

УТВЕРЖДЕН
РОФ.КФДЛ.00398-03 32 01-ЛУ

Программное обеспечение «GRAIP»

Руководство системного программиста

РОФ.КФДЛ.00398-03 32 01

2019

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

Аннотация

Программное обеспечение (ПО) «GRAIP» предназначено для программирования и отладки программ, разработанных для микропроцессора 1906BM016.

ПО предназначено для работы в среде операционной системы Windows.

Взаимодействие персонального компьютера с установленным ПО «GRAIP» и микропроцессора может осуществляться через интерфейсы Ethernet, ANBUART.

Полная функциональность ПО «GRAIP» гарантируется только при работе с отладочным устройством КФДЛ.424939.013.

Для корректной работы ПО необходима установка пакета Visual C++ Redistributable, который можно скачать по ссылке: <https://www.microsoft.com/ru-ru/download/details.aspx?id=53840>.

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

Содержание

1	Интерфейс отладки Ethernet.....	4
2	Интерфейс отладки ANBUART	5
3	Порядок работы	5
4	Поддерживаемые команды.....	8
4.1	Список команд	8
4.2	Описание команд.....	8
	Лист регистрации изменений.....	17

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

1 Интерфейс отладки Ethernet

Для установки соединения необходимо, чтобы сетевое устройство находилось в той же сети, что и процессор 1906BM016. Адрес процессора по умолчанию 192.168.0.51.

Например, если адрес процессора не был изменен после сброса (кодом микропрограммы или через другой отладочный интерфейс), то при использовании маски подсети 255.255.255.0 первые три байта подсети на компьютере должны быть 192.168.0, адрес устройства в пределах этой сети (последний байт) может быть любым, кроме значения, которое соответствует адресу процессора.

Программа должна быть добавлена в список исключений в настройках брандмауэра Windows (см. рисунок 1).

Разрешить связь для программ через брандмауэр Windows

Чтобы добавить, изменить или удалить разрешенные программы и порты, нажмите кнопку "Изменить параметры".

Риски разрешения связи для программы.

Изменить параметры

Для обеспечения безопасности управление некоторыми параметрами осуществляет системный администратор.

Разрешенные программы и компоненты:

Название	Домен	Домашняя или ...	Публичные	Групповая пол...
<input type="checkbox"/> eclipse.exe	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет
<input checked="" type="checkbox"/> FileZilla FTP Client	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет
<input checked="" type="checkbox"/> Firefox (C:\Program Files\Moz...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Нет
<input checked="" type="checkbox"/> GRAIP	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет
<input type="checkbox"/> grmon	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет
<input type="checkbox"/> Intranet Chat	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет
<input type="checkbox"/> java.exe	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет
<input type="checkbox"/> java.exe	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет
<input type="checkbox"/> javaw.exe	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет
<input type="checkbox"/> javaw.exe	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Нет
<input checked="" type="checkbox"/> Kaspersky Administration Kit	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Нет
<input checked="" type="checkbox"/> Kaspersky Administration Kit	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Нет

Сведения

Удалить

Разрешить другую программу...

Рисунок 1 – Настройка брандмауэра

Специализированных требований к Ethernet-кабелю не предъявляется.

В разделе 3 «Порядок работы» приведены доступные параметры конфигурации интерфейса.

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

2 Интерфейс отладки АНВUART

Для подключения через интерфейс АНВUART (AMBA АНВ Serial Debug Interface) к отладочному устройству КФДЛ.424939.013 используется кабель USB Type A – Type B. Это соединение подразумевает использование «виртуального» последовательного порта.

Для корректного функционирования может потребоваться установить FTDI Virtual Com драйвер. Драйвер можно скачать по ссылке: <https://www.ftdichip.com/Drivers/VCP.htm>.

Рекомендуется драйвер «VCP Drivers 2.12.28».

В отличие от отладки по интерфейсу «Ethernet», при отладке по интерфейсу «АНВUART» необходимо выполнять инициализацию (команда «init») после сброса питания.

В разделе 3 «Порядок работы» приведены доступные параметры конфигурации интерфейса.

3 Порядок работы

Перед началом работы необходимо убедиться, что у процессора есть разрешение на вход в режиме отладки. Для этого на внешний вывод микропроцессора DSUEN необходимо подавать высокий логический уровень.

В корневой папке ПО «GRAIP» находится файл «settings.txt», который содержит пользовательские настройки для работы программы. Если требуется изменить настройки, необходимо закрыть ПО, если оно запущено, затем отредактировать файл настроек, сохранить его и закрыть. Только после этого можно выполнять перезапуск ПО «GRAIP».

В файле «settings.txt» имеется 3 блока «main, Ethernet, АНВUART».

В блоке «main» настраиваются уровни логирования, задержка для стирания микросхемы проверочных бит. Также в данном блоке задается интерфейс APB UART, который будет инициализирован для обеспечения вывода отладочной информации запуска образа Linux в консоль «GRAIP» (по умолчанию – APB UART 1).

В блоке «Ethernet» имеется возможность изменять максимальное количество байт данных в пакете при чтении и записи в режиме отладки.

В блоке «АНВUART» содержатся настройки подключения: номер порта «port» и скорость работы «baud». Выбор данных настроек может быть произведен автоматически без участия пользователя. Так же имеется возможность изменять максимальное количество байт данных в пакете при чтении и записи в режиме отладки.

В случае удаления файла «settings.txt» при запуске ПО он будет создан заново со стандартными настройками.

Запустить исполняемый файл «graip_eth.exe» или «graip_ahbuart.exe» в зависимости от выбранного отладочного интерфейса.

В папке «log», находящейся в корневой папке ПО «GRAIP», автоматически сохраняется информация о выполнении текущей сессии.

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

Если с первого раза не удастся установить соединение с отладочным устройством, ПО «GRAIP» предпримет еще несколько попыток (с увеличивающимся временем ожидания между ними).

В начале работы программы посылается пакет данных и ожидается ответ процессора. В случае если соединение так и не удалось установить (процессор не отвечает), программа закрывается. Возможные аппаратные причины такого поведения: отсутствие тактирования процессора, проблемы с физическим подключением платы к компьютеру, плохой контакт процессора в соquete. Если запуск выполняется с использованием интерфейса Ethernet, то возможны следующие программно-аппаратные проблемы: отсутствие необходимых разрешений в брандмауэре Windows, IP-адрес отладочного интерфейса Ethernet процессора не принадлежит подсети сетевой карты, IP-адрес сетевой карты совпадает с IP-адресом отладочного интерфейса Ethernet процессора, аппаратные проблемы с сетевой картой компьютера.

При успешном запуске программы окно программы будет иметь вид, представленный на рисунке 2.

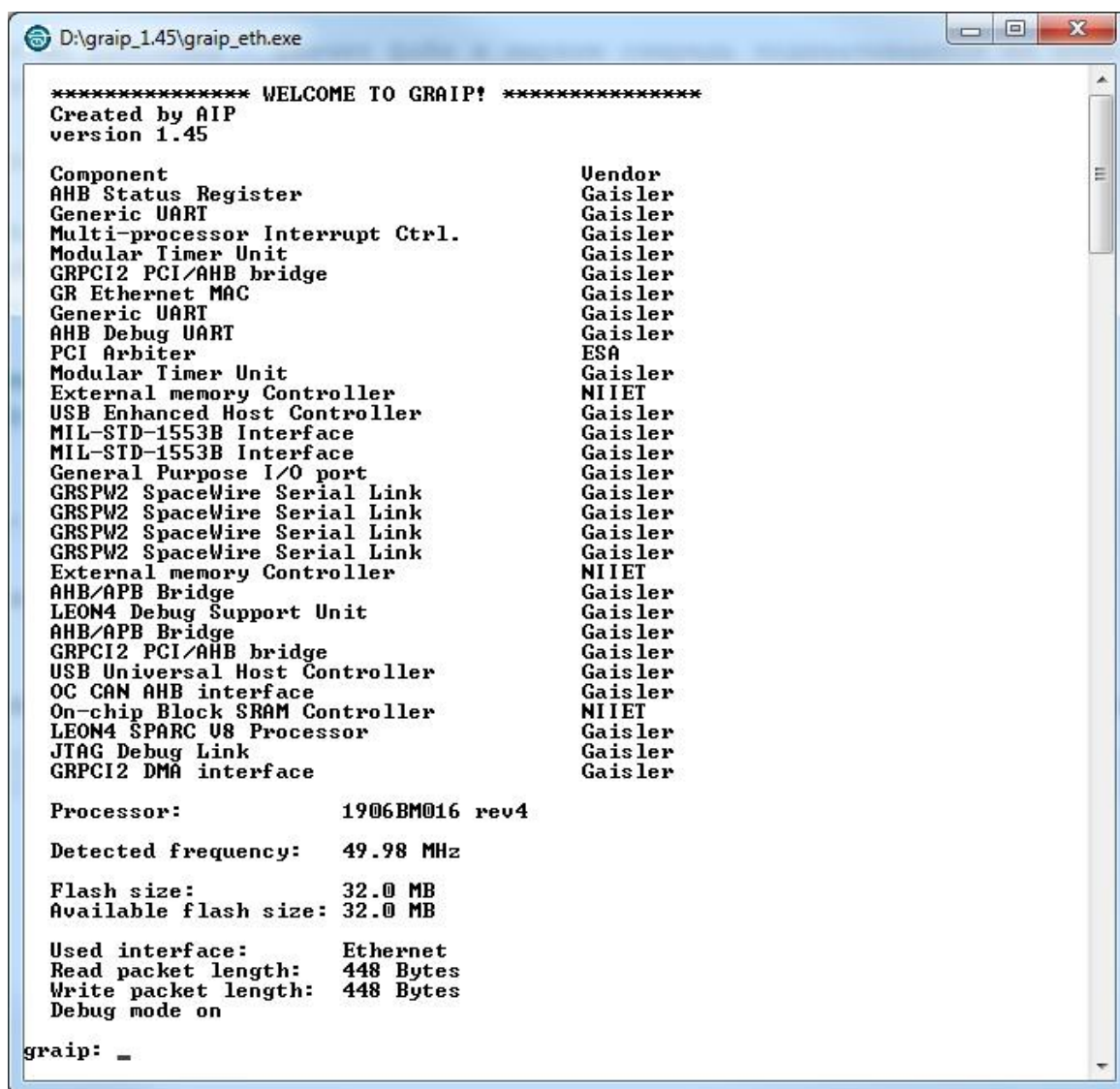


Рисунок 2 – Внешний вид окна программы при успешном запуске

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

При неудачном запуске программы окно программы будет иметь вид, представленный на рисунке 3.

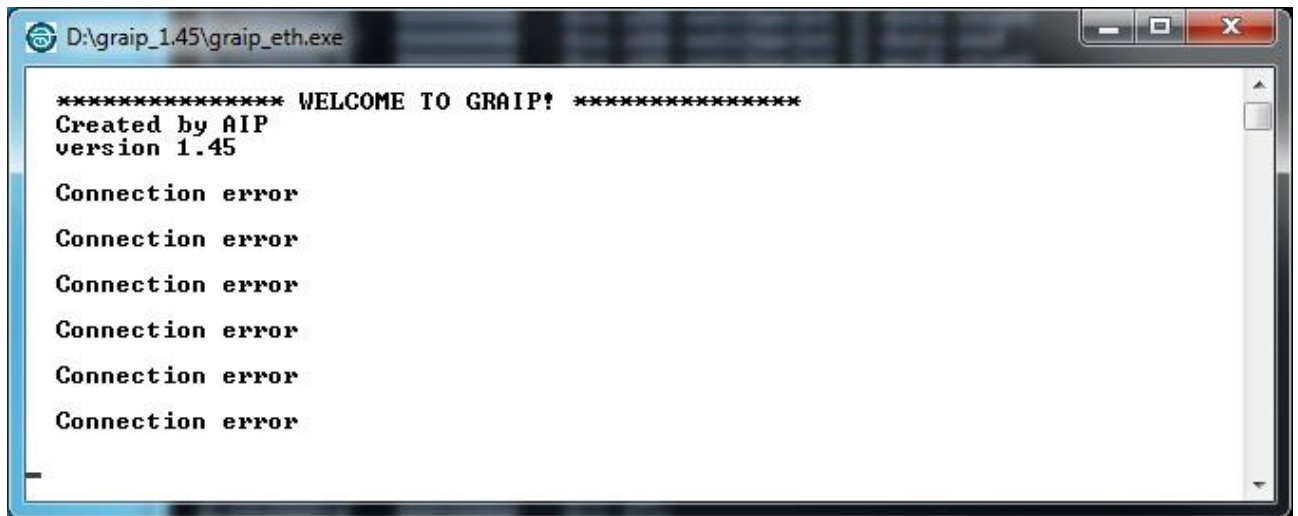


Рисунок 3 – Внешний вид окна при неудачном запуске

Примечание

Программа поддерживает пакетное исполнение команд.

В папке «scripts» находятся примеры исполняемых файлов инициализации работы процессора с памятью SDRAM, а также файлов загрузки рабочей программы в память и запуска на выполнение.

Программа поддерживает запуск с передачей пакета исполняемых команд. Таким образом можно создавать «.bat» файлы, которые сразу после запуска ПО «GRAIP» выполняют определенный набор команд. После выполнения набора команд ПО «GRAIP» переходит в режим приема команд управления от пользователя. Примеры исполняемых «.bat» файлов приведены в папке «bat_file».

Прервать выполнение текущей команды можно нажатием комбинации клавиш «Ctrl + C».

Инв. N подл.	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

4 Поддерживаемые команды

4.1 Список команд

Команды, которые реализованы в ПО «GRAIP», представлены в таблице 1.

Таблица 1 – Команды, поддерживаемые ПО «GRAIP»

Имя команды	Описание
ahb	Группа команд для работы с буфером трассировки шины АНВ
bp	Группа команд для работы с точками останова
cont	Возобновление выполнения работы программы после остановки
disassemble	Дизассемблирование памяти
exit	Выход из ПО «GRAIP»
flash	Группа команд для работы с Flash-памятью
go	Запуск выполнения программы без программного сброса
help	Встроенная справка
init	Начальная инициализация процессора
inst	Вывод содержимого буфера трассировки инструкций
load	Загрузка в RAM файла прошивки
mem	Чтение данных по адресу
ocram init	Инициализация внутренней статической памяти OCRAM
reg	Установка/вывод значений регистров регистрового окна и регистров статуса процессора
reset	Программный сброс
run	Запуск выполнения программы с предварительным программным сбросом
runfile	Исполнение пакета команд
show	Вывод информации о блоке/регистрах процессора
sdclk	Подбор параметра задержки сигнала SDCLK
sdram init	Инициализация процессора для работы с памятью SDRAM
sram init	Инициализация процессора для работы с памятью SRAM
stack	Установка/вывод значения указателя вершины стека
step	Пошаговое выполнение
tmode	Группа команд выбора режима работы буферов трассировки
verify	Верификация записанной программы или области памяти
wash	Заполнение памяти заданными данными
wmem	Пословная запись данных по заданному адресу

4.2 Описание команд

[ahb](#)

Группа команд для работы с буфером трассировки шины АНВ. За включение/выключение и отображение статуса буфера трассировки отвечает группа команд [tmode](#).

Набор представленных команд:

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

ahb – вывод содержимого буфера трассировки шины АНВ.

ahb force <true*/false> – включение/выключение режима принудительной трассировки АНВ в режиме отладки. Если принудительная трассировка активирована, то в буфер трассировки шины АНВ будут попадать все обращения, включая запросы используемого интерфейса отладки. * – значение по умолчанию. Вместо false допустимо использовать 0, вместо true ненулевое число.

ahb status – вывод статуса буфера трассировки АНВ.

bp

Группа команд для установки, удаления и вывода информации о точках останова. Поддерживается два типа точек останова: программные (тип «soft») и аппаратные (тип «hard»). Программные точки останова реализуются через подмену инструкций, которые содержатся в памяти, на «ta 1» (не подходят для установки в программе, выполняемой из flash или не перезаписываемой памяти). Аппаратные точки останова не изменяют содержимое памяти, но их число ограничено.

Одновременно может быть установлено только две точки типа «hard». Механизм хранения точек останова организован по типу буфера FIFO, т.е. при добавлении новой точки останова, если свободных ячеек нет, удаляется самая старая точка.

При запуске записанной программы (например, командой «run») происходит запись адресов и масок точек останова в соответствующие регистры процессора или подмена инструкций, на которые пришлись программные точки останова. При корректном завершении работы программы данные регистры будут обнулены, подменяемые инструкции – восстановлены. Таким образом, выполнение только команды «bp» не приводит к изменению регистров процессора. Остановка процессора происходит перед выполнением инструкции по адресу точки останова.

Набор представленных команд:

bp – вывод информации.

bp hard – вывод информации о точках останова типа «hard».

bp soft – вывод информации о точках останова типа «soft».

bp soft <адрес> – установка программной точки останова.

bp <адрес> – эквивалентно команде bp soft <адрес>.

bp hard <адрес> <маска> – добавление точки останова по указанному адресу/диапазону адресов.

<адрес> – адрес точки останова.

<маска> – маска точки останова. Параметр маски не обязателен, значение по умолчанию – 0xffffffffc.

Пример команды: «bp hard 0x4000000f 0xffffffff0».

bp del hard <номер точки останова> – удаление аппаратной точки останова.

Пример команды: «bp del hard 1».

bp del soft <номер точки останова> – удаление программной точки останова.

Пример команды: «bp del soft 1».

bp del all – удаление всех точек останова.

cont

Команда возобновляет выполнение программы с инструкции, располагающейся по адресу, который содержится в регистре «PC».

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

disassemble

Дизассемблирование участка памяти.

disassemble <address> <length> – дизассемблирование участка памяти начиная с <address>. Значение <address> по умолчанию равно значению PC (program counter). Значение количества дизассемблированных слов <length> по умолчанию равно 16.

Пример команды: «disassemble 0x40000000 25».

disassemble -r <start_address> <stop_address> – дизассемблирование участка памяти начиная с <start_address> и заканчивая <stop_address> включительно.

Так же поддерживается краткий формат записи данной команды – «dis».

Примеры с краткой формой:

«dis»

«dis 0x40001000»

«dis -r 0x40000000 0x40000040»

exit

Выход из программы.

flash

Группа команд для работы с flash-памятью, предназначенных для выполнения записи, стирания и верификации записанной программы, активации/деактивации защиты блоков flash от записи.

Набор представленных команд:

flash erase all – стирание всего содержимого flash-памяти. Так же, если отладочная плата и процессор поддерживают механизм битов защиты, происходит стирание и flash-памяти, содержащей проверочные биты.

flash load <путь к файлу прошивки> – загрузка файла прошивки во flash. Файл прошивки должен иметь формат «srec».

В качестве последнего параметра должен быть введен либо полный путь до файла прошивки, либо путь до файла относительно корневой папки ПО «GRAIP».

Пример команды: «flash load srec/prom.srec».

flash loadcb <файл прошивки> – загрузка файла прошивки и проверочных битов защиты во flash. Файл прошивки должен быть скомпилирован с учетом диапазона адреса flash-памяти и иметь формат «srec». Проверочные биты формируются процессором автоматически и записываются в соответствующую микросхему flash-памяти.

В качестве последнего параметра должен быть введен либо полный путь до файла прошивки, либо путь до файла относительно корневой папки ПО «GRAIP».

Пример команды: «flash loadcb srec/prom.srec».

flash lock <номер блока flash> – включение защиты блока flash от записи. Выполнение команды без указания номера блока приведет к включению защиты всех блоков памяти.

Пример команды: «flash lock 3».

flash unlock <номер блока flash> – выключение защиты блока flash от записи. Выполнение команды без указания номера блока приведет к выключению защиты всех блоков памяти.

Пример команды: «flash unlock 3».

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

flash status – проверка состояния(lock/unlock) блоков flash-памяти.

go

Запуск выполнения программы без программного сброса.

Формат команды: go <стартовый адрес> – выполнение программы с указанного адреса. Выполнение команды без входного параметра приведет к запуску программы либо с адреса по 0x00000000 (поведение по умолчанию), либо с начального адреса загруженной ранее прошивки.

Пример команды: «go 0x40000000».

help

Встроенная справка по командам.

Формат: help <название команды> – вызов справки по заданной команде. При выполнении команды без указания входного параметра будет выведен список поддерживаемых команд.

Примеры команды: «help», «help go».

init

Начальная инициализация. Выполняется автоматически при старте ПО «GRAIP», но может быть выполнена в любое время, например, после повторной подачи питания на процессор. Сканирует шины APB/AHB, выводит список устройств. На данном этапе ПО выполняет разметку областей, составление карты регистров модулей, реализованных в процессоре. Так же выполняется расчет тактовой частоты, область PROM анализируется на наличие модулей flash-памяти, если таковые обнаружены, выводится их суммарный объем.

inst

Вывод содержимого буфера трассировки инструкций.

Формат команды: inst <количество выводимых записей>.

Пример команды: «inst 20» – вывести последние 20 записей в буфере трассировки инструкций. Значение по умолчанию – 10. Глубина кольцевого буфера трассировки инструкций процессора 1906BM016 – 64 записи.

load

Загрузка файла прошивки во внешнюю/внутреннюю оперативную память. Файл прошивки должен быть скомпилирован с учетом диапазона доступной RAM-памяти и иметь формат «srec». Первый адрес загружаемого файла используется как стартовый по умолчанию, т.е. при выполнении команды «run» или «go» с него будет осуществляться старт программы.

Формат команды: load <файл прошивки>.

В качестве параметра «файл прошивки» должен быть введен либо полный путь до файла прошивки, либо путь до файла относительно корневой папки ПО «GRAIP».

Пример команды: «load srec/ram.srec».

mem

Чтение данных по заданному адресу.

Формат команды: mem <начальный адрес> <количество считываемых байт>.

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

Начальный адрес – адрес, с которого начинается считывание данных, должен быть выравнен по границе слова. Перед выполнением команды адрес будет принудительно выравнен (2 младших бита адреса будут сброшены).

Количество байт – количество считываемых байт, должно быть кратно четырем. Значение по умолчанию – 64. Если количество байт не кратно четырем, происходит выравнивание значения в меньшую сторону.

Пример команды: «mem 0x40000000 64» – считать 64 байт данных, начиная с адреса 0x40000000.

«mem 0x40000003 7» – значение количества бит и адрес будут выравнены, результат команды будет аналогичен результату команды «mem 0x40000000 4».

ocram init

Инициализация массива памяти OCRAM значениями 0x00000000.

reg

Установка/вывод значений регистров регистрового окна и регистров статуса процессора.

1. Вывод информации

Формат команды: reg <имя регистра>.

При выполнении команды без аргументов будет выведена информация по всем регистрам текущего регистрового окна и регистрам статуса процессора.

Примеры команды:

«reg pc» – вывод информации о регистре PC

«reg g1» – выводится информация о первом глобальном регистре регистрового окна.

2. Установка значения регистра

Формат команды: reg <имя регистра> <значение>.

Пример команды:

«reg об 0x40ffffc» – установить регистр об.

reset

Программный сброс процессора.

run

Запуск выполнения программы с предварительным программным сбросом. Представляет собой последовательное выполнение команд «reset» и «go».

runfile

Запуск пакета команд, хранящихся в файле с расширением txt.

Формат команды: runfile <файл_с_набором_команд.txt>.

Пример txt-файла:

init

load ram.srec

ve ram.srec

run

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

Пример команды: «runfile test.txt».

В таком исполнении команды файл «test.txt» должен располагаться в корневой папке ПО «GRAIP».

Допускается сокращение имени команды до «rf».

show

Вывод информации о блоке/регистрах процессора. Команда совершает несколько проходов по карте памяти процессора и собирает все возможные наиболее близкие совпадения. Выводится адрес, значение и полное название регистра(-ов). При выполнении команды без аргументов выводится список всех доступных модулей процессора.

Формат команды: show <название регистра/блока>.

Примеры команды:

«show dsu» – выводится вся информация о регистрах модуля DSU.

«show status» – выводится информация обо всех регистрах, содержащих слово «status» в названии.

sdclk

Подбор параметра задержки сигнала SDCLK. Команда актуальна только для процессора 1906BM016. Запись выбранного параметра в регистр sdclk не производится. Данная команда может быть выполнена только после инициализации SDRAM.

Формат команды: sdclk.

sdram init

Инициализация процессора для работы с памятью SDRAM.

sram init

Инициализация процессора для работы с памятью SRAM.

stack

Установка/вывод значения указателя вершины стека.

Формат команды: stack <адрес вершины стека>.

Выполнение команды эквивалентно установке значения регистра об (sp) текущего окна («reg sp <адрес вершины стека>» или «reg об <адрес вершины стека>»).

Пример команды установки начального адреса стека: «stack 0x40ffffc».

При выполнении команды без аргумента будет выведено значение указателя вершины стека текущего регистрового окна.

step

Заставить процессор выполнить одну инструкцию – шаг, либо последовательность таких шагов. После выполнения требуемого количества шагов процессор возвращается в режим отладки, управление передается пользователю.

Формат команды: step <количество шагов>.

По умолчанию программа делает один шаг.

Пример команды: «step 3».

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

tmode

Группа команд выбора режима работы буферов трассировки.

Набор представленных команд:

tmode – вывод информации о текущем режиме трассировки.

tmode none – выключение как трассировки буфера АНВ, так и буфера инструкций. Команда эквивалентна выполнению пары команд «tmode ahb false», «tmode proc false».

tmode both – включение как трассировки буфера АНВ, так и буфера инструкций. Команда эквивалентна выполнению пары команд «tmode ahb true», «tmode proc true».

tmode ahb <true*/false> – включение/выключение буфера трассировки АНВ. * – значение по умолчанию. Вместо false допустимо использовать 0, вместо true ненулевое число.

tmode proc <true*/false> – включение/выключение буфера трассировки инструкций. * – значение по умолчанию. Вместо false допустимо использовать 0, вместо true ненулевое число.

tmode break <delay*> – установка задержки буфера трассировки АНВ. Значение по умолчанию – 0.

verify

1. Верификация записанной программы. Работает как с областью PROM (flash), так и с областью RAM.

Формат команды: verify <файл прошивки> <количество ошибок>.

В качестве параметра «файл прошивки» должен быть введен либо полный путь до файла прошивки, либо путь до файла относительно корневой папки ПО «GRAIP».

Количество ошибок – максимальное количество ошибок, выводимое в случае несоответствия записанной и верифицируемой прошивки. По умолчанию количество ошибок не ограничено.

Пример команды: «verify srec/ram.srec»

Так же поддерживается краткий формат записи данной команды – «ve».

Пример с краткой формой: «ve srec/ram.srec»

2. Верификация области памяти.

Верификация области памяти.

Формат команды: verify <начальный адрес> <конечный адрес> <проверочное слово>.

Начальный адрес – адрес, с которого начинается верификация данных, должен быть выравнен по границе слова. Перед выполнением команды адрес будет принудительно выравнен (2 младших бита адреса будут сброшены).

Конечный адрес – адрес, до которого происходит верификация данных, должен быть выравнен по границе слова. Перед выполнением команды адрес будет принудительно выравнен (2 младших бита адреса будут сброшены).

Проверочное слово – слово данных, с которым сравниваются считанные из памяти значения. По умолчанию равно нулю.

Необязательные параметры.

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

«inc <значение>» – значение, на которое увеличивается проверочное слово с каждым увеличением адреса. По умолчанию равно единице. Если параметр «inc» не указан, проверочное слово не инкрементируется.

«err <значение>» – максимальное количество сообщений об ошибке верификации, выводимое пользователю. По умолчанию равно 10000.

Примеры команды:

«verify 0x40000000 0x40000020 0x5 inc 2 err 5»;

«verify 0x00000000 0x00001000 0xffffffff err 1».

Пример с краткой формой: «ve 0x40000000 0x40000020 0x5 inc 2 err 5».

wash

Заполнение памяти заданными данными. Команда не подходит для заполнения flash-памяти, поскольку последовательность записи состоит из пословных запросов на запись по шине АНВ. Для организации записи во flash-память следует использовать группу команд [flash](#).

Формат команды: wash <начальный адрес> <конечный адрес>.

Начальный адрес – адрес, с которого начинается заполнение данными, должен быть выравнен по границе слова. Перед выполнением команды адрес будет принудительно выравнен (2 младших бита адреса будут сброшены).

Конечный адрес – адрес, до которого происходит заполнение данными (не включительно), должен быть выравнен по границе слова. Перед выполнением команды адрес будет принудительно выравнен (2 младших бита адреса будут сброшены).

Необязательные параметры.

«Слово данных» – слово данных, которым заполняются ячейки памяти. По умолчанию равно нулю.

«inc <значение>» – значение, на которое увеличивается проверочное слово с каждым увеличением адреса. По умолчанию равно единице. Если параметр «inc» не указан, слово данных не инкрементируется.

«ve <значение>» – выполнять верификацию после заполнения участка памяти, где поле «значение» определяет максимальное количество сообщений об ошибке верификации, выводимое пользователю. По умолчанию оно равно 10000. Если параметр «ve» не указан, верификация данных не выполняется. Параметр «ve <значение>» можно заменить параметром «err <значение>».

Пример команды: «wash 0x40000000 0x40000020 0x5 inc 2 err 10».

Данная команда заполнит диапазон памяти 0x40000000 – 0x40000020 инкрементируемыми на 2 значениями, начиная с 0x00000005, верифицирует записанные данные и выведет 10 ошибок в случае несоответствия записанных и считанных данных.

wmem

Пословная (по 32 бита) запись данных по заданному адресу. Команда формирует один или несколько пословных запросов на запись по шине АНВ. Если требуется выполнить запись во flash-память, то следует использовать группу команд [flash](#).

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

Формат команды: `wmem <начальный адрес> <данные>`.

Начальный адрес – адрес, с которого начинается запись данных, должен быть выравнен по границе слова. Перед выполнением команды адрес будет принудительно выравнен (2 младших бита адреса будут сброшены).

Данные – одно или несколько значений данных. Количество значений данных для одновременной записи ограничено 128.

Примеры команды:

«`wmem 0x40000000 0x12345678 0x99999999`» – записать слово данных 0x12345678 по адресу 0x40000000, слово данных 0x99999999 по адресу 0x40000004.

«`wmem 0x40000003 7`» – значение адреса будет принудительно выравнено, результат команды будет аналогичен результату команды «`wmem 0x40000000 7`».

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата

[illegible]

Инв. N подл	Подп. и дата	Взам. инв. N	Инв. N дуб.	Подп. и дата