

Пример малого копировщика загрузки

В этой секции описывается пример копировщика загрузки с малым размером кода для пользователей, заинтересованных в использовании насколько возможно малой памяти.

Средства малого копировщика загрузки

Пример малого копировщика загрузки – это минимальная программа, использующая очень мало программной памяти. Она выполняет только следующие операции:

1. Читает запись загрузки приложения из флеш памяти
2. Копирует её в RAM
3. Переводит на точку входа приложения

Малый копировщик загрузки поддерживает только один образ загрузки, не может выполнять проверку на ошибки и не поддерживает вывод сообщений в JTAG UART. Если вас интересуют больше средств копировщика загрузки, обратитесь к секции "Пример расширенного копировщика загрузки" на стр. 4.

Этот пример разработан для запуска примера проекта Altera Nios II Ethernet Standard. За дополнительной информацией о примере проекта и поддержке плат разработчика Altera, обратитесь на страницу "[Пример проекта Nios II Ethernet Standard](#)" на веб сайте Altera.

Реализация на языке Nios II ассемблер

Чтобы получить минимально возможный размер кода, пример малого копировщика загрузки написан на языке Nios II ассемблер. Все переменные, используемые копировщиком загрузки, реализованы в регистрах общего назначения процессора Nios II, а не в RAM. Поэтому собственно копировщику загрузки не требуется память данных. Малый копировщик загрузки не имеет секций `.rodata`, `.rdata`, стек или куча. Поскольку ему не нужна память данных, этот копировщик загрузки может запросто размещаться в любом месте памяти, а также может быть запущен непосредственно из энергонезависимой памяти, без установки секции данных в памяти.

Инициализация системы

Малый копировщик загрузки выполняет только минимально необходимую инициализацию системы. Следующие задачи инициализации выполняются копировщиком загрузки:

- Сброс регистра статус процессора для запрета прерываний
- Сдвиг кеша инструкций
- Сдвиг процессорного конвейера

Размер кода

Малый копировщик загрузки компилируется в исполняемый файл длиной всего 200 байт. Такой копировщик загрузки достаточно мал, чтобы разместиться в одном блоке M9K, наименьшем элементе памяти в Cyclone III FPGA.

Реализация примера малого копировщика загрузки

В этой секции описываются шаги, необходимые для сборки и запуска примера малого копировщика загрузки на примере проекта Nios II Ethernet Standard. Этот копировщик загрузки - абсолютно малая по размеру кода версия, написанная на языке ассемблера. Если вас интересуют больше средств копировщика загрузки, обратитесь к секции "Реализация примера расширенного копировщика загрузки" на стр. 11.

Пример малого копировщика загрузки собирается в командной строке Nios II, используя утилиту **make**.

Установки программных инструментов и платы разработки

Для создания и запуска примера расширенного копировщика загрузки, вы должны выполнить следующие пункты:

1. Убедиться, что на компьютере установлены Nios II EDS версии 10.0 (или старше) и программа Quartus II версии 10.0 (или старше).
2. Питание и кабель USB-Blaster™ подключены к плате разработки Altera.

Создание соответствующего аппаратного проекта

В следующих пунктах вы открываете, модифицируете и генерируете систему Nios II, на которой вы собираетесь запускать пример малого копировщика загрузки.

Для открытия примера проекта и добавления внутри чиповой ROM в систему:

1. Убедитесь, что вы загрузили и распаковали файлы, описанные в секции "Файлы аппаратного проекта" на стр. 2 в вашу директорию `<project>`.
2. В программе Quartus II кликните на меню **File > Open Project** и откройте файл проекта `<project>\niosii_ethernet_standard_<board>.qpf`.
3. Кликните **Tools > SOPC Builder** для запуска SOPC Builder.
4. В SOPC Builder на вкладке **System Contents** раскройте **Memories and Memory Controllers**, раскройте **On-Chip** и выберите **On-Chip Memory (RAM or ROM)**.
5. Кликните **Add** для добавления компонента в систему. Используйте следующие настройки для спецификации памяти:
 - **Memory Type: ROM (Read-only)**
 - **Single-port access**
 - **Data width: 32 бита**
 - **Total memory size: 512 Bytes**

Заданный размер памяти означает, что никакое пространство памяти не будет израсходовано впустую. Наименьший используемый блок памяти Cyclone III FPGA – 512 байт (один блок M9K). Несмотря на то, что примеру малого копировщика загрузки требуется только 200 байт памяти, оставшаяся часть блока M9K может быть использована только после того, как вы разрешите его. Поэтому Altera рекомендует вам разрешить этот блок, что предпочтительнее, чем просто потерять его.

6. Правым кликом на компонент **On-Chip Memory** выберите **Rename**. Переименуйте компонент на описательное имя `boot_rom`.
7. В матрице соединений SOPC Builder проследите за тем, чтобы слейв порт внутри чиповой памяти был подключен к мастеру инструкций Nios II и к мастеру данных Nios II.
8. Если в нижнем окне SOPC Builder рапортует об ошибке, вызванной перекрытием адресов новой внутри чиповой памяти с другой периферией, выберите соответствующий базовый адрес для внутри чиповой памяти, чтобы ни с чем не перекрываться.
9. Модифицируйте тактовый вход для новой внутри чиповой памяти, чтобы сделать её тактируемой одинаковым тактом с компонентом **cpu**.
10. Для разрешения запуска копировщика загрузки из внутри чиповой памяти, правым кликом на компонент **cpu** вашей системы, выберите **Edit**.
11. В окне настроек процессора Nios II установите **Reset Vector Memory** на **boot_rom** с **Offset 0x00000000**.
12. Кликните **Finish** для выхода из окна настроек Nios II.
13. Кликните **Generate** для генерирования системы SOPC Builder.

Сборка примера малого копировщика загрузки с использованием 'make'

В следующих пунктах вы соберёте пример копировщика загрузки из командной строки Nios II.

Для сборки примера копировщика загрузки в новой директории проекта Quartus II выполните следующие пункты:

1. Проверьте, что вы загрузили и разархивировали файлы, описанные в секции "Файлы аппаратного проекта" на стр. 2, в вашу директорию `<project>`.
2. Откройте командную строку Nios II. (В Windows кликните **Пуск > Все программы > Altera > Nios II EDS > Nios II Command Shell**).
3. Смените директорию на `<project>/boot_copier_sw/app/small_boot_copier`.
4. В SOPC Builder задайте базовый адрес вашего компонента `ext_flash` (`<flash_base_address>`).
5. В командной строке Nios II введите следующую команду:

```
make all FLASH_BASE=<flash_base_address> \
      BOOT_IMAGE_OFFSET=0x00240000
```

Этой командой собирается малый копировщик загрузки, кодируется в виде образа загрузки по офсету 0x00240000 во флеш памяти и препятствует размещению здесь других важных данных. Вы легко можете изменить этот офсет на значение, более подходящее вашему приложению, но если ваша программа имеет образ загрузки во флеш памяти (п. 2 на стр. 22), проконтролируйте, чтобы её офсет совпадал с офсетом в этом пункте.

В этом примере используется makefile вместо инструментов разработки программы под Nios II, поскольку вам нужно скомпилировать один ассемблерный файл без ассоциированных с ним драйверов.

Сейчас у вас есть исполняемый файл копировщика загрузки - **small_boot_copier.hex** – готовый для запуска в процессоре Nios II. Далее вам нужно создать приложение, загружаемое новым копировщиком загрузки.

Сборка тестового приложения для загрузки

Чтобы собрать тестовое приложение для загрузки малым копировщиком загрузки, выполните шаги из секции "Сборка тестового приложения для загрузки" на стр. 13.

Загрузка из внутри чиповой памяти

В этой секции вы используете программу Quartus II для программирования малого копировщика загрузки в FPGA **boot_rom** память, а затем используете флеш программатор Nios II для программирования тестового приложения записи загрузки в CFI флеш память.

Для программирования копировщика загрузки в память FPGA **boot_rom**, выполните следующие пункты:

1. В командной строке Nios II смените поддиректорию вашего проекта Quartus II `<project>/boot_copier_sw/app/small_boot_copier`.
2. Скопируйте **small_boot_copier.hex** в директорию проекта Quartus II и переименуйте его на **boot_rom.hex**, используя команду:

```
cp small_boot_copier.hex ../../../../boot_rom.hex ↵
```

Вы можете увидеть предупреждение, о том, что такой файл уже присутствует в директории. Если вас спросят, перезаписать старый файл, кликните **Yes**.

При следующей компиляции Quartus II реализует исполняемый файл копировщика загрузки в качестве содержимого памяти **boot_rom**.

3. Если SOPC Builder ещё открыт, войдите в него и крикните **Exit** для его закрытия.
4. В окне программы Quartus II кликните **Assignments > Settings**.
5. В списке **Category** кликните **Compilation Process Settings**, затем включите **Use Smart Compilation**. Эта опция запрещает перекомпиляцию всего проекта, когда требуется только обновить содержимое внутри чиповой памяти. Первая компиляции, однако, должна быть полной, поскольку добавление внутри чиповой памяти изменяет проект.
6. В программе Quartus II кликните **Processing > Start Compilation** для компиляции проекта.
7. Когда компиляция закончится, кликните **Tools > Programmer**.
8. Убедитесь, что файл `<project>\niosii_ethernet_standard_<board>.sof` находится в столбце **File**.
9. Убедитесь, что опции **Program/Configure** включены.
10. Кликните **Start** для конфигурирования вашего FPGA .sof файлом.

Память **boot_rom** в FPGA теперь содержит исполняемый образ примера копировщика загрузки.

Для программирования тестового приложения в CFI флеш память, выполните следующие пункты:

1. В командной строке Nios II смените поддиректорию вашего проекта Quartus II `<project>/boot_copier_sw/app/hello_world`.
2. Установите офсет во флеш памяти, где будет расположен образ загрузки `hello_world`, введя следующие команды:

```
bin2flash --input=hello_world.elf.flash.bin \  
--output=hello_world.flash \  
--location=0x00240000 ↵
```

Установите размещение на 0x240000, поскольку в режиме загрузки из CFI флеш, малый копировщик загрузки найдёт по этому офсету образ загрузки. Правильное значение для параметра `--location` – это значение, заданное в п. 5 на стр. 21.

3. Запрограммируйте загрузочный образ `hello_world` во флеш память, введя следующую команду:

```
nios2-flash-programmer --base=<flash base> \  
hello_world.flash ↵
```

где `<flash_base>` - это базовый адрес компонента CFI флеш в вашей системе SOPC Builder.

Запуск примера малого копировщика загрузки

Для запуска примера малого копировщика загрузки на вашей плате разработчика:

1. После завершения работы флеш программатора, в командной строке Nios II введите следующую команду для сброса процессора Nios II:

```
nios2-download -r -g ↵
```

Загрузчик должен запустить тестовое приложение.

2. Чтобы проверить, что тестовое приложение правильно загружено и выполняется, запустите утилиту **nios2-terminal** в командной строке Nios II, введя следующую команду:

```
nios2-terminal ↵
```

Если копировщик загрузки был успешно запущен, вы увидите результат в **nios2-terminal**, как показано на рис. 4.

Если **nios2-terminal** не подключен к JTAG UART с настройками по умолчанию, запустите его с опцией `--help`, чтобы посмотреть ключи командной строки, которые могут понадобиться.

Figure 4. Small Boot Copier Output

