPC2148

VPB-периферия	Базовый адрес	Периферийное устройство
0	0xE000 0000	Сторожевой таймер
1	0xE000 4000	Таймер 0
2	0xE000 8000	Таймер 1
3	0xE000 C000	UART0
4	0xE001 0000	UART1
5	0xE001 4000	PWM
6	0xE001 8000	Не используется
7	0xE001 C000	I ² C0
8	0xE002 0000	SPI0
9	0xE002 4000	RTC
10	0xE002 8000	GPIO
11	0xE002 C000	Блок коммутации выводов
12	0xE003 0000	Не используется
13	0xE003 4000	ADC0
14...22	0xE003 8000 - 0xE005 8000	Не используется
23	0xE005 C000	I ² C1
24	0xE006 0000	ADC1
25	0xE006 4000	Не используется
26	0xE006 8000	SSP
27	0xE006 C000	DAC
28...35	0xE007 0000 - 0xE008 C000	Не используется
36	0xE009 0000	USB
37...126	0xE009 4000 - 0xE01F 8000	Не используется
127	0xE01F C000	Блок управления системой

2.8 Структура микроконтроллера ARM7

В лабораторном курсе дисциплины “Микроконтроллерные системы автоматизации и управления” используется учебно-отладочный стенд от компании NXP, в состав которого входит микроконтроллер LPC2148, поэтому ниже на рисунке 2.6 представлена детальная структура именно этого микроконтроллера.

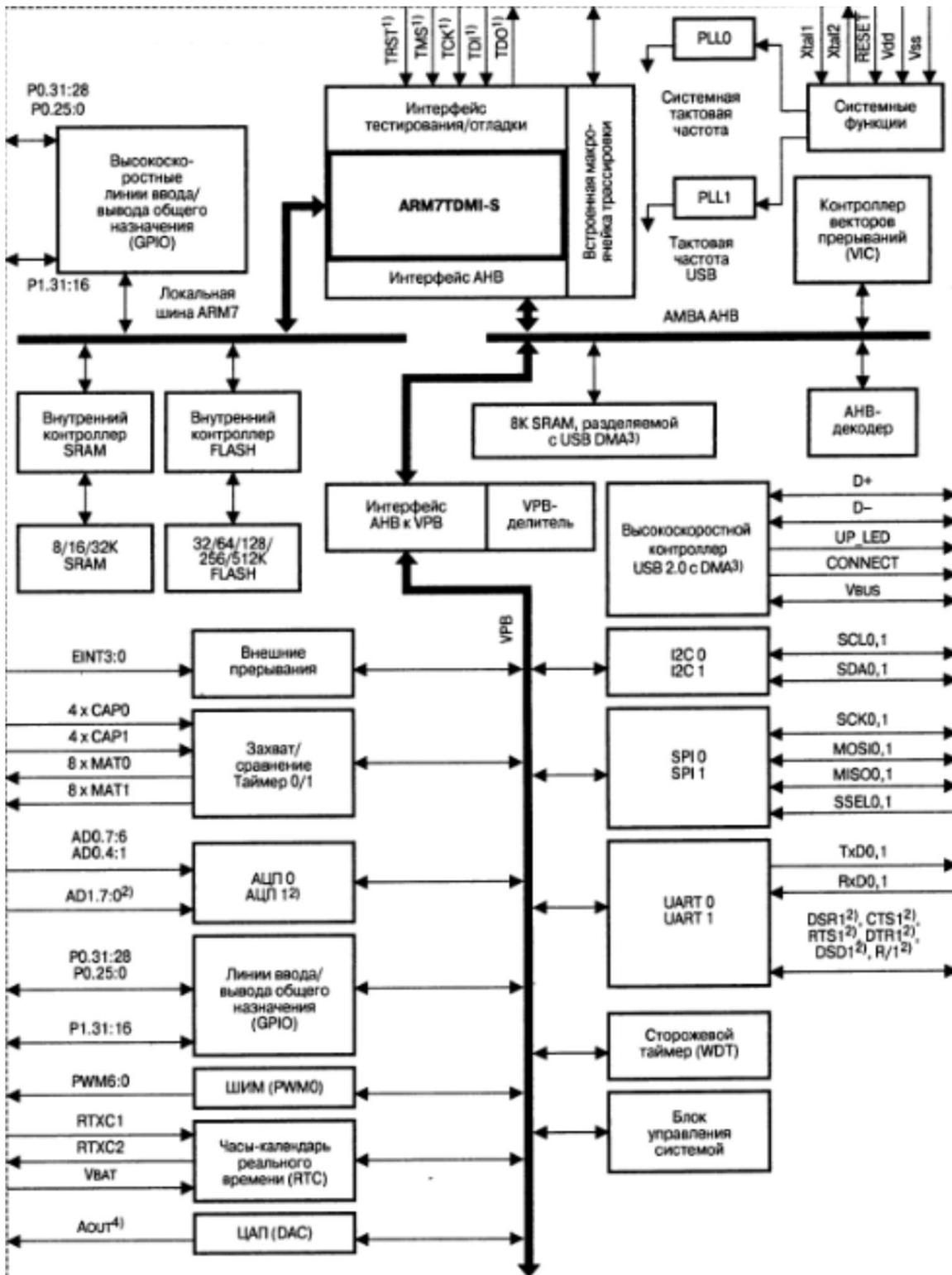


Рисунок 2.6 – Структура микроконтроллера LPC2148

3 СИСТЕМНЫЕ УСТРОЙСТВА МИКРОКОНТРОЛЛЕРОВ СЕМЕЙСТВА ARM7

3.1 Блок управления системой

Механизм управления всей системой для микроконтроллеров семейства LPC2000 включает в себя несколько аппаратных узлов и регистров управления множеством независимых друг от друга функций, связанных с определенными периферийными устройствами. Блок управления микроконтроллером включает в себя:

- генератор, стабилизированный кварцевым резонатором;
- входы внешних прерываний;
- узел управления памятью;
- узел ФАПЧ(PLL);
- узел управления потребляемой мощностью;
- узел сброса;
- VPB-делитель;
- таймер “пробуждения”.

Каждая функция (узел) имеет свои собственные регистры, если они требуются для контроля и управления, а неиспользуемые в них биты определены как зарезервированные, с учетом последующего их использования в перспективных устройствах. Различные функции никогда не используют совместно регистры с одинаковыми адресами.

В таблице 3.1 приведено описание тех выводов микроконтроллеров семейства LPC2000, которые связаны с блоком управления системой.

Таблица 3.1 – Описание выводов, связанных с блоком управления системой

Название	Направление	Описание
EINT0	Вход	Вход внешнего прерывания 0 – универсальный вход прерывания с низким активным уровнем. Вывод может использоваться, чтобы “пробудить” микроконтроллер из режимов пониженного энергопотребления Idle или Power Down.
EINT1	Вход	Вход внешнего прерывания 1 – смотри EINT0. Низкий уровень на этом выводе сразу после сброса рассматривается как запрос внешних аппаратных средств для запуска обработчика внутрисхемного программирования (ISP).
EINT2	Вход	Вход внешнего прерывания 2 – смотри EINT0.
EINT3	Вход	Вход внешнего прерывания 3 – смотри EINT0.
XTAL1	Вход	Вход кварцевого генератора – вход генератора и внутренних цепей генерации синхросигнала
XTAL2	Выход	Выход кварцевого генератора – выход усилителя генератора
RESET	Вход	Вход внешнего сброса. Низкий уровень на этом выводе сбрасывает микроконтроллер

3.1.1 Генератор, стабилизированный кварцевым резонатором

В микроконтроллерах семейства LPC2000 может использоваться внешний сигнал тактирования прямоугольной формы (“меандр”) с частотой 1 ... 50 МГц. Его следует подавать на вход XTAL1. Встроенная схема генератора микроконтроллера поддерживает работу с внешними кварцевыми резонаторами лишь в частотном диапазоне 1 ... 30 МГц. Если в микроконтроллере используется встроенная система ФАПЧ или загрузчик, то входная частота тактового сигнала ограничена диапазоном 10 ... 25 МГц.

Частоты F_{osc} , формируемая тактовым генератором, и clk – частота тактирования микроконтроллерного ядра совпадают лишь в том случае, если встроенная система ФАПЧ не используется (разомкнута).

Встроенный генератор в микроконтроллерах семейства LPC2000 может работать в одном из двух режимов: ведомом и режиме автоколебаний (генерации). В ведомом режиме входной внешний тактовый сигнал должен быть подан на вход генератора (вход XTAL1) через конденсатор ёмкостью 100 пФ и амплитудой не менее 200 мВ. Вывод XTAL2 в этом случае можно оставить неподключенным. Если выбран ведомый режим, частотный диапазон сигнала F_{osc} типа “меандр” может составлять 1 ... 50 МГц.

Для использования режима автоколебаний необходимо наличие внешних компонентов. Сопротивление обратной связи, необходимое для генерации, уже встроено в микроконтроллер, а вот ёмкости C_{X1} и C_{X2} должны быть соответствующим образом подключены, для обеспечения генерации на основной частоте используемого кварцевого резонатора (основная частота резонатора определяется величинами индуктивности, ёмкости и объёмного сопротивления эквивалентной схемы кварцевого резонатора). Значения частоты параллельного резонанса, ёмкостей, индуктивности и сопротивления кварцевого резонатора обычно указываются производителем. Выбранный режим автоколебаний ограничивает значение F_{osc} частотным диапазоном в 1 ... 30 МГц.

3.1.2 Входы внешних прерываний

Микроконтроллеры семейства LPC2000 имеют 4 входа внешних прерываний, выбор которых осуществляется путем настройки соответствующих контактов ввода/вывода микроконтроллера. Все контакты ввода/вывода имеют дополнительные функции, выбор которых производится с помощью мультиплекторов блока коммутации выводов (pin select block). Так, входы внешних прерываний физически “делят” одни и те же контакты с линиями порта ввода/вывода общего назначения, входами ЦАП, АЦП и т.д.

Входы внешних прерываний также могут быть произвольным образом задействованы, чтобы “пробудить” процессор из режима пониженного энергопотребления Power Down.

С функцией обработки внешних прерываний связаны четыре регистра, указанные в таблице 3.2. Регистр EXTINT содержит флаги прерываний. EXTWAKE – это регистр разрешения пробуждения от внешних прерываний. В зависимости от значения записанных в него битов, внешним прерываниям раз-

решается или запрещается “пробуждать” микроконтроллер (т.е. переводить его из режима Power Down в активное состояние). Регистры EXTMODE и EXTPOLAR задают активные уровень и передний/задний фронт сигнала, вызывающего внешнее прерывание.

Таблица 3.2 – Описание регистров внешних прерываний

Адрес	Название	Описание	Доступ
0xE01FC140	EXTINT	Регистр флагов внешних прерываний. Содержит флаги прерываний EINT0, EINT1, EINT2, EINT3	R/W
0xE01FC144	EXTWAKE	Регистр “пробуждения” от внешних прерываний. Содержит 3 бита, управляющие возможностью “пробуждения” процессора от внешних прерываний из режима Power Down	R/W
0xE01FC148	EXTMODE	Регистр режима внешних прерываний. Задаёт, появление фронта или уровня на входах EINT0 ... EINT3 будет инициироваться соответствующее внешнее прерывание	R/W
0xE01FC14C	EXTPOLAR	Регистр управления полярностью внешних прерываний. Задаёт, какой перепад или уровень импульса будет инициировать внешнее прерывание	R/W

Битовая структура регистра флагов внешних прерываний EXTINT приведена ниже, в таблице 3.3.

Когда выбрана функция обработки внешнего прерывания с вывода микроконтроллера, уровень или фронт/срез входного сигнала на этом выводе (в зависимости от состояния битов в регистрах EXTPOLAR и EXTMODE) вызывает установку соответствующего флага прерывания в регистре EXTINT. Далее запрос на прерывание поступает в контроллер векторов прерываний VIC (Vectored Interrupt Controller). Если внешние прерывания разрешены, то VIC инициирует запуск соответствующего обработчика прерываний.

Для того, чтобы очистить (сбросить) любой из битов EINT0...EINT3 регистра EXTINT, необходимо предварительно программным путем записать в него логическую 1. В этом случае, чтение соответствующего бита возвращает 0. В режиме, когда внешнее прерывание инициируется активным уровнем сигнала, сбросить таким способом флаг EINTn можно лишь после того, как прекратилась подача активного уровня сигнала на вход внешнего прерывания.

Таблица 3.3 – Битовая структура регистра EXTINT

Номер бита	Название бита	Функция бита	Значение после сброса
0	EINT0	Когда внешнее прерывание вызывается уровнем сигнала, бит устанавливается, если для вывода микроконтроллера выбрана функция EINT0 и на этот вывод подан внешний сигнал активного уровня. Когда внешнее прерывание вызывается фронтом/срезом сигнала, бит устанавливается, если для вывода микроконтроллера выбрана функция EINT0 и на этот вывод подан активный перепад внешнего сигнала. Для сброса бита, следует программно записать в него “1” (кроме случая, когда внешнее прерывание вызывается уровнем сигнала и этот внешний активный уровень присутствует на входе внешнего прерывания)	0
1	EINT1	Назначение бита аналогично биту EINT0	0
2	EINT2	Назначение бита аналогично биту EINT0	0
3	EINT3	Назначение бита аналогично биту EINT0	0
4:7	Зарезервированы	Биты зарезервированы. Пользовательское ПО не должно производить запись в эти биты. Чтение этих битов возвращает неопределенное значение	Неопределено

Битовая структура регистра пробуждения от внешних прерываний EXTWAKE приведена ниже в таблице 3.4.

В зависимости от состояния регистра EXTWAKE, внешним прерываниям разрешается или запрещается “пробуждать” микроконтроллер, если он находится в режиме Power Down. Чтобы “пробуждение” состоялось, для вывода микроконтроллера должна быть задана соответствующая функция EINTn. При этом, для “пробуждения” не требуется разрешения данного прерывания в контроллере векторов прерываний, что предоставляет пользователю дополнительные возможности. Например, можно произвести “пробуждение” из режима Power Down без вызова прерывания (простое возобновление выполнения программы), либо, наоборот, выполнить в режиме Power Down разрешенное прерывание без “пробуждения” микроконтроллера. Таким образом обеспечивается возможность обойтись без запрета внешнего прерывания, если в конкретном пользовательском приложении “пробуждение” из режима Power Down нежелательно.

Таблица 3.4 – Битовая структура регистра EXTWAKE

Номер бита	Название бита	Функция бита	Значение после сброса
0	EXTWAKE0	Если бит установлен в “1”, заданный активный уровень или перепад на входе EINT0 позволяет “пробудить” микроконтроллер из режима Power Down	0
1	EXTWAKE1	Назначение бита аналогично биту EXTWAKE0	0
2	EXTWAKE2	Назначение бита аналогично биту EXTWAKE0	0
3	EXTWAKE3	Назначение бита аналогично биту EXTWAKE0	0
4:7	Зарезервированы	Биты зарезервированы. Пользовательское ПО не должно производить запись в эти биты. Чтение этих битов возвращает неопределенное значение	Неопределено

Битовая структура регистра режима внешних прерываний EXTMODE приведена ниже в таблице 3.5.

Биты этого регистра управляют возможностью задавать, по какому событию (уровень внешнего сигнала или его фронт/срез на некотором выводе) будет вызываться каждое из прерываний EINTn. Внешние прерывания могут быть инициированы входными сигналами только на тех выводах, для которых выбрана функция EINTn, а также разрешена функция обработки внешних прерываний в регистре VICIntEnable. Если внешнее прерывание было запрещено в регистре VICIntEnable, перед повторным разрешением прерывания пользовательская программа должна записать логическую “1” в соответствующий бит регистра EXTINT, чтобы сбросить этот бит, который ранее мог быть аппаратно установлен.

Битовая структура регистра полярности внешних прерываний EXTPOLAR приведена ниже в таблице 3.6.

Когда внешнее прерывание генерируется уровнем сигнала, биты этого регистра задают, будет ли этот активный уровень высоким или низким. Когда внешнее прерывание генерируется перепадом сигнала, биты этого регистра задают, будет ли этот перепад фронтом или срезом. Функция обработки внешнего прерывания может быть инициирована входными сигналами только на тех выводах, которые выбраны для выполнения функции EINTn, и прерывания от которых разрешены в регистре VICIntEnable. Стоит отметить, что входные сигналы на выводах, назначенных для выполнения других функций, могут также вызывать прерывания от этих функций.

Если внешнее прерывание было запрещено в регистре VICIntEnable, перед повторным разрешением прерывания пользовательская программа должна записать логическую “1” в соответствующий бит регистра EXTINT, чтобы

сбросить этот бит, который мог быть ранее аппаратно установлен при изменении полярности внешних сигналов.

Таблица 3.5 – Битовая структура регистра EXTMODE

Номер бита	Название бита	Функция бита	Значение после сброса
0	EXTMODE0	Если бит сброшен, прерывание EINT0 вызывается уровнем внешнего сигнала. Если бит установлен, прерывание EINT0 вызывается перепадом внешнего сигнала.	0
1	EXTMODE1	Назначение бита аналогично биту EXTMODE0	0
2	EXTMODE2	Назначение бита аналогично биту EXTMODE0	0
3	EXTMODE3	Назначение бита аналогично биту EXTMODE0	0
4:7	Зарезервированы	Биты зарезервированы. Пользовательское ПО не должно производить запись в эти биты. Чтение этих битов возвращает неопределенное значение	Неопределено

Таблица 3.6 – Битовая структура регистра EXTPOLAR

Номер бита	Название бита	Функция бита	Значение после сброса
0	EXTPOLAR0	Если бит сброшен, прерывание EINT0 вызывается низким уровнем или срезом внешнего сигнала (в зависимости от значения бита EXTMODE0). Если бит установлен, прерывание EINT0 вызывается высоким уровнем или фронтом внешнего сигнала (в зависимости от значения бита EXTMODE0)	0
1	EXTPOLAR1	Назначение бита аналогично биту EXTPOLAR0	0
2	EXTPOLAR2	Назначение бита аналогично биту EXTPOLAR0	0
3	EXTPOLAR3	Назначение бита аналогично биту EXTPOLAR0	0
4:7	Зарезервированы	Биты зарезервированы. Пользовательское ПО не должно производить запись в эти биты. Чтение этих битов возвращает неопределенное значение	Неопределено

Для каждого из внешних прерываний EINT3:0 в регистре выбора выводов можно программно задавать соответствие тем или иным выводам микроконтроллера (выбирая их из определенного множества доступных выводов). Состояние встроенных логических модулей микроконтроллера, отвечающих за обработку внешних прерываний EINT3:0, определяется входными сигналами на соответствующих этим прерываниям выводах, а также заданными программно битами выбора этих выводов для функции EINTn.

Если для выбранного внешнего прерывания назначено более одного вывода, выходные состояния аппаратной логики внешних прерываний определяются состояниями соответствующих битов в регистрах режима и полярности. Ниже приведены возможные варианты такой установки:

- в режиме с активным низким уровнем все выводы, назначенные для функции EINTn, объединены по формуле логическое И (AND);
- в режиме с активным высоким уровнем все выводы, назначенные для функции EINTn, объединены по формуле логическое ИЛИ (OR);
- в режиме с активным передним фронтом, независимо от полярности, используется вывод с наименьшим номером линии порта ввода/вывода общего назначения GPIO.

Назначение нескольких выводов внешних прерываний EINTn в режиме с активным передним фронтом не рекомендуется, т.к. может привести к ошибке.

Ниже, на рисунке 3.1, приведена схема электрическая структурная узла обработки внешних прерываний.

Если во множестве выводов внешних прерываний имеется более одного вывода EINTn и они объединены по схеме логическое ИЛИ, подпрограмма обработки прерывания может считывать состояния этих выводов как линий ввода/вывода общего назначения (GPIO) с использованием регистров IOPIN0 и IOPIN1, чтобы определить, на каком из выводов сигнал вызвал прерывание.

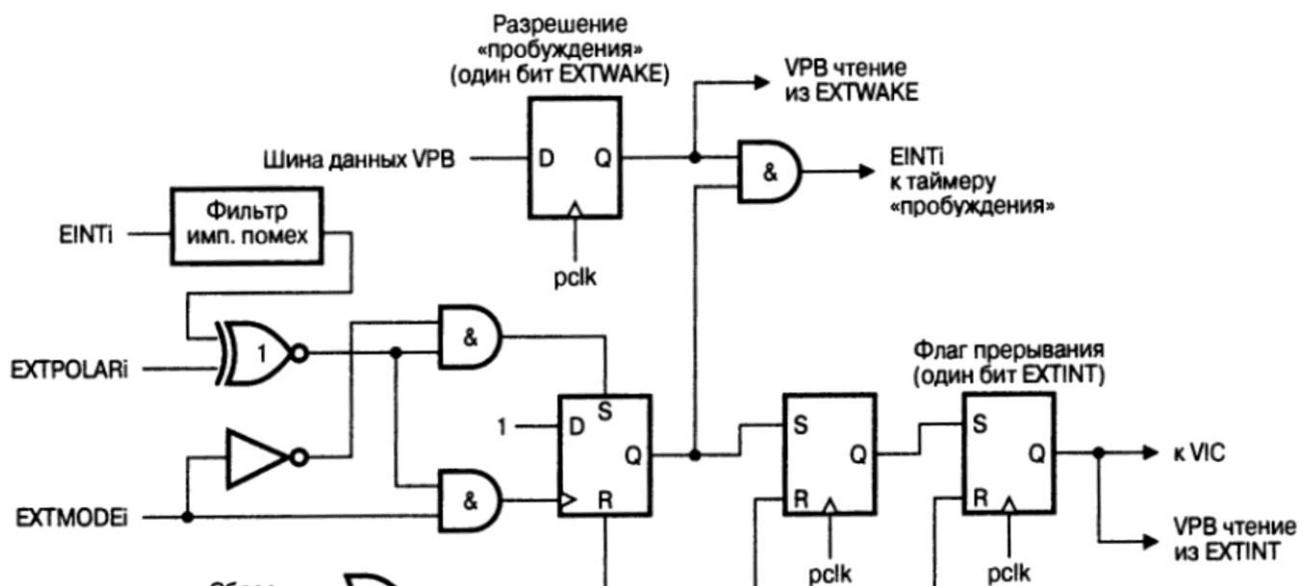


Рисунок 3.1 – Структурная схема узла обработки внешних прерываний EINT

3.1.3 Система фазовой автоподстройки частоты

Микроконтроллеры семейства LPC2000 имеют встроенную систему фазовой автоподстройки частоты – ФАПЧ (или PLL – Phase Locked Loop). Входной сигнал системы ФАПЧ может лежать в частотном диапазоне 10 ... 25 МГц. Эта частота, умноженная на некоторый коэффициент, с помощью управляемого током генератора Current Controlled Oscillator (ССО) преобразуется в частоту *csclk*, лежащую в диапазоне 10 ... 60 МГц. Множителем является целое число в интервале 1 ... 32 (однако, значение множителя в микроконтроллерах LPC2000 вследствие наличия верхнего предела тактовой частоты ядра не превышает 6). Управляемый ток генератор ССО работает в диапазоне 156 ... 320 МГц, поэтому для того, чтобы частота *csclk* на выходе ФАПЧ находилась в диапазоне 10 ... 60 МГц, в цепь обратной связи ФАПЧ включен дополнительный программируемый делитель. Его коэффициент деления может равняться 2, 4, 8 или 16. Поскольку минимальный коэффициент деления выходного делителя ФАПЧ равен двум, с выхода ФАПЧ всегда снимается “меандр”. Схема электрическая структурная узла ФАПЧ приведена на рисунке 3.2.

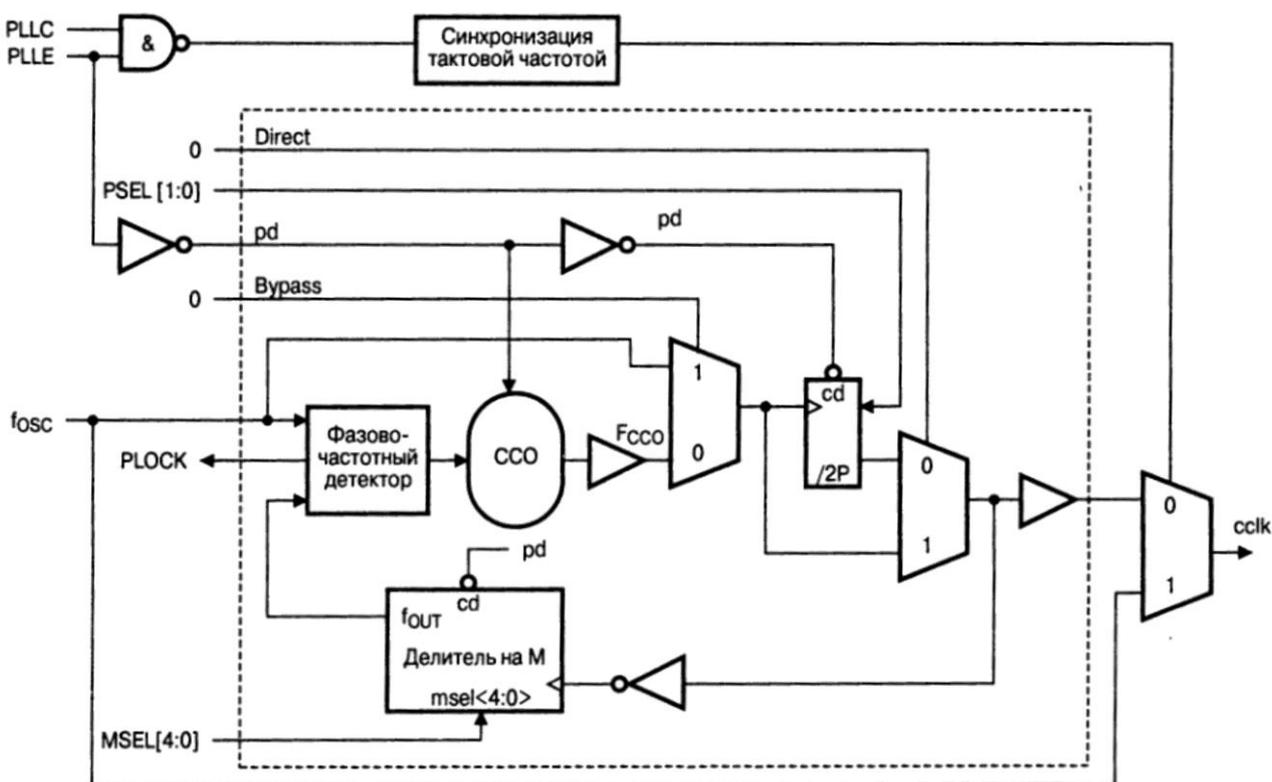


Рисунок 3.2 – Структурная схема узла ФАПЧ

Активация системы ФАПЧ производится посредством регистра PLLCON. Значения множителя ФАПЧ и коэффициента деления выходного делителя задаются в регистре PLLCFG. Оба эти регистра имеют защиту, чтобы предотвратить случайное изменение параметров ФАПЧ или случайную её деактивацию. Защита настроек системы ФАПЧ, подобно защите сторожевого таймера, обес-

печивается тем, что для доступа к ним необходимо генерировать специальную последовательность.

По умолчанию, после сброса микроконтроллера, а также в режиме Power Down система ФАПЧ отключена, а ее работа может быть разрешена программой пользователя. Программа должна сконфигурировать и активизировать ФАПЧ, дождаться замыкания петли ФАПЧ, а затем скоммутировать выход ФАПЧ в качестве источника тактовой частоты микроконтроллера.

Управляющие регистры системы ФАПЧ представлены в таблице 3.7. Более детальное их описание здесь и далее приводиться не будет. Микроконтроллеры LPC2000, имеющие в своем составе модуль USB, помимо указанного в таблице 3.7 набора регистров ФАПЧ (PLL0), имеют дополнительный набор регистров ФАПЧ (PLL1).

Таблица 3.7 – Описание регистров системы ФАПЧ

Адрес	Название	Описание	Доступ
0xE01FC080	PLLCON	Регистр управления ФАПЧ. Служит для временного хранения информации, необходимой для модификации служебных битов ФАПЧ. Значения, записанные в регистр, не действуют, пока не выработана корректная последовательность ввода данных.	R/W
0xE01FC084	PLLCFG	Регистр конфигурации ФАПЧ. Служит для временного хранения информации, необходимой для модификации настроек конфигурации ФАПЧ. Значения, записанные в регистр, не действуют, пока не выработана корректная последовательность ввода данных.	R/W
0xE01FC088	PLLSTAT	Регистр состояния ФАПЧ. Чтение регистра возвращает информацию, только что записанную в регистры управления и конфигурации. Регистр не будет достоверно отражать состояние ФАПЧ в случае, если запись в PLLCON или PLLCFG была произведена, но корректная последовательность ввода данных не была выработана.	RO
0xE01FC08C	PLLFEED	Регистр подачи ФАПЧ. В регистре разрешается загрузка данных управления и конфигурации ФАПЧ из PLLCON и PLLCFG в теневые регистры, которые фактически обеспечивают управление и конфигурацию ФАПЧ.	WO

Следует всегда помнить, что некорректная установка настроек ФАПЧ может привести к неправильному функционированию микроконтроллера.

Переход в режим Power Down автоматически выключает ФАПЧ и отключает ее от микроконтроллера. “Пробуждение” из режима Power Down не влечет за собой автоматического восстановления параметров настройки ФАПЧ, поэтому оно должно быть выполнено пользовательской программой. Для этого, разрабатывается подпрограмма, которая должна активизировать ФАПЧ, дождаться замыкания ее петли, а затем подключить ФАПЧ к микроконтроллеру. Её можно вызывать в начале любого программного блока обработки прерывания, которому передается управление после “пробуждения”.

Не следует пытаться перезапустить систему ФАПЧ после пробуждения из режима Power Down просто путем выработки корректной последовательности ввода данных для текущего содержимого управляющих регистров. Необходимо полностью произвести включение ФАПЧ и подключение ее к микроконтроллеру, после чего следует дождаться замыкания петли ФАПЧ.

3.1.4 Узел управления потребляемой мощностью

Микроконтроллеры семейства LPC2000 поддерживают 2 режима пониженного энергопотребления: Idle Mode и Power Down. В режиме Idle Mode выполнение команд приостановлено, однако сброс и прерывания разрешены. Периферийные устройства продолжают функционировать и могут генерировать прерывания, которые заставляют микроконтроллер возобновить выполнение программы. В режиме Idle Mode мощность, потребляемая непосредственно ядром, а также системами памяти, связанными с ними устройствами и внутренними шинами снижается до нуля.

В режиме Power Down тактовый генератор остановлен, и микроконтроллер не получает никаких сигналов синхронизации. Состояние процессора и его регистров, а также регистров периферийных устройств и содержимое статического ОЗУ сохраняются неизменными. Также сохраняются логические уровни на выводах микроконтроллера. Выход из режима Power Down в режим нормального выполнения программы может быть произведен путем сброса или с использованием определенных прерываний, которые могут генерироваться даже при отсутствии синхросигнала. Так как все динамические операции в микроконтроллере приостановлены, потребляемая им мощность в режиме Power Down уменьшается практически до нуля.

Включение режимов Idle Mode и Power Down должно быть скоординировано с выполнением программы. “Пробуждение” из режимов Idle Mode и Power Down через прерывание возобновляет выполнение программы таким образом, что не происходит потери каких-либо команд, их неполного или повторного выполнения.

Управление потребляемой мощностью периферийных устройств микроконтроллера LPC2000 организовано таким образом, чтобы обеспечивать возможность индивидуального отключения, если они не используются пользова-

тельским приложением. Это позволяет снизить непроизводительное энергопотребление системы.

Управление потребляемой микроконтроллером мощностью осуществляется с помощью двух регистров, описание которых приведено в таблице 3.8.

Таблица 3.8 – Описание регистров управления потребляемой мощностью

Адрес	Название	Описание	Доступ
0xE01FC0C0	PCON	Регистр управления мощностью потребления. Содержит служебные биты, которые разрешают переход в один из двух режимов пониженного энергопотребления.	R/W
0xE01FC0C4	PCONP	Регистр управления мощностью потребления периферийных устройств. Содержит служебные биты, индивидуально включающие и отключающие питание периферийных устройств микроконтроллера.	R/W

Битовая структура регистра управления мощностью потребления PCON приведена ниже в таблице 3.9.

Регистр PCON содержит 2 бита. Запись “1” в соответствующий бит вызывает переход микроконтроллера в режим Power Down или режим Idle mode. Если установлены оба бита, то микроконтроллер находится в режиме Power Down.

Таблица 3.9 – Битовая структура регистра PCON

Номер бита	Название бита	Функция бита	Значение после сброса
0	IDL	Установка бита переводит микроконтроллер в режим Idle mode и останавливает тактирование, но встроенные периферийные устройства остаются активными	0
1	PD	Установка бита переводит микроконтроллер в режим Power Down, останавливает тактирование процессора и всей встроенной периферии	0
2:7	Зарезервированы	Биты зарезервированы. Пользовательское ПО не должно производить запись в эти биты. Чтение этих битов возвращает неопределенное значение	Неопределено

Битовая структура регистра управления мощностью потребления периферийных устройств PCONP приведена ниже в таблице 3.10.

Регистр PCONP позволяет отключать периферийные устройства, для уменьшения потребляемой мощности. Некоторые периферийные устройства не могут быть отключены (например, сторожевой таймер, модуль GPIO, блок коммутации выводов и блок управления микроконтроллером). Каждый бит в регистре PCONP отвечает за одно периферийное устройство. Номера битов соответствуют периферийным устройствам, которые подключены к шине VPB микроконтроллера (смотри рисунок 2.6).

Таблица 3.10 – Битовая структура регистра PCONP

Номер бита	Название бита	Функция бита	Значение после сброса
1	2	3	4
1	PCTIM0	Если бит установлен, разрешается работа TIMER0. Если бит сброшен, TIMER0 заблокирован и не потребляет мощности	1
2	PCTIM1	Назначение бита аналогично биту PCTIM0	1
3	PCURT0	Если бит установлен, разрешается работа UART0. Если бит сброшен, UART0 заблокирован и не потребляет мощности	1
4	PCURT1	Назначение бита аналогично биту PCURT0	1
5	PCPWM0	Если бит установлен, разрешается работа PWM0. Если бит сброшен, PWM0 заблокирован и не потребляет мощности	1
7	PCI2C	Если бит установлен, разрешается работа интерфейса I ² C. Если бит сброшен, интерфейс I ² C заблокирован и не потребляет мощности	1
8	PCSPI0	Если бит установлен, разрешается работа интерфейса SPI. Если бит сброшен, интерфейс SPI заблокирован и не потребляет мощности	1
9	PCRTC	Если бит установлен, разрешается работа модуля реального времени RTC. Если бит сброшен, модуль RTC заблокирован и не потребляет мощности	1
10	PCSPI1	Назначение бита аналогично биту PCSPI0	1

Продолжение таблицы 3.10

1	2	3	4
12	PCAD	Если бит установлен, разрешается работа модуля АЦП. Если бит сброшен, модуль АЦП заблокирован и не потребляет мощности	1
13	PCCAN1	Если бит установлен, разрешается работа контроллера CAN1. Если бит сброшен, контроллер CAN1 заблокирован и не потребляет мощности. Работа приемного фильтра разрешена, если разрешена работа любого из контроллеров CAN (1 или 2).	1
14	PCCAN2	Назначение бита аналогично биту PCCAN1	1
0, 6, 11, 15:31	Зарезервированы	Биты зарезервированы. Пользовательское ПО не должно производить запись в эти биты. Чтение этих битов возвращает неопределенное значение	Неопределено

Сразу после сброса микроконтроллера в регистре PCONP содержатся значения, разрешающие работу всех управляемых регистром интерфейсов и периферийных устройств. Поэтому пользовательскому приложению нет необходимости модифицировать содержимое PCONP, кроме случая, когда какое-либо периферийное устройство в приложении не используется и его можно отключить, в целях экономии потребляемой мощности. В регистре PCONP должны быть установлены только те биты, которые соответствуют периферийным устройствам, действительно используемым в приложении. Все другие биты, в том числе зарезервированные, должны быть сброшены.

3.1.5 Узел сброса микроконтроллера

В микроконтроллерах семейства LPC2000 предусмотрено 2 источника сброса: вход внешнего сброса RESET и сброс от сторожевого таймера. Вход RESET имеет встроенный триггер Шмитта, который является дополнительным фильтром помех. После сброса от любого из источников сразу же запускается встроенный таймер “пробуждения”. Активный уровень сигнала сброса должен удерживаться до окончания счета этого таймера. Когда тактовый генератор вырабатывает заданное количество импульсов, подсчитанных таймером “пробуждения”, сигнал сброса поступает в контроллер Flash-памяти чем производит его инициализацию.

Ниже на рисунке 3.3. изображена структурная схема внутреннего узла сброса. На ней отображены связи между цепями сброса, генератором и таймером пробуждения. Фильтр помех в цепи сброса позволяет микроконтроллеру игнорировать внешние импульсы сброса, длительность которых недостаточна, а также задавать минимальную длительность импульса RESET, который дол-

жен быть подан на одноименный вход, чтобы гарантировать сброс устройства. Такая схема организации узла сброса применена производителем с целью обеспечения стабильности генерации тактовых импульсов и получения устойчивой частоты тактирования. Предполагается, что для достижения такой стабильности активный уровень сигнала сброса после включения питания микроконтроллера должен удерживаться на входе RESET не менее 10 мс. Для всех последующих сбросов, когда тактовый генератор уже находится в режиме устойчивой генерации и на соответствующем входе присутствуют стабильные тактовые импульсы, минимальная длительность импульса сброса равна 300 нс.

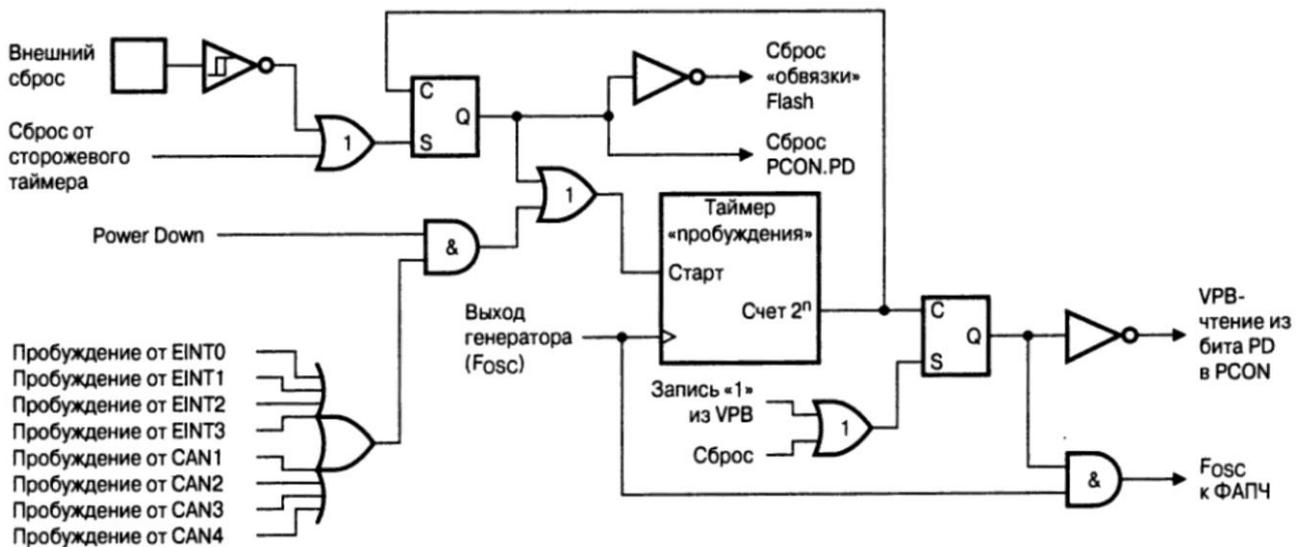


Рисунок 3.3 – Структурная схема узла сброса микроконтроллера

Как правило, к номиналам питающих напряжений на выводах питания микроконтроллера не предъявляется особых требований. Однако для осуществления корректного сброса необходимо подать заданный уровень напряжения на соответствующий вывод, поскольку цепи сброса, тактовый генератор и другие аппаратные средства, участвующие в сбросе, питаются от него. Отсутствие питающего напряжения на выводе, обеспечивающем питание интерфейса микроконтроллера через цифровые линии ввода/вывода, не нарушит последовательность сброса, но заблокирует связь микроконтроллера с внешним миром.

После завершения процедуры внутреннего сброса, ядро микроконтроллера приступает к выполнению программы с адреса 0, на который и отображен вектор сброса в блоке загрузки. В этот момент само ядро и периферийные регистры инициализируются установленными по умолчанию значениями.

Процедуры внешнего и внутреннего сбросов имеют небольшие различия. При внешнем сбросе функции некоторых выводов автоматически реконфигурируются. При внутреннем сбросе этого не происходит, т.к. внешние цепи микроконтроллера, отвечающие за сброс, не могут отслеживать наступления состояния сброса и производить в этот момент реконфигурирование функций выводов. Вывод P0.14 может использоваться для нужд встроенного загрузчика при выполнении программного кода после завершения процедуры сброса.

Поскольку сброс микроконтроллера обычно происходит в случайные моменты времени, то возможно наступление сброса во время осуществления операций программирования или стирания встроенной Flash-памяти. В этом случае, контроллер Flash-памяти прервёт указанные операции, и будет удерживать центральный процессор в состоянии сброса до тех пор, пока внутренние высокие напряжения программирования не будут сняты.

3.1.6 Делитель частоты для шины периферийных устройств VPB

Делитель частоты для шины VPB служит для задания соотношений между тактовой частотой ядра (сclk) и частотой, используемой для тактирования периферийных устройства (ps1k).

Использование в микроконтроллерах семейства LPC2000 такого делителя преследует 2 цели. Первая состоит в том, чтобы обеспечить периферийные устройства через шину VPB синхросигналом ps1k такой частоты, чтобы они могли обеспечить необходимую производительность. Для достижения этого, частота синхросигнала шины VPB может быть снижена до значения 1/2 или 1/4 тактовой частоты ядра. Значение частоты синхросигнала шины VPB после сброса микроконтроллера составляет 1/4 тактовой частоты ядра (задано по умолчанию) и может быть изменено только через регистры управления VPB. Таким образом, гарантируется устойчивая работа шины VPB сразу при включении питания микроконтроллера.

Вторая цель состоит в том, чтобы обеспечить возможность экономии потребляемой электроэнергии микроконтроллером, если приложение пользователя не требует использования периферийных устройств, работающих на тактовой частоте ядра.

Структурная схема подключения делителя частоты шины VPB к цепям тактирования микроконтроллера показана на рисунке 3.4. Как видно, делитель частоты шины VPB связан с выходом системы ФАПЧ, поэтому система ФАПЧ должна оставаться активной (если она, конечно, вообще используется) во время функционирования в режиме Idle mode.

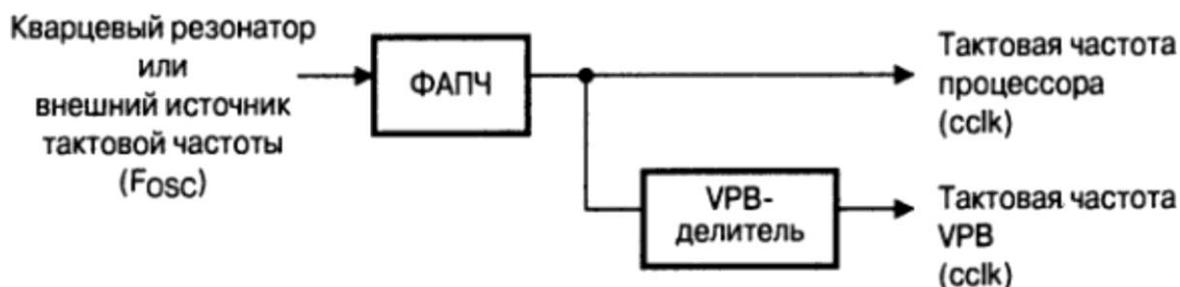


Рисунок 3.4 – Подключение делителя частоты шины VPB

Для управления значением тактовой частоты периферийных устройств используется регистр VPBDIV, который размещается по адресу 0xE01FC100. Битовая структура этого регистра представлена ниже, в таблице 3.11. Регистр делителя шины VPB содержит 2 бита и позволяет задать 3 значения для коэффициента деления.

Таблица 3.11 – Битовая структура регистра VPBDIV

Номер бита	Название бита	Функция бита	Значение после сброса
0:1	VPBDIV	В зависимости от состояния битов, частота синхросигнала VPB может принимать следующие значения: 00: частота VPB равна 1/4 частоты тактового сигнала ядра. 01: частота VPB равна частоте тактового сигнала ядра. 10: Частота VPB равна 1/2 частоты тактового сигнала ядра. 11: Зарезервировано. При записи этого значения в регистр VPBDIV сохраняется текущее значение битов VPBDIV	0
4:5	XCLKDIV	В микроконтроллерах со 144-выводным корпусом эти биты управляют синхросигналом, который формируется внутренней логикой микроконтроллера и может подаваться на вывод XCLK. В зависимости от состояния этих битов, частота синхросигнала может принимать следующие значения: 00: частота XCLK равна 1/4 частоты тактового сигнала ядра. 01: частота XCLK равна частоте тактового сигнала ядра. 10: Частота XCLK равна 1/2 частоты тактового сигнала ядра. 11: Зарезервировано. При записи этого значения в регистр VPBDIV сохраняется текущее значение битов VPBDIV	0
2:3, 6:7	Зарезервированы	Биты зарезервированы. Пользовательское ПО не должно производить запись в эти биты. Чтение этих битов возвращает неопределенное значение	0

3.1.7 Таймер “пробуждения” микроконтроллера

Назначение таймера “пробуждения” состоит в том, чтобы гарантировать, что тактовый генератор и другие необходимые встроенные аналоговые схемы микроконтроллера полностью выйдут на свои рабочие режимы прежде, чем ядру будет разрешено выполнять команды. Таймер “пробуждения” используется в следующих случаях:

- при включении питания микроконтроллера;

- при внутреннем или внешнем сбросе;
- после отключения любой из аналоговых схем.

Таймер “пробуждения” используется также при любом “пробуждении” микроконтроллера из режима Power Down, т.к. в этом режиме генератор и другие функции отключены.

Таймер “пробуждения” контролирует работу стабилизированного кварцевым резонатором генератора, определяя степень готовности микроконтроллера к выполнению программы пользователя. Когда производится включение питания или микроконтроллер “пробуждается” из режима Power Down, требуется некоторое время, чтобы вырабатываемый генератором синхросигнал достиг амплитуды, достаточной для тактирования логических схем синхронизации микроконтроллера. Это время зависит от многих факторов: времени нарастания питающего напряжения, типа кварцевого резонатора, его электрических характеристик или характеристик иных внешних схем задания частоты (например, конденсаторов), а также характеристик самого генератора при заданных условиях окружающей среды.

Обнаружив наличие в системе сигнала синхронизации требуемой частоты, таймер “пробуждения” отсчитывает 4096 импульсов, а затем разрешает инициализацию Flash-памяти. После завершения инициализации Flash-памяти и при условии снятия сигнала внешнего сброса, ядро готово выполнять команды. Если в системе используется внешний источник тактовых сигналов, имеется возможность свести к минимуму или вовсе исключить задержку таймера “пробуждения” после появления синхроимпульсов. Таким образом, наличие таймера “пробуждения” гарантирует, что любые узлы и модули микроконтроллера будут функционировать в рабочих режимах до начала выполнения программы.

Микроконтроллеры семейства LPC2000 не имеют в своем составе аналоговых функциональных модулей, способных работать без сигналов синхронизации (например, компараторов), а также независимых источников синхросигналов, например, отдельного генератора сторожевого таймера. Исключение составляют те микроконтроллеры, у которых модуль часов-календаря реального времени (RTC) тактируется от подключаемого отдельно генератора с внешним кварцевым резонатором на частоту 32768 Гц. Только 2 встроенные периферийные функции способны работать при отсутствии источника синхронизации – это внешние прерывания (EINT0 – EINT3) и CAN-контроллеры. В случае, если разрешено “пробуждение” от внешнего прерывания и происходит событие, вызывающее это прерывание, начинается цикл “пробуждения” генератора. Аналогично, цикл “пробуждения” генератора начинается, если разрешено “пробуждение” микроконтроллера от сигналов CAN и соответствующие сигналы появляются на шине CAN. Фактически, микроконтроллер реагирует на любое прерывание после того, как истекает время “пробуждения” и произведена обработка этого прерывания контроллером VIC.

Чтобы обеспечить возможность выхода микроконтроллера из режима Power Down с помощью других периферийных устройств, был реализован механизм мультиплексирования выводов. Можно разрешать прерывания от собы-

тий, относящихся к UART0 или UART1, SPI0 или SPI1, либо I²C, назначая соответствующим выводам микроконтроллера альтернативные функции.

Прежде чем перевести микроконтроллер в режим Power Down, программа пользователя должна обеспечить включение соответствующих функций или интерфейсов, задать необходимые альтернативные функции выводов, выбрать корректный режим и полярность для прерывания. При “пробуждении”, прежние функции тех или иных выводов могут быть восстановлены программно.

При работе с пользовательским приложением, возможна ситуация, когда “пробуждение” микроконтроллера осуществляется активным низким уровнем внешнего сигнала на одной из линий шины. Если один канал EINTn используется совместно несколькими выводами микроконтроллера, а “пробуждение” из режима Power Down требуется обеспечить по активному уровню на каком-то одном из этих выводов, для данного канала следует задать режим чувствительности к низкому уровню внешнего сигнала. Только в этом режиме сигналы на соответствующих линиях шины будут объединены функцией логическое ИЛИ (OR) и использованы для “пробуждения” микроконтроллера. Недостаток такого подхода заключается в том, что требуется фиксировать состояние шины или линии, которая вызвала “пробуждение”, на интервал времени, необходимый для стабилизации работы генератора.

В режиме Idle mode внутренние цепи, связанные с приемом внешних сигналов, отключены.

3.2 Контроллер векторов прерываний VIC

Контроллер векторов прерываний (VIC) обслуживает 32 входа запросов прерываний, которые могут назначаться для обслуживания в трех категориях:

- быстрый запрос прерывания (FIQ);
- векторизированный запрос прерывания (IRQ);
- не векторизированный запрос прерывания (IRQ).

Такие назначения могут программироваться. Поэтому приоритеты прерываний от различных периферийных устройств могут динамически назначаться и корректироваться в процессе выполнения программы пользователя.

Быстрый запрос прерывания (FIQ) имеет самый высокий приоритет. Если в данный момент времени поступил более чем один запрос FIQ, то контроллер векторов прерываний подает на ARM-ядро сигналы запросов FIQ, объединенные в соответствии с логической функцией ИЛИ. Самое малое из возможных времен ожидания обслуживания для FIQ достигается, если только один запрос прерывания классифицирован контроллером как FIQ, в этом случае сервисная подпрограмма обслуживания запускается для взаимодействия непосредственно с устройством, запросившим прерывание. Если в текущий момент времени имеется более одного запроса, классифицированного контроллером как FIQ, сервисная программа сначала считает из контроллера векторов прерываний слово, идентифицирующее источники, запросившие прерывания.

Векторизированные запросы прерываний имеют средний приоритет. Только 16 из 32 возможных запросов могут назначаться контроллером в эту ка-

тегорию. Любой из 32 запросов может быть назначен на любой из 16 слотов векторизованных запросов прерываний, среди которых слот 0 имеет самый высокий приоритет, а слот 15 – самый низкий.

Невекторизованные запросы прерываний обладают самым низким приоритетом.

Контроллер векторов прерываний объединяет в соответствии с функцией логическое ИЛИ сигналы от всех векторизованных и невекторизованных запросов прерываний (IRQ) для их подачи на ядро. Сервисная подпрограмма обслуживания запроса прерывания (IRQ) читает регистр из VIC и производит переход на требуемый вектор. Если поступает любой из векторизованных запросов прерывания, то VIC выделяет для него адрес с самым высоким приоритетом, соответствующий сервисной подпрограмме обслуживания запросов прерываний. В ином случае, VIC по умолчанию выделяет для запроса адрес подпрограммы, которая обслуживает все невекторизованные запросы прерываний. Эта выделяемая по умолчанию подпрограмма читает другой регистр VIC, чтобы определить, какие запросы прерываний в данный момент активны.

3.2.1 Регистры контроллера векторов прерываний

Все регистры контроллера векторов прерываний имеют размер слова. Запись и чтение байта и полуслова не поддерживаются. Структурная схема контроллера векторов прерываний приведена на рисунке 3.5. Все регистры, которые используются для управления VIC, указаны в таблице 3.12. После выполнения сброса содержимое абсолютно всех этих регистров равно нулю.

Запись “1” в соответствующий бит регистра программных прерываний VICSoftInt разрешает генерировать запрос прерывания с номером, соответствующим порядковому номеру этого бита. Появление “0” не разрешает генерировать запрос прерывания с соответствующим номером. Запись нулей в биты регистра VICSoftInt осуществляется через регистр VICSoftIntClear.

Запись “1” в заданный бит регистра сброса программных прерываний VICSoftIntClear сбрасывает соответствующий бит в регистре программных прерываний VICSoftInt, т.е. производит сброс соответствующего запроса. Запись “0” оставляет соответствующий бит в регистре неизменным.

Чтение “1” из указанного бита регистра состояния необработанных прерываний VICRawIntr означает, что запрос на прерывание или программное прерывание с номером, соответствующим порядковому номеру этого бита, присутствует. Чтение “0” означает, что запрос на прерывание или программное прерывание с этим порядковым номером отсутствует.

Чтение “1” из указанного бита регистра разрешения прерываний VICIntEnable означает, что соответствующий запрос прерывания или программного прерывания разрешен как запрос FIQ или IRQ. Запись “1” в бит регистра разрешает соответствующий запрос прерывания или программного прерывания как запрос FIQ или IRQ. Запись нулей в биты регистра VICIntEnable осуществляется через регистр VICIntEnClear.

Запись “1” в заданный бит регистра разрешения сброса прерываний VICIntEnClear сбрасывает соответствующий бит в регистре разрешения прерывания VICIntEnable, т.е. запрещает прерывание для этого запроса. Запись “0” оставляет соответствующий бит в регистре неизменным.

Запись или чтение “1” в/из заданный бит регистра выбора прерываний VICIntSelect означает, что запрос прерывания с этим порядковым номером назначен в категорию запросов FIQ. Запись или чтение “0” – запрос прерывания с этим порядковым номером назначен в категорию запросов IRQ.

Чтение “1” из указанного бита регистра состояния запросов IRQ VICIRQStatus означает, что запрос прерывания с номером, соответствующим порядковому номеру этого бита, разрешается, классифицируется как запрос IRQ и может присутствовать в системе.

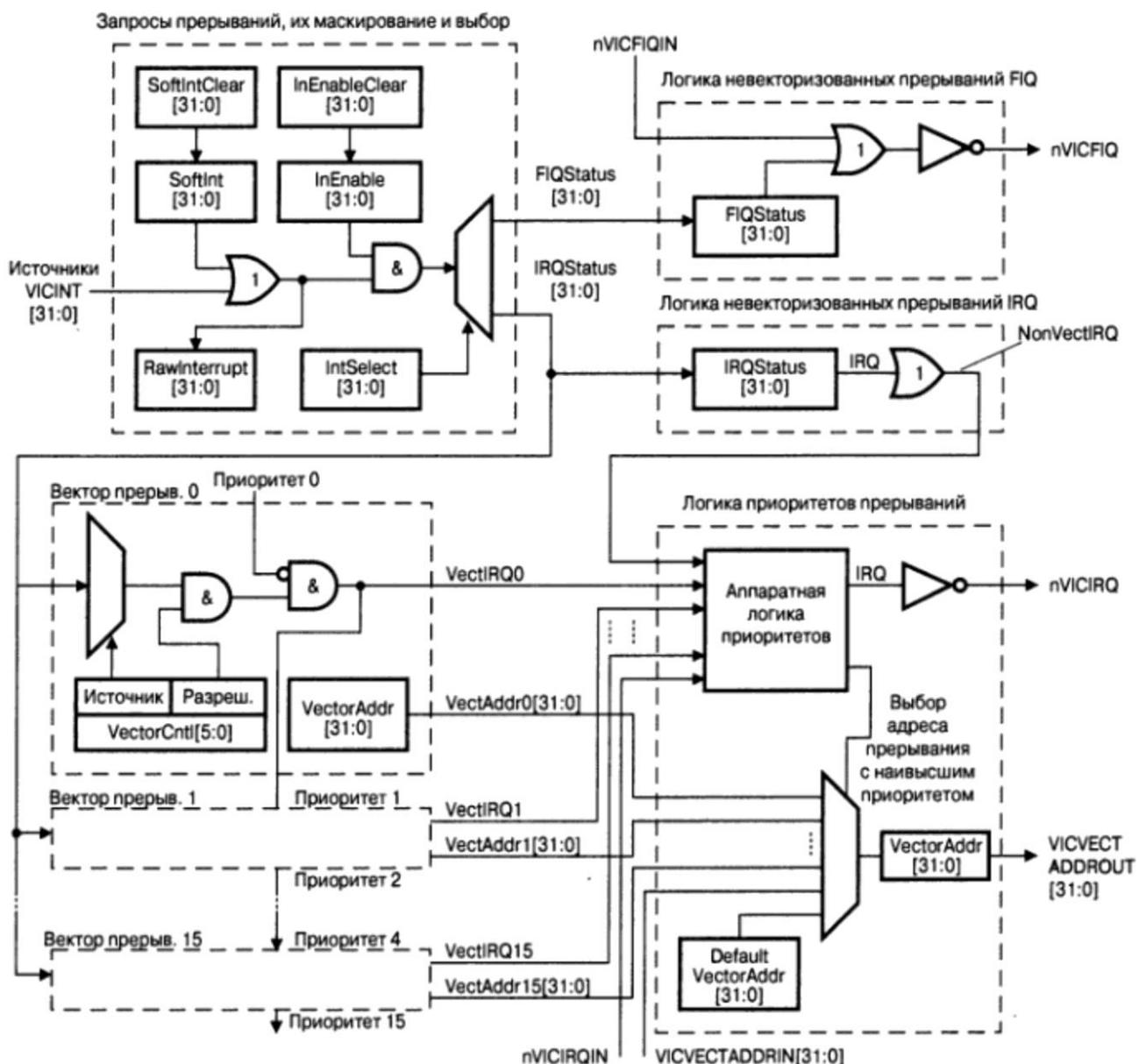


Рисунок 3.5 – Структурная схема контроллера векторов прерываний VIC

Чтение “1” из указанного бита регистра состояния запросов FIQ VICFIQStatus означает, что запрос прерывания с номером, соответствующим

порядковому номеру этого бита, разрешается, классифицируется как запрос FIQ и может присутствовать в системе.

Таблица 3.12 – Регистры контроллера векторов прерываний

Адрес	Название	Описание	Доступ
1	2	3	4
0xFFFF 0000	VICIRQStatus	Регистр состояния IRQ. Чтение регистра возвращает текущее состояние разрешенных запросов прерываний классифицированных как запросы IRQ	RO
0xFFFF F004	VICFIQStatus	Регистр состояния FIQ. Чтение регистра возвращает текущее состояние разрешенных запросов прерываний и классифицированных как запросы FIQ	RO
0xFFFF F008	VICRawIntr	Регистр состояния необработанных прерываний. Чтение регистра возвращает текущее состояние 32 запросов прерываний/программных прерываний вне зависимости от их разрешенности или классификации	RO
0xFFFF F00C	VICIntSelect	Регистр выбора прерываний. Классифицирует каждое из 32 запросов прерываний как запрос FIQ или запрос IRQ	R/W
0xFFFF F010	VICIntEnable	Регистр разрешения прерываний. Управляет разрешениями, которые даются каждому из 32 запросов прерываний и программным прерываниям как FIQ или IRQ	R/W
0xFFFF F014	VICIntEnClear	Регистр разрешения сброса разрешенных прерываний. Разрешает программе сбрасывать один или более битов в регистре разрешения прерываний	W
0xFFFF F018	VICSoftInt	Регистр программных прерываний. Содержимое регистра побитно объединяется функцией ИЛИ с сигналами 32 запросов прерываний от различных периферийных устройств	R/W

Продолжение таблицы 3.12

1	2	3	4
0xFFFF F01C	VICSoftIntClear	Регистр сброса программных прерываний. Разрешает программе сбрасывать один или более битов в регистре программных прерываний	W
0xFFFF F020	VICProtection	Регистр разрешения защиты. Позволяет ограничивать доступ к регистрам VIC для программ, выполняющихся в режиме пользователя	R/W
0xFFFF F030	VICVectAddr	Регистр адреса вектора. При запросе IRQ, сервисная подпрограмма обслуживания запросов IRQ может прочитать этот регистр и передать управление по прочитанному адресу	R/W
0xFFFF F034	VICDefVectAddr	Регистр адреса вектора по умолчанию. Содержит адрес для сервисной подпрограммы обработки прерываний (ISR) для не векторизированных запросов прерываний	R/W
0xFFFF F100	VICVectAddr0	Регистр адреса вектора 0. Регистры адреса векторов 0 ... 15 содержат адреса для сервисных подпрограмм обработки прерываний для 16 векторных слотов IRQ	R/W
0xFFFF F104	VICVectAddr1	Регистр адреса вектора 1	R/W
0xFFFF F108	VICVectAddr2	Регистр адреса вектора 2	R/W
0xFFFF F10C	VICVectAddr3	Регистр адреса вектора 3	R/W
0xFFFF F110	VICVectAddr4	Регистр адреса вектора 4	R/W
0xFFFF F114	VICVectAddr5	Регистр адреса вектора 5	R/W
0xFFFF F118	VICVectAddr6	Регистр адреса вектора 6	R/W
0xFFFF F11C	VICVectAddr7	Регистр адреса вектора 7	R/W
0xFFFF F120	VICVectAddr8	Регистр адреса вектора 8	R/W
0xFFFF F124	VICVectAddr9	Регистр адреса вектора 9	R/W
0xFFFF F128	VICVectAddr10	Регистр адреса вектора 10	R/W
0xFFFF F12C	VICVectAddr11	Регистр адреса вектора 11	R/W
0xFFFF F130	VICVectAddr12	Регистр адреса вектора 12	R/W
0xFFFF F134	VICVectAddr13	Регистр адреса вектора 13	R/W
0xFFFF F138	VICVectAddr14	Регистр адреса вектора 14	R/W
0xFFFF F13C	VICVectAddr15	Регистр адреса вектора 15	R/W

Продолжение таблицы 3.12

1	2	3	4
0xFFFF F200	VICVectCntl0	Регистр управления вектором 0. Регистры управления векторами 0 ... 15 управляют одним из 16 векторных слотов IRQ. Слот 0 имеет самый высокий приоритет, а слот 15 – самый низкий	R/W
0xFFFF F204	VICVectCntl1	Регистр управления вектором 1	R/W
0xFFFF F208	VICVectCntl2	Регистр управления вектором 2	R/W
0xFFFF F20C	VICVectCntl3	Регистр управления вектором 3	R/W
0xFFFF F210	VICVectCntl4	Регистр управления вектором 4	R/W
0xFFFF F214	VICVectCntl5	Регистр управления вектором 5	R/W
0xFFFF F218	VICVectCntl6	Регистр управления вектором 6	R/W
0xFFFF F21C	VICVectCntl7	Регистр управления вектором 7	R/W
0xFFFF F220	VICVectCntl8	Регистр управления вектором 8	R/W
0xFFFF F224	VICVectCntl9	Регистр управления вектором 9	R/W
0xFFFF F228	VICVectCntl10	Регистр управления вектором 10	R/W
0xFFFF F22C	VICVectCntl11	Регистр управления вектором 11	R/W
0xFFFF F230	VICVectCntl12	Регистр управления вектором 12	R/W
0xFFFF F234	VICVectCntl13	Регистр управления вектором 13	R/W
0xFFFF F238	VICVectCntl14	Регистр управления вектором 14	R/W
0xFFFF F23C	VICVectCntl15	Регистр управления вектором 15	R/W

Запись или чтение “1” в/из заданный бит регистра разрешения защиты VICProtection означает, что к регистрам VIC можно обращаться только в привилегированном режиме. Запись или чтение “0” – к регистрам VIC можно обращаться в пользовательском или привилегированном режимах.

Если один или более запросов прерываний или программных прерываний разрешены, классифицированы как запросы IRQ, присутствуют в системе и назначены на включенный векторный слот запроса IRQ, то чтение регистра адреса вектора VICVectAddr возвращает адрес, хранящийся в регистре адреса вектора для слота с самым высоким приоритетом (VICVectAddr0). Если эти условия не выполнены, то чтение этого регистра возвращает адрес, хранящийся в заданном по умолчанию регистре адреса вектора VICVectAddr. Запись данных в этот регистр не означает, что соответствующий адрес будет использоваться и при обработке последующих прерываний. Каждый раз в конце сервисной подпрограммы обработки прерывания в этот регистр должна быть произведена запись, чтобы обновить состояние аппаратных средств микроконтроллера.

Запись или чтение в/из регистр адреса вектора по умолчанию VICDefVectAddr устанавливает/возвращает адрес сервисной подпрограммы обработки прерываний по умолчанию. Это необходимо в случае, если в момент времени, когда сервисная подпрограмма запроса IRQ читает регистр адреса

вектора (VICVectAddr), и ни один из слотов IRQ не отвечает, то чтение возвращает адрес, содержащийся в регистре VICDefVectAddr.

Если один или более запросов прерываний или программных прерываний разрешены, классифицированы как запросы IRQ, присутствуют в системе и назначены на включенный векторный слот запроса IRQ, то из заданного регистра адреса вектора (VICVectAddrn) сервисная подпрограмма обработки запросов IRQ для слота с самым высоким приоритетом прочитает адрес вектора обработчика прерывания.

Битовая структура n-го регистра управления векторами прерываний VICVectCntln представлена ниже, в таблице 3.13.

Таблица 3.13 – Битовая структура регистра VICVectCntln

Номер бита	Название бита	Функция бита	Значение после сброса
0:4	VICVectCntN	Номер запроса прерывания или программного прерывания, назначенного на этот векторный слот запроса IRQ. ПО не должно назначать одно прерывание более чем одному разрешенному векторному слоту запроса IRQ. Если же это произошло, следует использовать слот с наименьшим номером, если запрос прерывания или программного прерывания разрешен, классифицируется как запрос IRQ и присутствует в системе	0
5	VICVectCntV	Запись “1” включает векторный слот запроса IRQ, и может определять уникальный адрес сервисной подпрограммы обработки прерывания (ISR), если разрешен назначенный запрос прерывания или программного прерывания, который классифицируется как запрос IRQ и присутствует в системе	0
6:31	Зарезервированы	Биты зарезервированы. Пользовательское ПО не должно производить запись в эти биты. Чтение этих битов возвращает неопределенное значение	Неопределено

Следует отметить, что к каждому периферийному устройству подключена одна линия прерывания, связанная с VIC. Каждое периферийное устройство может иметь несколько внутренних флагов прерываний. Более конкретно об этом можно узнать из документации на заданный микроконтроллер.

3.2.2 Ложные прерывания

В микроконтроллерах с ядром ARM7TDMI могут происходить так называемые ложные прерывания, возникающие из-за асинхронной обработки запросов прерываний. Асинхронный характер обработки прерывания является следствием характера взаимодействия ядра и VIC. Если состояние VIC изменяется между моментами, когда ядро обнаруживает прерывание и фактически обрабатывает прерывание, могут возникнуть некоторые проблемы. В реальном приложении пользователя может иметь место следующий сценарий:

- 1) VIC “решает”, что есть запрос IRQ, и посылает сигнал запроса ядру;
- 2) ядро фиксирует состояние запроса прерывания;
- 3) обработка запроса ядром продолжается в течение нескольких циклов из-за наличия конвейера команд;
- 4) ядро загружает адрес запроса прерывания из VIC.

Существует вероятность, что состояние VIC изменилось во время этапа 3. Например, оно изменилось так, что вызвало последовательность, начинающуюся с шага 1. При этом прежнее прерывание перестает обрабатываться. В этом случае, VIC уже не будет способен идентифицировать прерывание, сгенерировавшее запрос, и в результате VIC возвратит заданный по умолчанию адрес вектора прерывания VICDefVectAddr. Этот потенциально аварийный сценарий может быть предотвращен двумя способами:

– прикладной код должен содержать программные блоки, предотвращающие ложные прерывания, которые могут когда-либо случиться (при этом простой защиты от изменения состояния VIC может оказаться недостаточно, поскольку, например, сбои при обработке прерывания могут также вызвать ложные прерывания);

– обработчик значения по умолчанию для VIC должен быть программно установлен и проверен должным образом.

3.2.3 Рекомендации по программированию VIC

Если пользовательское приложение выполняется во встроенной оперативной памяти и использует прерывания, то векторы прерываний должны быть переотображены на адрес 0x0 Flash-памяти. Это необходимо, поскольку все векторы исключительных состояний (к ним относятся и прерывания) расположены по адресам 0x0 и выше. Этого легко достигнуть путем конфигурирования регистра MEMMAP блока управления системой в режиме пользовательской оперативной памяти. Прикладной код должен быть связан с таблицей векторов прерываний, которая постоянно расположена по адресу 0x40000000.

Несмотря на то, что в качестве генераторов запросов FIQ могут быть выбраны многие источники (VICIntSelect), для обслуживания всех доступных/существующих запросов FIQ должна быть выделена только одна сервисная подпрограмма обработки прерываний. Поэтому в случае, когда более чем один источник прерывания классифицирован как запрос FIQ, подпрограмма обработки прерывания FIQ должна читать регистр VICFIQStatus, чтобы по результатам чтения содержимого выбрать алгоритм обработки запроса прерывания.

Настоятельно рекомендуется классифицировать как FIQ только один источник прерывания, иначе увеличится время ожидания начала обработки прерывания.

После завершения сервисной подпрограммы обработки прерывания, сброс флага прерывания на уровне периферии сведется к изменению состояний некоторых битов в регистрах VICRawIntr, VICFIQStatus и VICIRQStatus. До возвращения из прерывания необходимо выполнить запись в регистр VICVectAddr, прежде чем может быть обслужено следующее прерывание. Эта запись сбрасывает соответствующий флаг прерывания внутренней логики аппаратной обработки прерываний.

Чтобы запретить прерывание в VIC, необходимо сбросить соответствующий бит в регистре VICIntEnClear, который, сбросит связанный с ним бит в регистре VICIntEnable. Производится также обращение к регистрам VICSoftInt и VICSoftIntClear. Единичный бит регистра VICSoftIntClear будет сбрасывать соответствующий ему бит регистра VICSoftInt. Перед новой операцией сброса того же самого бита регистра VICSoftInt, необходимо предварительно задать VICSoftIntClear = 0x00000000. Поскольку запись “1” в любой бит регистров с суффиксом Clear будет иметь одноразовый эффект для соответствующего бита регистра, которым он управляет.

Если разрешены прерывания от сторожевого таймера, по антипереполнению или недействительной последовательности счета, то не существует способа сбросить эти прерывания. Единственный вариант возврата из таких прерываний заключается в запрещении их путём использования VICIntEnClear.

3.3 Блок коммутации выводов

Блок коммутации выводов позволяет производить индивидуальное конфигурирование выводов микроконтроллера для назначения им требуемых функций. Регистры управления блока коммутации выводов управляют мультиплексорами, которые производят соединение выводов микроконтроллера со встроенными периферийными устройствами. Периферийные устройства должны быть подключены к соответствующим выводам ещё до того, как они будут активизированы, а также до разрешения любых связанных с ними прерываний. Активизация любой периферийной функции, которая не подключена к связанному с ней выводу, приведет к неопределённому результату.

Выбор одной из функций для подключения к выводу порта полностью исключает все другие функции, доступные этому выводу. Частичным исключением из этого правила являются входы АЦП. Независимо от функции, выбранной для вывода порта, к которому подключен вход АЦП, этот вход может читаться микроконтроллером в любое время и изменение входного напряжения на нем будет отражено в регистре результата аналого-цифрового преобразования.

Однако корректное чтение результата преобразования может быть произведено, если для данного вывода выбрана функция аналогового входа АЦП, поскольку только тогда будет активизирована соответствующая схема интерфейса между выводом и модулем АЦП. Во всех других случаях, подключение цифро-

вых логических схем, необходимых для активизации этой функции, приведет к некорректному проведению аналого-цифрового преобразования.

Рассмотрим карту регистров блока коммутации выводов на примере микроконтроллера LPC2148. Она приведена ниже в таблице 3.14.

Таблица 3.14 – Регистры блока коммутации выводов LPC2148

Адрес	Название	Описание	Доступ
0xE002 C000	PINSEL0	Регистр выбора функций выводов 0	R/W
0xE002 C004	PINSEL1	Регистр выбора функций выводов 1	R/W
0xE002 C014	PINSEL2	Регистр выбора функций выводов 2	R/W

После выполнения сброса микроконтроллера, значения всех битов в регистрах PINSEL0 и PINSEL1 равно “0”. Значения битов регистра PINSEL2 в этом случае будет определяться тем, используются ли для данного микроконтроллера внутрисистемная отладка или трассировка, или нет.

Регистры PINSEL0 и PINSEL1 управляют функциями выводов микроконтроллера, согласно перечисленным в таблице 3.15 параметрам настройки. Направление передачи данных (ввод/вывод) для каждого вывода определяется состоянием служебного бита выбора направления в регистре IODIR0 лишь в том случае, если для вывода назначена функция линии ввода/вывода общего назначения (GPIO). Если вывод выбран для выполнения других функций, то направление передачи данных устанавливается автоматически соответствующим периферийным устройством.

Таблица 3.15 – Параметры настройка функций выводов микроконтроллера

Значение бита i+1	Значение бита i	Функция вывода	Значение после сброса
0	0	Первичная функция (по умолчанию), как правило, линия ввода/вывода общего назначения	00
0	1	Альтернативная функция 1	
1	0	Альтернативная функция 2	
1	1	Альтернативная функция 3, либо зарезервированная комбинация	

Таким образом, блок коммутации выводов позволяет настраивать исполняемые функции только для неслужебных выводов микроконтроллера (он не может управлять выводами питания, сброса, генератора тактовых сигналов и т.д.). Каждый из неслужебных выводов может иметь до 3-х альтернативных функций. По умолчанию любой такой вывод работает как линия ввода/вывода общего назначения соответствующего порта. Выбор исполняемой функции определится заданным значением пары соответствующих битов.

Битовые структуры регистров PINSEL0, PINSEL1 и PINSEL2 приведены ниже в таблицах 3.16, 3.17 и 3.18 соответственно.

Таблица 3.16 – Битовая структура регистра PINSEL0

Номер битов	Название вывода	Функция при 00	Функция при 01	Функция при 10	Функция при 11
1:0	P0.0	GPIO P0.0	TxD (UART0)	PWM1	Зарезерв.
3:2	P0.1	GPIO P0.1	RxD (UART0)	PWM3	EINT0
5:4	P0.2	GPIO P0.2	SCL0 (I ² C0)	Захват 0.0 (Timer0)	Зарезерв.
7:6	P0.3	GPIO P0.3	SDA0 (I ² C0)	Совпадение 0.0 (Timer0)	EINT1
9:8	P0.4	GPIO P0.4	SCK0 (SPI0)	Захват 0.1 (Timer0)	AD0.6 (ADC)
11:10	P0.5	GPIO P0.5	MISO0 (SPI0)	Совпадение 0.1 (Timer0)	AD0.7 (ADC)
13:12	P0.6	GPIO P0.6	MOSI0 (SPI0)	Захват 0.2 (Timer0)	AD1.0 (ADC)
15:14	P0.7	GPIO P0.7	SSEL0 (SPI0)	PWM2	EINT2
17:16	P0.8	GPIO P0.8	TxD (UART1)	PWM4	AD1.1
19:18	P0.9	GPIO P0.9	RxD (UART1)	PWM6	EINT3
21:20	P0.10	GPIO P0.10	RTS (UART1)	Захват 1.0 (Timer1)	AD1.2 (ADC)
23:22	P0.11	GPIO P0.11	CTS (UART1)	Захват 1.1 (Timer1)	SCL1 (I ² C1)
25:24	P0.12	GPIO P0.12	DSR (UART1)	Совпадение 1.0 (Timer1)	AD1.3 (ADC)
27:26	P0.13	GPIO P0.13	DTR (UART1)	Совпадение 1.1 (Timer1)	AD1.4 (ADC)
29:28	P0.14	GPIO P0.14	DCD (UART1)	EINT1	SDA1 (I ² C1)
31:30	P0.15	GPIO P0.15	RI (UART1)	EINT2	AD1.5 (ADC)

Таблица 3.17 – Битовая структура регистра PINSEL1

Номер битов	Название вывода	Функция при 00	Функция при 01	Функция при 10	Функция при 11
1	2	3	4	5	6
1:0	P0.16	GPIO P0.16	EINT0	Совпадение 0.2 (Timer0)	Захват 0.2 (Timer0)
3:2	P0.17	GPIO P0.17	Захват 1.2 (Timer1)	SCK (SSP)	Совпад. 1.2 (Timer1)
5:4	P0.18	GPIO P0.18	Захват 1.3 (Timer1)	MISO (SSP)	Совпад. 1.3 (Timer1)
7:6	P0.19	GPIO P0.19	Совпадение 1.2 (Timer1)	MOSI (SSP)	Захват 1.2 (Timer1)

Продолжение таблицы 3.17

1	2	3	4	5	6
9:8	P0.20	GPIO P0.20	Совпадение 1.3 (Timer1)	SSEL (SSP)	EINT3
11:10	P0.21	GPIO P0.21	PWM5	AD1.6 (ADC)	Захват 1.3 (Timer1)
13:12	P0.22	GPIO P0.22	AD1.7 (ADC)	Захват 0.0 (Timer0)	Совпад. 0.0 (Timer0)
15:14	P0.23	GPIO P0.23	V _{BUS}	Зарезерв.	Зарезерв.
17:16	P0.24	Зарезервирован			
19:18	P0.25	GPIO P0.25	AD0.4 (ADC)	A _{OUT} (DAC)	Зарезерв.
21:20	P0.26	Зарезервирован			
23:22	P0.27	Зарезервирован			
25:24	P0.28	GPIO P0.28	AIN0.1 (ADC)	Захват 0.2 (Timer0)	Совпад. 0.2 (Timer0)
27:26	P0.29	GPIO P0.29	AIN0.2 (ADC)	Захват 0.3 (Timer0)	Совпад. 0.3 (Timer0)
29:28	P0.30	GPIO P0.30	AIN0.3 (ADC)	EINT3	Захват 0.0 (Timer0)
31:30	P0.31	GPIO P0.31	UP_LED	CONNECT	Зарезерв.

Таблица 3.18 – Битовая структура регистра PINSEL2

Номер бита	Функция бита	Значение после сброса
1:0	Зарезервированы	NA
2	Если бит равен “0”, выводы P1.36:26 используются как выводы GPIO. Если бит равен “1”, выводы P1.31:26 используются как порт отладки	инверсно относительно уровня на данном выводе
3	Если бит равен “0”, выводы P1.25:16 используются как выводы GPIO. Если бит равен “1”, выводы P1.25:16 используются как порт трассировки	инверсно относительно уровня на данном выводе
4:31	Зарезервированы	NA

Следует отметить, что при доступе к регистру PINSEL2 для проведения операции “чтение-модификация-запись”, случайная запись 0 в бит 2 и/или бит 3 приведет к потере и/или возобновлению функции трассировки.

3.4 Управление линиями ввода/вывода общего назначения

Для линий ввода/вывода общего назначения (GPIO – General Purpose Input Output) микроконтроллеров семейства LPC2000 характерны следующие особенности:

- управление направлением передачи данных (ввод или вывод) для каждой линии индивидуально;
- раздельное управление установкой и сбросом состояния линии;

– все линии по умолчанию после сброса настраиваются как входные.

Линии GPIO могут использоваться для следующих приложений:

– ввод/вывод общего назначения;

– управление светодиодами, или другими индикаторами;

– управление другими внешними устройствами, подключенными к микроконтроллеру.

Общее количество доступных линий ввода/вывода общего назначения определяется типом корпуса, в котором реализован конкретный микроконтроллер: 48-выводной, 64-выводной, 144-выводной.

Микроконтроллеры семейства LPC2000 в 48-выводном корпусе имеют 1 32-битный порт ввода/вывода общего назначения (PORT0) с 32 доступными линиями. Микроконтроллеры в 64-выводном корпусе имеют 2 32-битных порта ввода/вывода общего назначения (PORT0 и PORT1). В качестве линий PORT0 доступно 30 из 32 выводов. PORT1 содержит всего лишь 16 выводов, доступных как линии ввода/вывода общего назначения. Микроконтроллеры в 144-выводном корпусе снабжены ещё 2 дополнительными 32-битными портами (PORT2 и PORT3), которые могут быть сконфигурированы для использования в качестве шин данных и адреса внешней памяти или как линии ввода/вывода общего назначения, а также для выполнения дополнительных альтернативных функций.

Таким образом, если пользовательское приложение не задействует никаких альтернативных функций для выводов микроконтроллера LPC2148 в 64-выводном корпусе, то ему доступно 45 линий ввода/вывода общего назначения: 29 линий ввода/вывода порта 0 (все линии, кроме P0.24, P0.26, P0.27) и 16 линий ввода/вывода порта 1 (P1.16 – P1.31).

3.4.1 Регистры управления вводом/выводом общего назначения

Карта регистров, с помощью которых осуществляется управление портами ввода/вывода общего назначения (PORT0 и PORT1) для микроконтроллера LPC2148, отображена в таблице 3.19.

Таблица 3.19 – Регистры управления PORT0 и PORT1 для LPC2148

Адрес	Название	Описание	Доступ
1	2	3	4
0xE002 8000	IOPIN0	Регистр состояния выводов PORT0. Отображает текущие состояния выводов, сконфигурированных как линии ввода/вывода общего назначения, независимо от заданного для них направления и режима работы.	R/W
0xE002 8010	IOPIN1	Регистр состояния выводов PORT1. Назначение аналогично регистру IOPIN0	R/W

Продолжение таблицы 3.19

1	2	3	4
0xE002 8008	IODIR0	Регистр направления выводов PORT0. Задаёт направление (ввод или вывод) для каждой линии порта по отдельности	R/W
0xE002 8018	IODIR1	Регистр направления выводов PORT1. Назначение аналогично регистру IODIR0	R/W
0xE002 8004	IOSET0	Регистр установки выводов PORT0. Позволяет устанавливать в “1” заданные выводы порта.	R/W
0xE002 8014	IOSET1	Регистр установки выводов PORT1. Назначение аналогично регистру IOSET0	R/W
0xE002 800C	IOCLR0	Регистр сброса выводов PORT0. Позволяет устанавливать в “0” заданные выводы порта, а также сбрасывает соответствующие биты в регистре IOSET0.	WO
0xE002 801C	IOCLR1	Регистр сброса выводов PORT1. Назначение аналогично регистру IOSET1	WO

Регистры IOPIN0 и IOPIN1 предназначены для отображения текущего состояния выводов порта. Биты регистров отображают любые изменения внешних сигналов, но только для выводов микроконтроллера, сконфигурированных как линии ввода/вывода общего назначения. При настройке выводов для выполнения альтернативных функций, контроль их состояния с использованием этих регистров не обеспечит достоверного результата. После сброса микроконтроллера значения битов этих регистров являются неопределёнными, что соответствует переводу линий ввода/вывода общего назначения в Z-состояние.

Регистры IODIR0 и IODIR1 предназначены для управления направлением выводов порта, если они сконфигурированы как линии ввода/вывода общего назначения. Запись “1” в заданные биты регистров устанавливает соответствующие выводы как выходные (для передачи данных). Запись “0” устанавливает соответствующие выводы как входные (для приёма данных). После сброса микроконтроллера значения битов этих регистров равны “0”.

Регистры IOSET0 и IOSET1 используются для установки выводов порта, если они сконфигурированы как линии ввода/вывода общего назначения и для них задан режим выходов. Запись “1” в заданные биты регистров вызывает установку соответствующих выводов портов. Запись “0” не имеет эффекта. Если выводы сконфигурированы как входные или для них выбраны альтернативные функции, то изменение состояний соответствующих битов этих регистров не имеет эффекта. Чтение регистра IOSETn возвращает значение, которое определено предыдущими записями в IOSETn и IOCLRn (или IOPINn). Это значение не отображает влияние внешних сигналов на линию ввода/вывода. После сброса микроконтроллера значения битов этих регистров равны “0”.

Регистры IOCLR0 и IOCLR1 используются для сброса выводов порта, если они сконфигурированы как линии ввода/вывода общего назначения и для

них задан режим выходов. Запись “1” в заданные биты регистров вызывает сброс соответствующих выводов портов и сбрасывает соответствующие биты в регистрах IOSET0 и IOSET1. Запись “0” не имеет эффекта. Если выводы сконфигурированы как входные или для них выбраны альтернативные функции, то изменение состояний соответствующих битов этих регистров не имеет эффекта. После сброса микроконтроллера значения битов этих регистров равны “0”.

3.4.2 Рекомендации по использованию линий ввода/вывода общего назначения

Если для некоторого вывода порта соответствующий бит установлен как в регистре установки выводов IOSETn, так и в регистре сброса выводов IOCLRn, то логический уровень на этом выводе будет определяться последней по времени записью в регистры IOSETn и IOCLRn. Это означает, что в случае выполнения последовательности команд:

```
IOSET0 = 0x00000080;
IOCLR0 = 0x00000080;
```

на выводе P0.7 после записи в регистр IOSET0 появится высокий уровень, а после записи в регистр IOCLR0 – низкий.

Приложения пользователя, которые требуют одновременного сброса всех линий порта ввода/вывода общего назначения, могут использовать прямой доступ к соответствующему регистру состояния выводов IOPINn. Предположим, что выводы P0.8 и P0.5 сконфигурированы как выходные, тогда запись в регистр IOPIN0:

```
IOPIN0 = 0x0000C700;
```

произведет тот же самый эффект, что и последовательность команд:

```
IOSET0 = 0x0000C700;
IOCLR0 = 0x00003800;
```

Легко видеть, что вариант с записью в регистры IOSET0 и IOCLR0 будет выполняться дольше, по сравнению с единственной командой записи в регистр IOPIN0.

При написании пользовательских приложений следует особенно внимательно выполнять задание направления работы для используемых линий ввода/вывода общего назначения. Для этого следует предварительно детально разбираться со схемами подключений внешних устройств к выводам микроконтроллера! Следует чётко понимать, что неверное указание направления работы вывода порта микроконтроллера может привести к выходу его из строя! Это может произойти, например, если выходную ножку внешнего устройства соединить с выводом микроконтроллера, ошибочно сконфигурированным как выход.

Аналогичное требование следует выполнять даже в том случае, когда выходы микроконтроллера выполняют любые альтернативные функции, описанные выше!

4 РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА

4.1 Цель и порядок выполнения расчётно-графической работы

Расчётно-графическая работа является одним из видов учебной работы, в процессе выполнения которой для решения поставленных задач студент применяет на практике знания, приобретенные им при изучении теоретических основ учебной дисциплины. В результате этого он закрепляет полученные ранее знания, расширяет их, а также приобретает навыки и умения практического решения типовых задач данной учебной дисциплины.

Темой расчётно-графической работы по дисциплине “Микроконтроллерные системы автоматики и управления” является генерация сигналов заданной формы и параметров при помощи микросхем цифро-аналогового преобразования.

Целью выполнения расчётно-графической работы является приобретение практических навыков генерации сигналов заданной формы и параметров с помощью цифро-аналоговых преобразователей.

Выполнение расчётно-графической работы включает следующие обязательные этапы:

- анализ известных методов построения цифро-аналоговых преобразователей сигналов;
- анализ организации механизма цифро-аналогового преобразования в учебно-отладочном стенде;
- описание варианта задания с учётом соответствующего уровня его выполнения;
- разработка схемы алгоритма генерации сигнала в соответствии с вариантом задания;
- расчёт значений отсчётов, подаваемых на ЦАП, для одного периода сигнала;
- написание кода программы на языке С;
- анализ работоспособности программного кода и оценка соответствия заданным требованиям;
- исправление выявленных ошибок, а также внесение изменений (при необходимости);
- выводы о проделанной работе;
- оформление отчёта о выполнении расчётно-графической работы.

Каждое задание расчётно-графической работы состоит из нескольких уровней. Каждый последующий уровень имеет более высокую сложность и оценивается большим количеством баллов по сравнению с предыдущими.

Выполнение задания более высокого уровня подразумевает обязательное выполнение всех заданий более низких уровней.

Практическая часть расчётно-графической работы выполняется при помощи инструментальных средств, входящих в состав учебно-отладочного стенда NXP LPC2148 Education Board и программных средств – ИСП Keil uVision компании Keil Software. Рекомендуется использовать номера версий, начиная с

3.24 и выше. Также допускается использовать ИСР IAR Embedded Workbench for ARM.

4.2 Требования к отчёту о выполнении расчётно-графической работы

Отчёт о выполнении расчётно-графической работы оформляется на стандартных листах формата А4 в соответствии с требованиями СОККР-2001 кафедры ИКС.

Отчёт о выполнении расчётно-графической работы должен содержать следующие обязательные структурные элементы:

- титульный лист с указанием темы расчётно-графической работы;
- анализ известных методов построения цифро-аналоговых преобразователей сигналов (объёмом не менее 5-ти листов формата А4), раскрывающий суть каждого из методов преобразования, а также особенности внутреннего построения и функционирования ЦАП, построенных по каждому из методов с обязательным отображением структурных схем устройств;
- анализ организации механизма цифро-аналогового преобразования в учебно-отладочном стенде (объёмом не менее 3-х листов формата А4), раскрывающий схему подключения аналоговых выходов и конфигурирование работы встроенного ЦАП;
- описание варианта задания, обязательно содержащее таблицу задания, состоящую из тех уровней, которые были выполнены непосредственно;
- разработка схемы алгоритма генерации сигнала (с необходимым текстовым описанием);
- расчёт значений отсчётов сигнала (содержащий таблицу значений всех отсчётов, подаваемых на ЦАП, для генерации одного периода сигнала, а также формулы для расчёта этих значений на всех различных временных участках сигнала);
- написание кода программы на языке С (программа должна быть представлена с необходимым форматированием и содержать все необходимые для понимания её работы комментарии);
- анализ работоспособности программного кода и оценка соответствия заданным требованиям, содержащий необходимые расчёты и выводы, касательно соблюдения значений для частот преобразования интегрированного ЦАП и выдачи отсчётов сигнала (смотри подраздел 3.3 абзац 3);
- выводы (здесь необходимо произвести подведение итогов о проделанной работе, дать оценку полученных результатов, указать особенности выполнения цифро-аналогового преобразования в учебно-отладочном стенде, выявленные в ходе выполнения работы).

Каждый из структурных элементов нумеруется, как отдельный раздел и начинается с нового листа.

4.3 Требования и рекомендации к выполнению расчётно-графической работы

Уровень I задания к расчётно-графической работе подразумевает выполнение только базовых требований: написание кода программы на языке C, которая осуществляет генерацию сигнала заданной формы и частоты в соответствии с вариантом задания. Обязательным условием выполнения задания этого уровня является использование настраиваемого таймера-счётчика Timer0 для организации временных задержек между выдачами отсчётов сигнала на ЦАП.

Генерация сигнала заданной формы и частоты должна выполняться так называемым "табличным" методом. При этом рекомендуется задавать не менее 100 отсчётов сигнала на один его период. Результат вычисления значения отсчёта сигнала (в соответствии с видом функции, задающей форму сигнала на конкретном участке) необходимо округлять до ближайшего целого числа. В дальнейшем эти вычисленные значения отсчётов следует сохранять в виде массива, значения элементов которого будут последовательно записываться в соответствующий регистр данных ЦАП.

При выполнении любых уровней задания расчётно-графической работы, кроме уровня II, для заданной частоты сигнала (или всех значений, перечисленных в частотном ряду) следует рассчитать частоту выдачи цифровых отсчётов на ЦАП. Затем сравнить это значение с рассчитанной и заданной Вами частотой преобразования интегрированного ЦАП. Соответствующие формулы для расчёта каждой из частот, все необходимые расчёты, а также вывод о результатах сравнения частот (превышает ли частота выдачи цифровых отсчётов сигнала частоту преобразования ЦАП) в этом случае должны быть отображены в разделе анализа работоспособности программного кода и оценка соответствия заданным требованиям.

Проверка выполнения задания расчётно-графической работы осуществляется с помощью осциллографа, который подключается к соответствующим выводам разъёма расширения учебно-отладочного стенда (вывод 42 – выход ЦАП – к сигнальному входу осциллографа; вывод 6 – заземление – ко входу заземления осциллографа). При этом переключки J3 и J30 должны быть установлены в соответствующие положения: J3 – крайняя левая позиция; J30 – крайняя правая позиция.

Прежде чем приступить к проверке выполнения задания следует осуществить настройку осциллографа.

4.4 Варианты заданий

Ниже, в таблицах 4.1 – 4.23, приводятся варианты заданий для выполнения расчётно-графической работы по дисциплине “Микроконтроллерные системы автоматики и управления”. Номер варианта задания выдаётся индивидуально каждому студенту лектором по дисциплине.

Задание №1

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.1 – Задание №1

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Синусоидальная.
Базовая частота	200 Гц.
Уровень II (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень III (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	50 Гц, 100 Гц, 200 Гц, 1000 Гц.

Задание №2

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.2 – Задание №2

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая и линейно спадающая.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.	
Уровень I (0 – 25 баллов)	
Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №3

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.3 – Задание №3

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая и линейно-спадающая.
Базовая частота	200 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	50 Гц, 100 Гц, 200 Гц, 500 Гц.

Задание №4

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.4 – Задание №4

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая, линейно спадающая, постоянная.
Базовая частота	200 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 400 Гц, 800 Гц.

Задание №5

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.5 – Задание №5

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая, линейно спадающая, постоянная.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №6

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.6 – Задание №6

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая, линейно спадающая, постоянная.
Базовая частота	100 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №7

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.7 – Задание №7

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая, линейно спадающая, постоянная.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	50 Гц, 100 Гц, 200 Гц, 500 Гц.

Задание №8

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.8 – Задание №8

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Синусоидальная, линейно возрастающая, линейно спадающая.
Базовая частота	200 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	50 Гц, 100 Гц, 200 Гц, 500 Гц.

Задание №9

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.9 – Задание №9

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Синусоидальная, линейно спадающая.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №10

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.10 – Задание №10

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая, постоянная, линейно спадающая, синусоидальная.
Базовая частота	100 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 400 Гц, 500 Гц.

Задание №11

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.11 – Задание №11

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Постоянная, линейно возрастающая.
Базовая частота	100 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №12

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.12 – Задание №12

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Постоянная, линейно спадающая.
Базовая частота	200 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	50 Гц, 100 Гц, 200 Гц, 500 Гц.

Задание №13

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.13 – Задание №13

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая, синусоидальная.
Базовая частота	400 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 400 Гц, 500 Гц.

Задание №14

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.14 – Задание №14

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая, постоянная, линейно спадающая.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №15

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.15 – Задание №15

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая, линейно спадающая.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 800 Гц.

Задание №16

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.16 – Задание №16

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Постоянная, линейно возрастающая, линейно спадающая.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №17

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.17 – Задание №17

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Постоянная, линейно возрастающая, линейно спадающая.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	50 Гц, 100 Гц, 200 Гц, 500 Гц.

Задание №18

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.18 – Задание №18

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Постоянная, линейно спадающая, линейно возрастающая.
Базовая частота	200 Гц.
Уровень I (0 – 17 баллов)	
Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.	
Уровень I (0 – 25 баллов)	
Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №19

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.19 – Задание №19

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Линейно возрастающая, постоянная, линейно спадающая.
Базовая частота	1000 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №20

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.20 – Задание №20

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Постоянная, линейно возрастающая, линейно спадающая.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №21

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.21 – Задание №21

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Косинусоидальная.
Базовая частота	200 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №22

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.22 – Задание №22

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Постоянная, линейно спадающая.
Базовая частота	100 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 1000 Гц.

Задание №23

Используя учебно-отладочный стенд NXP LPC2148 Education Board, обеспечить генерацию сигнала заданной формы и частоты.

Таблица 4.23 – Задание №23

Уровень I (0 – 12 баллов)	
Форма сигнала	
Функции, задающие форму сигнала	Постоянная, линейно возрастающая.
Базовая частота	500 Гц.
Уровень I (0 – 17 баллов)	
<p>Обеспечить непосредственное отображение частоты сгенерированного сигнала на ЖКИ в следующем формате: "Частота: (X)XXX Гц", где X – соответствующая цифра. Вывод сформированной строки начинается с крайнего левого знакоместа первой (верхней) строки индикатора.</p>	
Уровень I (0 – 25 баллов)	
<p>Обеспечить выбор значения частоты генерируемого сигнала из приведенного ниже частотного ряда при помощи джойстика SW2. При нажатии джойстика вверх частота сигнала повышается (т.е. становится равной следующему большему значению из заданного ряда частот), по нажатию вниз – следующему меньшему значению. При достижении максимального или минимального значения частоты не обрабатывать действия, связанные с повышением или понижением частоты соответственно. Вновь изменённое значение частоты должно отображаться на ЖКИ согласно указанным выше требованиям.</p>	
Частотный ряд	100 Гц, 200 Гц, 500 Гц, 800 Гц.


```

VICIntSelect= 0x00; //all interrupts are IRQs
/* Инициализация Timer0 */
TOMR0= 1500; //Timer match
/*Входная частота сигнала тактирования Timer0 = 15 МГц, по-
этому значение 1500000 соответствует периоду генерации прерывания
в 0,1 с; 15000 - 1 мс; 1500 - 0,1 мс или 100 мкс*/
TOMCR= 0x03; //Interrupt on Match0, reset timer on match
TOPC= 0x01; // Prescaler to 2
TOTC= 0x00; // reset Timer counter
TOTCR= 0x01; // enable Timer
return;
}

int main(void) {
//Разрешаем ЦАП управление выводом P0.25
PINSEL1&= ~((uint32)0x03 << 18);
PINSEL1|=((uint32)0x02 << 18);

/*Задаём режим работы ЦАП (время установления выходного напряжения
и потребляемый ток): если 0 - 1 мкс (700 мкА); если 1 - 2.5 мкс
(350 мкА)*/
DACR&= ~((uint32)0x01 << 16);
InitTimer0();
while(1) {;}
}

```

4.5.2 Пример генерации пилообразного сигнала

В этом примере происходит генерация пилообразного сигнала. Параметры сигнала: частота – 100 Гц, период – 10 мс, количество отсчётов сигнала на период – 100, время между загрузкой в ЦАП двух последовательных отсчётов – 100 мкс, амплитуда – 2,127 В, начальный уровень сигнала – 0 В.

```

#include <lpc214x.h>

typedef unsigned int uint32;
int i= 0;

/*Таблица значений отсчётов кода сигнала, загружаемых в ЦАП: 0 - 0
В, 990 ~ 2,12 В (поскольку Vref = 2.2 В, а V(Aout) = (Vref *
value[i] / 1024)*/
int value[100]= {0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110,
120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240,
250, 260, 270, 280, 290, 300, 310, 320, 330, 340, 350, 360, 370,
380, 390, 400, 410, 420, 430, 440, 450, 460, 470, 480, 490, 500,
510, 520, 530, 540, 550, 560, 570, 580, 590, 600, 610, 620, 630,
640, 650, 660, 670, 680, 690, 700, 710, 720, 730, 740, 750, 760,
770, 780, 790, 800, 810, 820, 830, 840, 850, 860, 870, 880, 890,
900, 910, 920, 930, 940, 950, 960, 970, 980, 990};

//Обработчик прерывания, которое генерируется с помощью Timer0
__irq void Timer0ISR(void) {

```

Методические указания

```
    i++;
    if(i== 100) {
        i= 0;
    }
    /*Загружаем значение i-го отсчёта кода сигнала в ЦАП. Размер
типа int равен 16 бит, разрядность ЦАП - 10 бит => надо из 16-ти
бит получить 10*/
    DACR= ((value[i] & 0x3FF) << 6);
    T0IR= 0x01;
    VICVectAddr= 0;
    return;
}

void InitTimer0(void) {
    /* Инициализация контроллера прерываний */
    VICDefVectAddr= (unsigned int) &Timer0ISR;
    VICIntEnable= 0x10;          //Channel#4 is the Timer0
    VICIntSelect= 0x00;         //all interrupts are IRQs
    /* Инициализация Timer0 */
    T0MR0= 1500; //Timer match
    /*Входная частота сигнала тактирования Timer0 = 15 МГц, по-
этому значение 1500000 соответствует периоду генерации прерывания
в 0,1 с; 1500 - 1 мс; 1500 - 0,1 мс или 100 мкс*/
    T0MCR= 0x03; //Interrupt on Match0, reset timer on match
    T0PC= 0x01; // Prescaler to 2
    T0TC= 0x00; // reset Timer counter
    T0TCR= 0x01; // enable Timer
    return;
}

int main(void) {
    //Разрешаем ЦАП управление выводом P0.25
    PINSEL1&= ~((uint32)0x03 << 18);
    PINSEL1|=((uint32)0x02 << 18);

    /*Задаём режим работы ЦАП (время установления выходного
напряжения и потребляемый ток): если 0 - 1 мкс (700 мкА); если 1 -
2.5 мкс (350 мкА)*/
    DACR&= ~((uint32)0x01 << 16);
    InitTimer0();
    while(1) {;}
}
```

РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

- 1 LPC2148 Education Board User's Guide . Datasheet. EA2-USG-0601 v2.1 Rev B. © Embedded Artists AB 2005 – 2007. – 51 p.
- 2 Muresan Radu. Embedded System Development and Labs for ARM / Radu Muresan. – Embest Inc., 2005. – 291 p.
- 3 Схемотехніка електронних систем: У 3 кн. Кн. 3. Мікропроцесори та мікро-контролери: Підручник/ В.І.Бойко, А.М.Гуржій, В.Я.Жуйков та ін. – 2-ге вид., допов. і переробл. К.: Вища шк., 2004. – 399 с.
- 4 Васильев А.Е. Микроконтроллеры. Разработка встраиваемых приложений. – СПб.: БХВ-Петербург, 2008. – 304 с.: ил.
- 5 Мартин Т. Микроконтроллеры ARM7. Семейство LPC2000 компании Philips. Вводный курс/Пер. с англ. – М.: Издательский дом «Додэка-XXI», 2006. – 240 с.: ил.
- 6 Редькин П.П. Микроконтроллеры ARM7 семейства LPC2000. Руководство пользователя. – М.: Издательский дом «Додэка-XXI», 2007. – 560 с.: ил.
- 7 Швец В.А., Шестакова В.В., Бурцева Н.В., Мелешко Т.В. Одноплатные микроконтроллеры. Проектирование и применение. – К.: «МК-Пресс», 2005. – 304 с.: ил.