

Современные 32-разрядные ARM-микроконтроллеры серии STM32: организация памяти

Олег Вальпа (г. Миасс, Челябинская обл.)

В статье приводится описание организации памяти 32-разрядных ARM-микроконтроллеров серии STM32 компании STMicroelectronics. Рассматриваются состав, архитектура и особенности памяти микроконтроллера, а также освещаются некоторые нюансы программирования.

ВВЕДЕНИЕ

Структура памяти любого микроконтроллера является важной областью при его изучении. Каждый микроконтроллер может иметь определённые особенности в организации памяти. Понимание этих особенностей позволяет успешно программировать и избегать ошибок. Сделаем неболь-

шой экскурс в память микроконтроллера STM32 [1], чтобы легче и лучше было понимать его работу.

ОРГАНИЗАЦИЯ ПАМЯТИ МИКРОКОНТРОЛЛЕРА

Адресное пространство микроконтроллера STM32 организовано в виде линейного пространства. Размер это-

го пространства определяется 32-разрядной шиной и может иметь предельное значение $2^{32} = 4$ Гбайт. Данный размер памяти очень велик и практически используется лишь частично. Неиспользуемые области памяти зарезервированы для новых моделей микроконтроллеров. Наглядное представление карты памяти STM32 представлено на рисунке 1.

Всё адресное пространство микроконтроллера STM32 разделено на 8 блоков по 512 Мбайт, т.е. с интервалом в $0x20000000$ байт.

Память программ начинается с адреса $0x00000000$. В этой области хранятся коды программ пользователя и системная информация. Оперативная память STM32 имеет начальный адрес $0x20000000$. Регистры функциональных блоков и устройств ввода-вывода микроконтроллера также отображены в памяти, начиная с адреса $0x40000000$. Специальные регистры микроконтроллера находятся в памяти, начиная с адреса $0xE0000000$.

Байты 32-разрядных слов организованы в памяти в порядке возрастания. Байт слова с меньшим номером считается самым младшим байтом этого слова, а байт с большим номером – самым старшим.

Блок начальной загрузки

Первые 2 Кбайт памяти, в зависимости от состояния выводов управления загрузкой, могут быть связаны с флэш-памятью, системной или оперативной памятью.

Микроконтроллеры STM32 имеют 3 различных режима начальной загрузки, которые могут быть выбраны с помощью выводов BOOT0 и BOOT1, как показано в таблице.

Чтобы получить требуемый режим начальной загрузки (см. таблицу), необходимо установить с помощью переключки определённое состояние входов BOOT0 и BOOT1, которое считывается микроконтроллером после сброса.

Кроме того, микроконтроллер считывает состояние входов BOOT0 и BOOT1

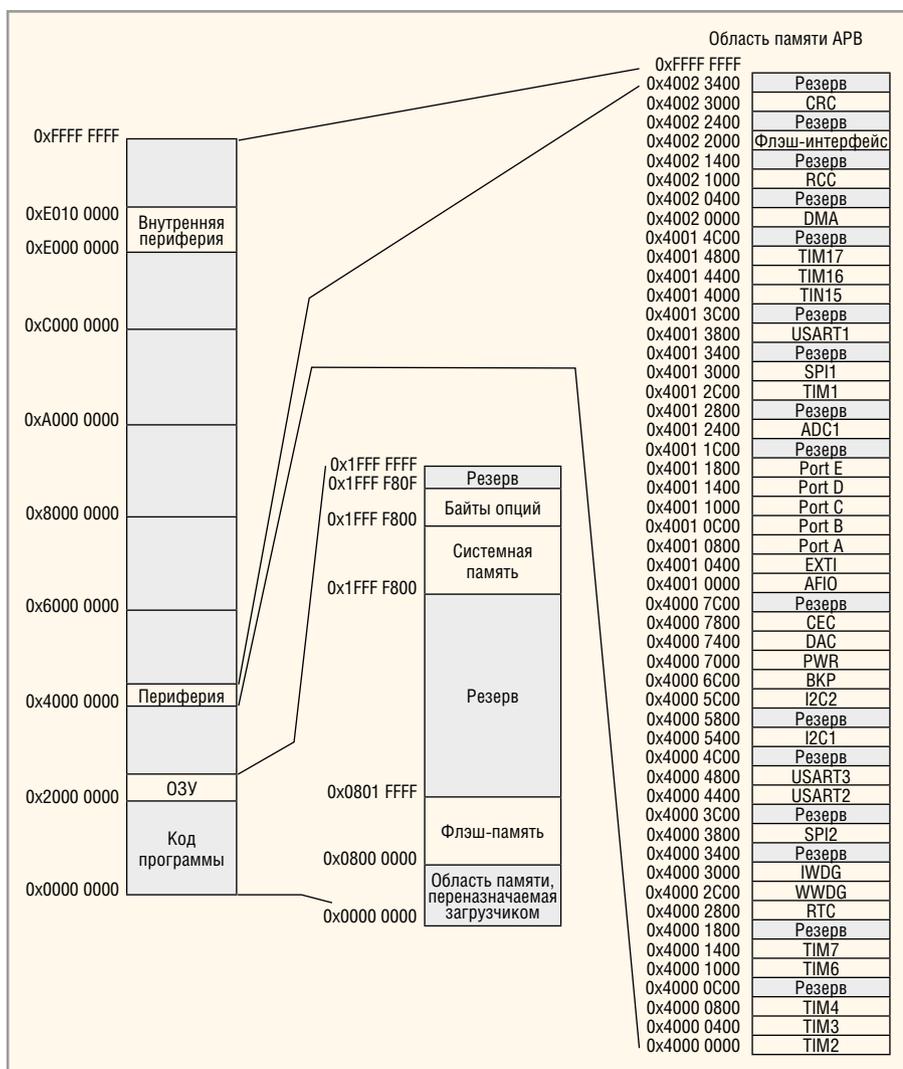


Рис. 1. Карта памяти микроконтроллера STM32

и при выходе из режима ожидания. Поэтому в режиме ожидания выставленные уровни должны удерживаться в состоянии, необходимом для режима загрузки.

Как правило, состояние входов BOOT0 и BOOT1 изменяют перед программированием микроконтроллера для активации внутреннего загрузчика. Затем состояние этих входов переопределяют на чтение памяти программ и не изменяют в течение всего времени работы.

Встроенный загрузчик расположен в системной памяти и программируется фирмой ST при изготовлении продукции. Он используется для перепрограммирования флэш-памяти с помощью одного из доступных последовательных интерфейсов.

Метод битового объединения

Обычно битовые операции в оперативной памяти микроконтроллеров выполняются с помощью инструкций AND и OR. Для этого необходимо выполнить последовательность таких операций, как чтение, модификация и запись. Использование этого метода требует несколько циклов на выполнение установки или сброса отдельных бит и увеличивает размер программы.

Устранить эти недостатки можно введением новых инструкций для сброса и установки бит, но это приводит к усложнению процессора. В STM32 применён способ, называемый битовым объединением (Bit Banding), который позволяет напрямую воздействовать на биты памяти, не используя при этом новых инструкций. Это реализовано с помощью специальных областей памяти, позволяющих выполнять побитовую адресацию.

Бит-адресуемые области карты памяти STM32 разделены на две части. Первая часть служит для хранения бит и в неё входит до 1 Мбайт физической памяти или регистров устройств ввода-вывода. Вторая часть служит для доступа к битам и занимает до 32 Мбайт памяти. При выполнении операций с областью доступа происходит обращение к фактическим адресам бит в памяти.

Метод битового объединения рассмотрим на примере работы с регистрами портов ввода-вывода GPIO. Для вывода данных в них используется регистр ODR, запись в который производится через операции «чтение-модификация-запись» с использованием 16-раз-

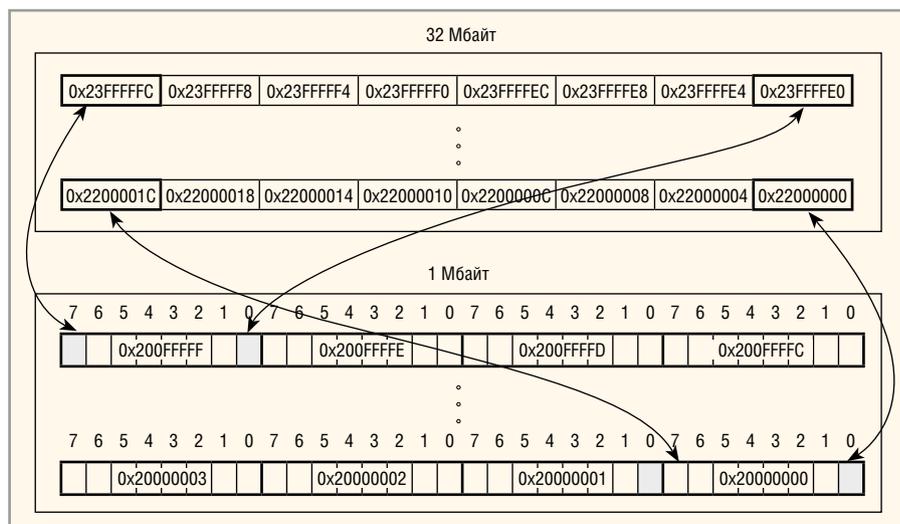


Рис. 2. Схема соответствия областей памяти битового объединения

Режимы начальной загрузки

Состояние выводов управления		Память	Режим загрузки
BOOT1	BOOT0		
0 или 1	0	Основная флэш-память	Программа пользователя в основной памяти
0	1	Системная память	Встроенный загрузчик системной памяти
1	1	Оперативная память	Программа в оперативной памяти

рядных значений. Но существует и другой способ управления битами этого регистра – с помощью регистра побитовой установки и сброса выводов порта BSRR. Примеры такого управления описаны ранее [2].

Приведём формулу для вычисления адреса слова в области доступа к битам на соответствующий бит в области хранения бит.

$$\text{bit_word_addr} = \text{bit_band_base} + (\text{byte_offset} \times 32) + (\text{bit_number} \times 4),$$

где:

bit_word_addr – адрес слова в области доступа к битам, которое отражается на нужный бит;

bit_band_base – начальный адрес области доступа к битам;

byte_offset – номер байта в области хранения бит, который содержит нужный бит;

bit_number – позиция нужного бита от 0 до 7 в байте.

Следующая формула показывает, как отобразить бит 2 из байта, расположенного в ОЗУ по адресу 0x20000300, на область доступа к битам:

$$\text{bit_word_addr} = 0x22000000 + (0x300 \times 32) + (2 \times 4) = 0x22006008.$$

Запись по адресу 0x22006008 будет иметь такой же эффект, что и операция «чтение-изменение-запись» для бита 2 из байта в ОЗУ по адресу 0x20000300.

Чтение по адресу 0x22006008 вернёт значение бита 2 для байта в ОЗУ по адресу 0x20000300 в виде значений

0x01 (если бит установлен) или 0x00 (если бит сброшен).

С помощью макроса можно создать указатель на этот адрес в виде следующей строки:

```
#define PB8 (*(volatile unsigned long *) 0x22006008) // Port B bit 8
```

Данный указатель можно использовать для установки и сброса бита 8 порта В следующим образом:

```
PB8 = 1; // Установить бит 8 в единичное состояние
```

```
PB8 = 0; // Установить бит 8 в нулевое состояние
```

Таким образом, можно воздействовать на значение отдельных бит путём обращения к определённой области памяти микроконтроллера за минимальное количество тактов.

На рисунке 2 наглядно видно соответствие между областью хранения бит размером 1 Мбайт и областью доступа к битам размером 32 Мбайт.

В программах, где часто используется установка и сброс множества бит в регистрах устройств ввода-вывода, использование данного метода позволяет существенно сэкономить размер кода программы и время его выполнения.

ЛИТЕРАТУРА

1. <https://www.st.com>
2. Вальна О. Современные 32-разрядные ARM-микроконтроллеры серии STM32: блок резервных данных ВКР. Современная электроника. № 3. 2014.