Московский государственный технический университет
имени Н.Э. Баумана

Методические указания

А.Ю. Попов

Проектирование радиоэлектронной аппаратуры на основе микроконтроллеров ARM7TDMI

2014

Работа №1. Разработка радиоэлектронной аппаратуры на основе микроконтроллеров ARM7 TDMI в интегрированной среде Keil uVISION

Цель работы – изучение архитектуры микроконтроллеров ARM7 TDMI и средств проектирования и отладки цифровых устройств на их основе.
В ходе работы студенту необходимо ознакомиться с теоретическим материалом, касающимся архитектуры и особенностей функционирования микроконтроллеров с ядром ARM7 TDMI, ознакомиться с возможностями интегрированной среды разработки Keil uVision, разработать и отладить простейшую программу функционирования микроконтроллера NXP LPC2368.

Описание микроконтроллеров семейства ARM7TDMI фирмы Philips.

Микроконтроллеры с ядром ARM7TDMI являются современными системами на кристалле, сочетающими такие достоинства, как:

- 32-х разрядная архитектура ядра с сокращенным набором команд (Reduce Instruction Set Computer, RISC).
- Поддержка двух наборов команд (32-х разрядный полный набор ARM и 16-разрядный упрощенный набор THUMB).
- Высокая производительность ядра благодаря сбалансированности фаз конвейера.
- Настраиваемая система прерываний.
- Широкий спектр микроконтроллеров семейства, отличающихся по составу периферии, частоте работы, стоимости, типам корпусов.

Микроконтроллеры с ядром ARM7TDMI выпускаются многими фирмами-производителями и применяются разработчиками при решении широкого круга задач: от реализации центральных устройств управления системами сбора и обработки информации до контроллеров периферии и внешних интерфейсов. Наиболее популярны такие ARM7TDMI микроконтроллеры, как: серия ADUC702x Analog Devices, серия STR71x фирмы ST Microelectronics, отличающиеся сбалансированным набором периферийных модулей и малыми размерами корпусов, а также микроконтроллеры серии AT91x фирмы Atmel и серия LPC2xxx фирмы NXP (Philips) с расширенной функциональностью.

Рассмотрим архитектуру микроконтроллеров семейства ARM7TDMI на

примере микроконтроллера NXP LPC2368. Упрощенная схема микроконтроллера показана на рисунке 1.

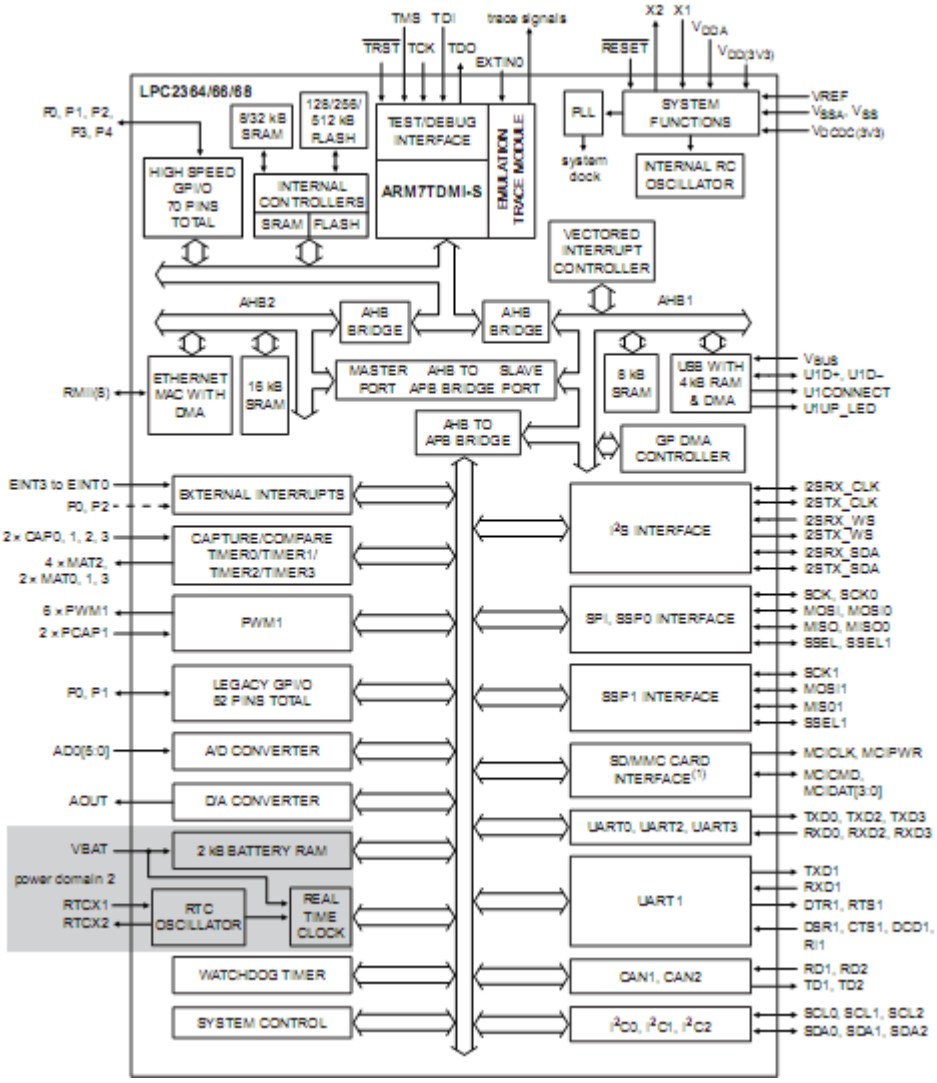

Рисунок 1 — Архитектура микроконтроллера NXP LPC2368

Микроконтроллер состоит из следующих подсистем:

- Микропроцессорное ядро ARM7TDMI, выполняющее вычислительные функции, а также управляющее работой многочисленных периферийных устройств.
- Внутренняя память типа SRAM для хранения результатов вычислений и FLASH-память для долговременного хранения как данных, так и программ функционирования микроконтроллера.
- Иерархическая система шин и мостов, объединяющих модули в единую систему на кристалле. Обмен информацией между микропроцессорным

ядром и памятью осуществляется по локальной шине ARM7. Шина AHB служит для подключения быстродействующих периферийных устройств (Ethernet, USB), контроллера внешней памяти, а также векторного контроллера прерываний (VIC). Периферийная шина APB работает на меньшей частоте и служит для подключения остальных периферийных модулей.

- Системные сервисные блоки, обеспечивающие работу системы. К ним относятся: встроенный осциллятор и модуль фазовой автоподстойки частоты (PLL) для формирования тактовых сигналов системы, модуль управления электропитанием, модуль прямого доступа в память (GPDMA), векторный контроллер прерываний (VIC). Для отладки программ функционирования микроконтроллера реализованы модуль пошаговой трассировки и интерфейс JTAG.
- Периферийные модули общего назначения, состав которых для разных моделей микроконтроллеров может различаться. В микроконтроллер NXP LPC2368 входят: порты ввода/вывода (GPIO), модуль внешних прерываний, четыре таймера с функциями захвата и совпадения (Timer0, Timer1, Timer2, Timer3), два модуля широтно-импульсной модуляции (PWM0, PWM1), 10-разрядные модули АЦП и ЦАП, контроллер ЖК-дисплея, часы реального времени, сторожевой таймер, контроллеры последовательных интерфейсов Ethernet, USB2.0, UART, CAN, SPI, SSP, I2C, I2S.

Как упоминалось ранее, микроконтроллеры ARM7TDMI поддерживают два набора команд (ARM и THUMB). Набор THUMB удобен для создания компактных программ, так как позволяет сократить размер исполняемого кода примерно на 30%, однако приводит к некоторому замедлению обработки (примерно на 40%). При этом возможно выполнять быстрое переключение между двумя наборами по команде BX (см. приложение 3). При возникновении исключительной ситуации процессор автоматически переключается к обработке ARM команд.

При написании программ на языке С возможно объявить ARM функцию (атрибут __arm) или THUMB функцию (атрибут __thumb), что приводит к добавлению инструкции переключения набора и генерации соответствующего кода функции. Например:

```
/* ARM функция */
void MyARMfunc (void) __arm  {
  c = a*b;
}
/* THUMB функция */
void MyTHUMBfunc (void) __thumb  {
  f = d*e;
}
```

Большинство команд ARM набора являются трех-адресными и условными. Это означает, что помимо кода операции в команде указывается дополнительно 4 бита, соответствующие сочетанию флагов регистра состояния программы (рисунок 2). Команда выполняется только в том случае, если флаги соответствуют заданному в команде условию. Дополнительно в команде присутствуют бит S, указывающий на необходимость модификации флагов после выполнения команды. Например по команде сложения с кодом E0810000: ADD R0,R1,R0 будет выполнено сложение регистров R0 и R1, а результат будет помещен в регистр R0. Первая тетрада (E) указывает на условие выполнения AL (выполнять всегда).

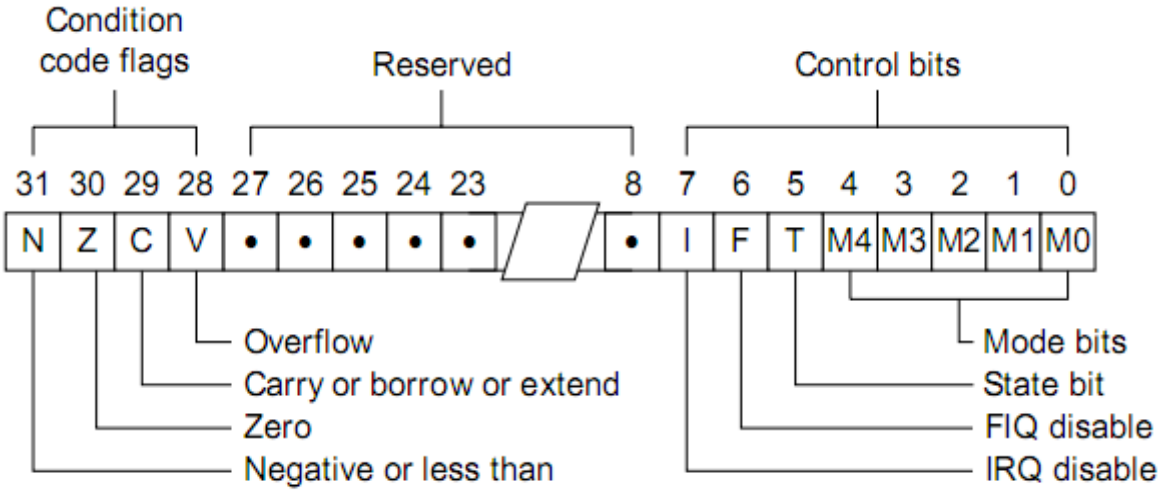

Рисунок 2 — Регистр состояния программы.

Команды THUMB набора являются 16-разрядными и не могут быть условными, а количество операндов в таких командах ограничено двумя.

Помимо обработки двух наборов команд, процессор может работать в одном из семи режимов:

- *User mode:* пользовательский режим, в котором функционируют большинство приложений.
- *Fast interrupt mode (FIQ):* режим обработки быстрых прерываний для обработки наиболее критичных к времени реакции системы участков кода.
- *Interrupt mode (IRQ):* режим обработки обычных прерываний.
- *Supervisor mode:* защищенный режим для поддержки многозадачных операционных систем.
- *Abort mode:* режим обработки сбоев при предвыборке команд или данных.
- *System mode:* режим приложений операционной системы.
- *Undefined mode:* режим неизвестной операции.

Программная модель микропроцессора ARM7 состоит из 37 регистров (рисунок 3): 31 регистра общего назначения и 6 регистров статуса. В зависимости от режима в каждый момент времени доступно различное количество регистров. Некоторые регистры повторены для хранения режимно-зависимых данных (регистры R13, R14, сохраненный регистр статуса SPSR). В THUMB командах регистры r8 – r12 не используются.

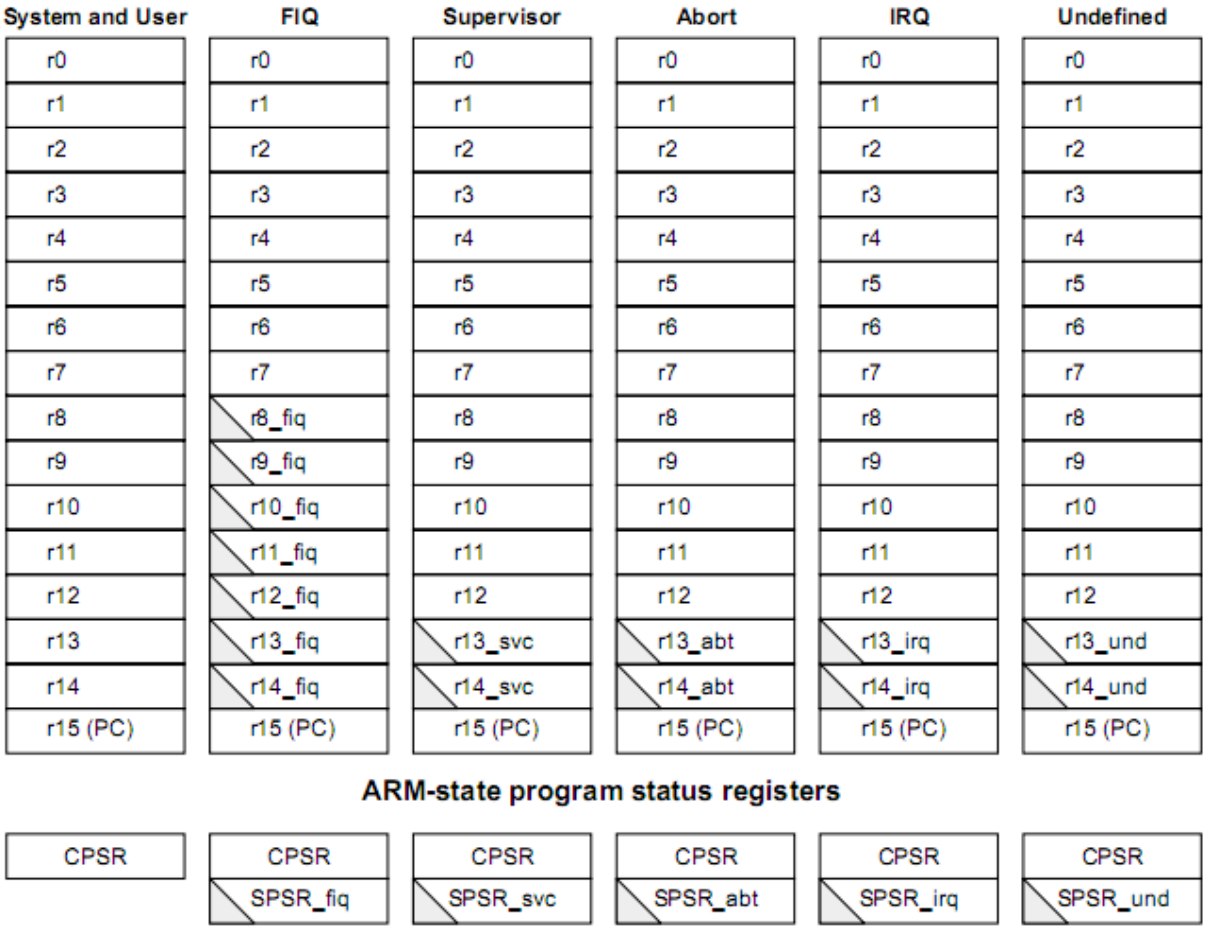

Рисунок 3 — Регистры микропроцессора ARM7TDMI

Доступ ко всем внутренним ресурсам микроконтроллера выполняется через 4ГБ совмещенное адресное пространство. Для управления работой модулей, чтения или записи данных необходимо знать состав, назначение и адреса их программно-доступных регистров, а также некоторые сведения о логике их работы. По мере ознакомления с периферийными модулями микроконтроллера эти сведения будут приведены в методических указаниях. В таблице 1 приведена карта памяти микроконтроллера NXP LPC2368. Более подробное описание регистров модулей ввода/вывода приведено в приложении 1.

Таблица 1. Карта памяти микроконтроллера NXP LPC2368

Диапазон адресов	Назначение	Подадрес	Модуль
0x00000000 - 0x3FFFFFFF	Внутренняя FLASH и высокоскоростные каналы ввода/вывода	0x00000000 - 0x0007FFFF	FLASH-память (512 КБ)
		0x3FFFC000 - 0x3FFFFFFF	Регистры высокоскоростных каналов ввода/вывода
0x4000 0000 - 0x7FFF FFFF	Внутреннее ОЗУ	0x40000000 0x4000FFFF	- 64 КБ ОЗУ микропроцессора
		0x7FE00000 0x7FE03FFF	- ОЗУ модуля Ethernet (16 КБ)
		0x7FD00000 0x7FD03FFF	- ОЗУ модуля USB (16 КБ)
0x80000000 - 0xDFFFFFFF	Внешняя память (16 МБ)	0x80000000 0x80FFFFFF	- Банк 0
		0x81000000 0x81FFFFFF	- Банк 1
0xA0000000 - 0xDFFFFFFF	Не задействовано		
xE0000000-0x EFFFFFFF	Периферийные модули, подключенные к шине APB	см. Приложение 1	36 периферийных модулей
0xF000 0000 - 0xFFFF FFFF	Модули, подключенные к шине AHB	см. Приложение 2	6 модулей

При написании программ удобно использовать не числовые значения адресов программно доступных регистров, а их имена, определенные в специальном файле. Например, для обращения к регистрам микроконтроллера LPC2368 достаточно подключить заголовочный файл LPC23xx.H.

Управление портами ввода/вывода микроконтроллера NXP LPC2368.

С целью сокращения количества выводов микроконтроллеров при сохранении их функциональности большинство контактов микросхемы подключается к нескольким внутренним модулям. Выбор конкретного модуля, который подключается к контакту микросхемы, определяется с помощью программно доступного модуля управления выводами (Pin Connect Block). В каждый момент времени только один модуль может быть подключен к

контактам микросхемы. Разработчик системы выбирает тот тип микроконтроллера, который обеспечивает одновременное подключение всех необходимых ему модулей. При включении питания все выводы микроконтроллера оказываются подключенными к модулю портов ввода/вывода общего назначения (GPIO). Далее, в случае необходимости, должна быть произведена перекоммутация модулей. Это необходимо сделать до того момента, когда коммутируемый модуль будет включен разработчиком и начнет работу.

Все управляемые входы микросхемы принято называть по подключенным к ним разрядам 32-х разрядных портов ввода/вывода, которых в микроконтроллере LPC2368 пять. Например: Port0.1 означает контакт микросхемы, к которому подключен вход №1 порта 0 модуля GPIO. Этот же вход может использоваться модулями UART1, UART3 и I2C1. Для конфигурации входов/выходов необходимо указать нужное в соответствующем регистре выбора функций (PINSEL0 для контактов 0..15 порта 0, PINSEL1 для контактов 16..31 порта 0 и т.д.) В таблице 2 приведены значения для разрядов регистра PINSEL0, обеспечивающие коммутацию модулей с контактами 0..15 порта 0.

Таблица 2 — Назначение бит регистра PINSEL0.

Биты	**Контакт**	**Выбор функции**				**По умолчанию**
		00	**01**	**10**	**11**	
01:00	P0[0]	GPIO Port 0.0	RD1	TXD3	SDA1	00
03:02	P0[1]	GPIO Port 0.1	TD1	RXD3	SCL1	00
05:04	P0[2]	GPIO Port 0.2	TXD0	Не исп.	Не исп.	00
07:06	P0[3]	GPIO Port 0.3	RXD0	Не исп.	Не исп.	00
09:08	P0[4]	GPIO Port 0.4	I2SRX_CLK/LCDVD[0]	RD2	CAP2[0]	00
11:10	P0[5]	GPIO Port 0.5	I2SRX_WS/LCDVD[1]	TD2	CAP2[1]	00
13:12	P0[6]	GPIO Port 0.6	I2SRX_SDA/LCDVD[8]	SSEL1	MAT2[0]	00
15:14	P0[7]	GPIO Port 0.7	I2STX_CLK/LCDVD[9]	SCK1	MAT2[1]	00
17:16	P0[8]	GPIO Port 0.8	I2STX_WS/LCDVD[16]	MISO1	MAT2[2]	00
19:18	P0[9]	GPIO Port 0.9	I2STX_SDA/LCDVD[17]	MOSI1	MAT2[3]	00
21:20	P0[10]	GPIO Port 0.10	TXD2	SDA2	MAT3[0]	00
23:22	P0[11]	GPIO Port 0.11	RXD2	SCL2	MAT3[1]	00
25:24	P0[12]	GPIO Port 0.12	USB_PPWR2	MISO1	AD0[6]	00
27:26	P0[13]	GPIO Port 0.13	USB_UP_LED2	MOSI1	AD0[7]	00
29:28	P0[14]	GPIO Port 0.14	USB_HSTEN2	USB_CONN ECT2	SSEL1	00
31:30	P0[15]	GPIO Port 0.15	TXD1	SCK0	SCK	00

Модуль портов ввода/вывода общего назначения (GPIO) позволяет управлять состоянием контактов микросхемы (в режим «вывод») или фиксировать их состояние для дальнейшего использования в программе (в режиме «ввод»). Модуль позволяет задавать режим работы для каждого разряда порта с помощью 32-разрядных регистров направления IODIRx (регистр IODIR0 для порта 0, регистр IODIR1 для порта 1 и т.д.). Запись нуля в какой-либо разряд регистра IODIRx конфигурирует соответствующий разряд порта на режим «ввод», а запись единицы на режим «вывод».

В режиме «вывод» используются регистр сброса порта IOCLRx и регистр установки порта IOSETx, изменяющие состояние выводов порта при записи единичных значений. Например, при записи в регистр IOCLR0 значения 0x00000002 уровень сигнала на контакте микросхемы Port0.1 будет установлен в состояние логического нуля (при условии правильной настройки модуля управления выводами и регистра IODIR0). Остальные выходы не изменят своего состояния. При записи значения 0x00000003 в регистр IOSET0 уровень сигнала на контактах микросхемы Port0.1 и Port0.0 будет установлен в состояние логической единицы. Остальные выходы также не изменят своего состояния.

В режиме «ввод» состояние контакта микросхемы сохраняется в каждом такте шины APB и может быть прочитано из регистра IOPINx.

Пример 1. Простая программа управления портами ввода/вывода.

```
/* Программа управляет восемью светодиодами, подключенными к выходам 0..7
Порта 0. В зависимости от состояния входа 8 Порта 0 светодиоды зажигаются
справа-налево или слева-направо */

#include <LPC23xx.H>                        /* Описание LPC24xx */
void delay(void) {
      unsigned int i;
      for (i=0;i<0xfffff;i++){}
}
int main (void) {
  unsigned int n;
//Конфигурировать функции входов/выходов порта 0 на модуль GPIO
  PINSEL0   = 0x00000000;
//IODIR0 - Регистр направления ввода вывода (1 - вывод; 0 - ввод)
  IODIR0 = 0x00038000; /* P0.15..17 программируем на вывод, остальные на ввод */
//IOSET0 - Регистр установки порта (1 - установка; 0 - нет изменений)
  IOSET0 = 0x00038000; /* Устанавливаем высокий уровень на выходах (гасим
светодиоды) */

  while (1) {          /* Бесконечный цикл */
      //Если PORT0.10=0 то влево, иначе вправо
    if (IOPIN0 & 0x400) {
      for (n = 0x00008000; n <= 0x00020000; n <<= 1) {
            //Бегущая единица
            //IOCLR0 - Регистр сброса порта (1 - сброс; 0 - нет изменения)
```

```
                  IOCLR0 = n;           /* Сбросить порт */
                  delay();              /* Задержка */
                  /* То же, что IOSET0 - Установить состояние порта */
                  *((volatile unsigned long *) 0xE0028004) = 0x00038000;
            }
    }
      else {
      for (n = 0x00020000; n >= 0x00008000; n >>= 1) {
            //Бегущая единица
                  IOCLR0 = n;           /* Сбросить порт */
                  delay();              /* Задержка */
                  /* Установить состояние порта */
                  IOSET0 = 0x00038000;
            }
      }
  }
}
```

Описание интегрированной среды разработки Keil uVision

Интегрированная среда Keil uVision3 фирмы Keil Elektronik предоставляет пользователю набор средств для написания и отладки кода программ для микроконтроллеров семейств ARM7, ARM9, Cortex M3 и других. В бесплатный дистрибутив входят следующие средства:

- интегрированная среда разработки;
- C/C++ компилятор RealView;
- Макроассемблер и линковщик RealView;
- Дополнительные утилиты RealView;
- Библиотека RTX Real Time Kernel
- Дебаггер uVision.

Ознакомительная версия, ограниченная по объему кода, доступна для скачивания по адресу: https://www.keil.com/demo/eval/arm.htm.

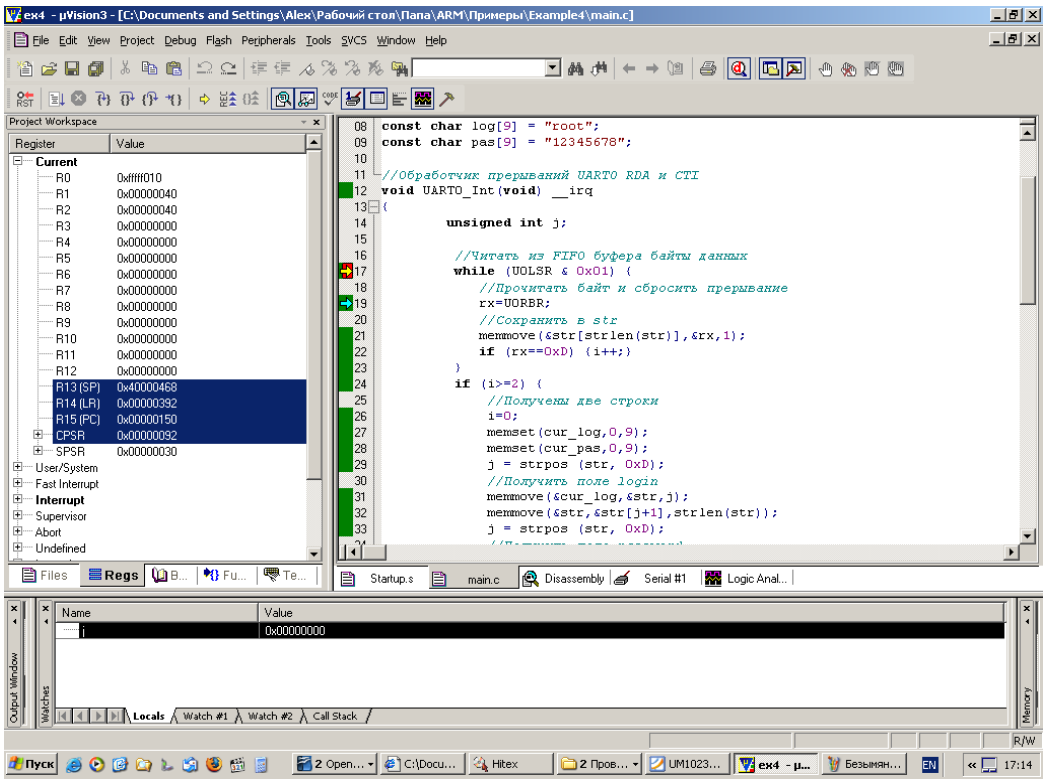

Рисунок 4 - интегрированной среды разработки Keil uVision

Создание проекта

Разработка программы начинается с создания нового проекта. Для этого в меню Project выбрать пункт New uVision Project. Далее указывается расположение и название проекта, модель микроконтроллера (например, NXP(founded by Philips) → LPC2368). После этого система предложит включить в проект стартовый код, содержащий команды инициализации. Данный код содержит таблицу векторов прерываний, код инициализации стека для различных режимов работы микропроцессорного ядра, код инициализации системы синхронизации и памяти.

Для управления списком файлов проекта следует вызвать диалог управления Project → Manage → Components. Диалог управления позволяет также настроить пути к библиотекам, выбрать компилятор, создать список справочников проекта.

Рабочее поле проекта показано на рисунке 4. Его основными элементами являются: редактор исходных описаний; окно управления рабочим полем, меню быстрого запуска; окно консоли.

Компиляция проекта.

Для правильной компиляции простых проектов следует отметить опцию

Project → Options for Target → Linker → Use Memory Layout from Target Dialog.

Отладка проекта в режиме симуляции.

Отладка проекта может выполняться как в режиме симуляции (на программной модели микроконтроллера), так и в режиме аппаратной отладки (с использованием микроконтроллера и специальных аппаратных средств). Для выбора режима используется диалог «Настройки проекта», вкладка Debug (Project → Options for Target → Debug), изображенный на рисунке 5. Следует выбрать пункт «Use Simulator».

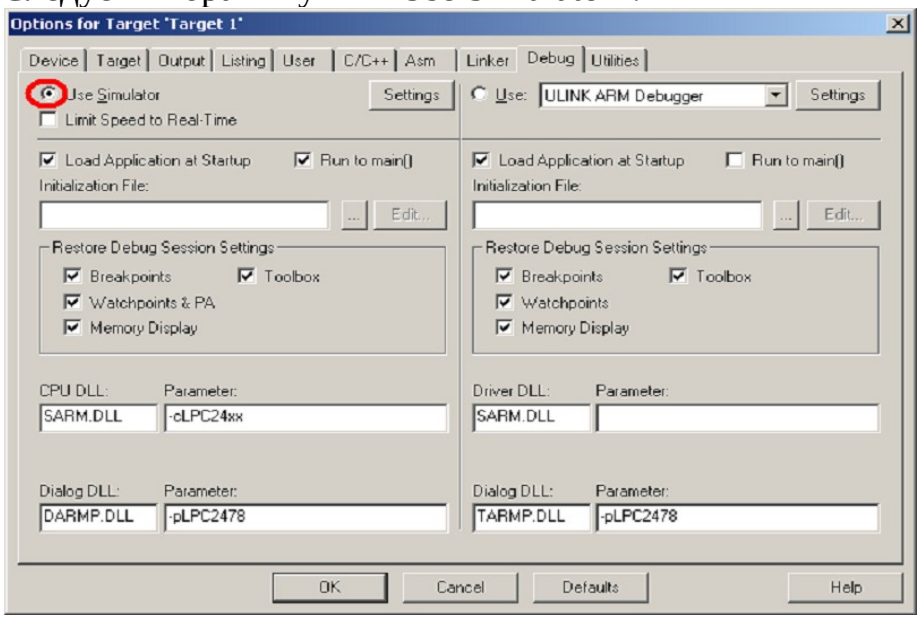

Рисунок 5 - Диалог «Настройки проекта»

После этого можно приступить к симуляции, для чего в меню «Debug» следует выбрать пункт «Start/Stop Debug Session» или использовать сочетание клавиш Ctrl+F5.

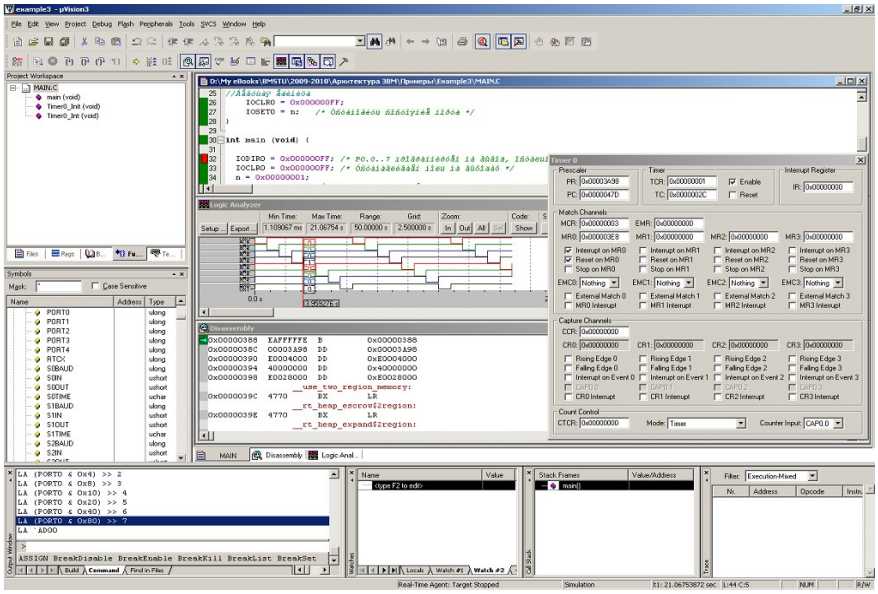

Рисунок 6 — Окно Keil uVision в режиме симуляции.

В режиме симуляции пользователю доступно большое количество средств отладки:

- Окна исходного кода, позволяющие назначать точки останова на уровне исходного кода (breakpoints).
- Окно дизассемблера, позволяющие назначать точки останова на уровне ассемблера.
- Логический анализатор (Logic Analyzer), позволяющий визуализировать изменение внешних сигналов и состояния модулей микроконтроллера.
- Окна визуализации состояния периферийных устройств.
- Окно навигации проекта, включающее вкладки: файлы проекта (Files); регистры ядра (Regs); книги и документация (Books); функции (Functions); типовые конструкции (Templates).
- Окно сигналов и переменных (Symbols), позволяющее просмотреть текущее состояние микроконтроллера, периферии, сигналов, а также упрощающее указание отображаемых сигналов логическому анализатору (для этого следует перенести сигнал или переменную из окна Symbols в окно в окно Logic Analyzer).
- Окно консоли (Output window).
- Окно стека (Call stack)
- Окно переменных (Watches).
- Окно трассировки (Trace).

Практическая часть

Задание 1. Ознакомиться с теоретическим материалом на стр. 2-13.

Задание 2. Создать проект С программы в среде Keil uVision для микроконтроллера NXP LPC2368 с частотой генератора 12 МГц. В проект должны входить файлы: начальной настройки микроконртоллера LPC2300.s и главный файл приложения Main.c.

Задание 3. Разработать и отладить в симуляторе программу функционирования микроконтроллера в соответствии с индивидуальным вариантом. В программе задействовать пины 26-29 порта 1 модуля GPIO.

Задание 4. Разработать функцию управления входными портами микроконтроллера и записать ее в файл ini. Текст функции управления занести в отчет. С использованием функции управления получить осциллограмму работы микроконтроллера для задействованных в проекте сигналов порта 0. Выполнить пошаговую трассировку программы. Осциллограмму и код программы занести в отчет.

Требования к отчету

Отчет по работе должен содержать: задание, листинг программы функционирования микроконтроллера, текст функции управления, осциллограмму, результаты тестирования программы, выводы о работоспособности программы.

Контрольные вопросы

1. Назовите подсистемы, из которых состоит микроконтроллер NXP LPC2368.
2. В чем отличие между наборами командами THUMB и ARM.
3. Перечислите внутренние шины микроконтроллера и опишите их назначение.
4. Перечислите регистры модуля GPIO и их назначение.

Приложение 1. Карта памяти периферийных модулей, подключенных к шине APB

Номер модуля	Базовый адрес	Название модуля
0	0xE000 0000	Watchdog Timer
1	0xE000 4000	Timer 0
2	0xE000 8000	Timer 1
3	0xE000 C000	UART0
4	0xE001 0000	UART1
5	0xE001 4000	PWM0
6	0xE001 8000	PWM1
7	0xE001 C000	I2C0
8	0xE002 0000	SPI
9	0xE002 4000	RTC
10	0xE002 8000	GPIO
11	0xE002 C000	Pin Connect Block
12	0xE003 0000	SSP1
13	0xE003 4000	ADC
14	0xE003 8000	CAN Acceptance Filter RAM
15	0xE003 C000	CAN Acceptance Filter Registers
16	0xE004 0000	CAN Common Registers
17	0xE004 4000	CAN Controller 1
18	0xE004 8000	CAN Controller 2
19 - 22	0xE004 C000 - 0xE005 8000	Не используется
23	0xE005 C000	I2C1
24	0xE006 0000	Не используется
25	0xE006 4000	Не используется
26	0xE006 8000	SSP0
27	0xE006 C000	DAC
28	0xE007 0000	Timer 2
29	0xE007 4000	Timer 3
30	0xE007 8000	UART2
31	0xE007 C000	UART3
32	0xE008 0000	I2C2
33	0xE008 4000	Battery RAM
34	0xE008 8000	I2S
35	0xE008 C000	SD/MMC Card Interface

36 - 126	0xE009 0000 - 0xE01F BFFF	Не используется
127	0xE01F C000	System Control Block

Приложение 2. Карта памяти периферийных модулей, подключенных к шине AHB

Номер модуля	Базовый адрес	Название модуля
0	0xFFE0 0000	Ethernet
1	0xFFE0 4000	GP DMA
2	0xFFE0 8000	Externel Memory Controller
3	0xFFE0 C000	USB Controller
4	0xFFE1 0000	LCD Controller
5	0xFFFF F000	Vectored Interrupt Controller

Приложение 3. Регистры PCLKSEL0 и PCLKSEL1.

PCLKSEL0		
Разряды	Описание модуля	По умолчанию
1:0	WDT	00
3:2	TIMER	00
5:4	TIMER	00
7:6	UART0	00
9:8	UART1	00
11:10	PWM0	00
13:12	PWM1	00
15:14	I2C0	00
17:16	SPI	00
19:18	RTC	00
21:20	SSP1	00
23:22	DAC	00
25:24	ADC	00
27:26	CAN1	00
29:28	CAN2	00
31:30	CAN фильтр	00
PCLKSEL1		
1:0	BAT_RAM	00
3:2	GPIO	00
5:4	Pin Connect block	00
7:6	I2C1	00
9:8	Не используется	00
11:10	SSP0	00
13:12	TIMER3	00
17:16	UART2	00
19:18	UART3	00
21:20	I2C2	00
23:22	I2S	00
25:24	MCI	00
27:26	Не используется	00

29:28	System Control block	00
31:30	Не используется	00