

УТВЕРЖДЕН
RU.ФЛАБ.00001-01 32 01-ЛУ

**БАЗОВАЯ СИСТЕМА
ВВОДА–ВЫВОДА NP-BIOS**

Руководство системного программиста

RU.ФЛАБ.00001-01 32 01

Инв. № подл.	Подп. и дата
Взам. инв. №	Инв. № дубл.
Подп. и дата	Подп. и дата

2021

Литера: О₁

Изм.	Лист	№ докум.	Подп.	Дата

АННОТАЦИЯ

В данном программном документе приведено руководство системного программиста по установке, настройке и использованию «Базовой системы ввода – вывода NP-BIOS» базирующейся на открытом исходном коде Coreboot и массивов двоичных данных от Intel.

Изм.	Лист	№ докум.	Подп.	Дата

СОДЕРЖАНИЕ

1	ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ	5
1.1	Назначение программы.....	5
1.2	Функции программы	5
1.3	Минимальный состав технических средств	6
1.4	Требования к персоналу	6
2	СТРУКТУРА ПРОГРАММЫ.....	7
2.1	Сведения о структуре программы	7
2.2	Сведения о связях с другими программами	7
3	КОМПИЛЯЦИЯ И СБОРКА ПРОГРАММЫ	8
3.1	Среда сборки программы	8
3.2	Состав среды сборки программы	8
3.3	Работа с файлом скрипта sboot.sh	9
3.4	Параметры файл скрипта.....	10
3.5	Запуск скрипта сборки программы	11
4	УСТАНОВКА ПРОГРАММЫ.....	12
4.1	Аппаратная прошивка.....	12
4.2	Программная прошивка.....	15
5	ВЫПОЛНЕНИЕ ПРОГРАММЫ.....	17
5.1	Загрузка и запуск.....	17
5.2	Выполнение программы	17
6	НАСТРОЙКА ПРОГРАММЫ	20
7	ПРОВЕРКА ПРОГРАММЫ.....	21
7.1	Описание способов проверки.....	21
7.2	Методы прогона	21
8	СООБЩЕНИЯ	22

Изм.	Лист	№ докум.	Подп.	Дата

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АПМДЗ	– аппаратно-программный модуль доверенной загрузки
БСВВ	– базовая система ввода-вывода
ОС	– операционная система
ЦП	– центральный процессор

Изм.	Лист	№ докум.	Подп.	Дата

1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1 Назначение программы

«БСВВ NP-BIOS» предназначена для проведения первичной инициализации платформы, старта загрузчика ОС с внутренних или внешних накопителей или посредством сетевой карты с сетевого сервера (PXE) на устройствах разработки ООО «Новые платформы».

1.2 Функции программы

1.2.1 БСВВ реализует следующие основные функции:

- работа в режимах Legacy (MBR) и UEFI;
- инициализация аппаратной платформы;
- запуск загрузчика ОС;
- выбор устройства загрузки ОС.

1.2.2 БСВВ реализует следующие дополнительные функции:

- настройка порядка загрузки устройств;
- поддержка видеоинтерфейса AST25xx;
- поддержка консольного порта RS-232;
- работа с АПМДЗ;
- возможность загрузки по сети PXE;
- поддержки оперативной памяти с коррекцией ошибок (ECC);
- поддержка режима многопоточности в ядре для процессоров, поддерживающих технологию Intel® Hyper-Threading;
- поддержка EFI Shell;
- выбор и интеграция текстового баннера и/или изображения в формате BMP/JPEG для вывода при старте на экран в текстовом/ графическом режиме запуска;

Изм.	Лист	№ докум.	Подп.	Дата

- возможность управления поддержкой аппаратной виртуализации VTХ;
- обеспечение при старте вывода дополнительной информации, в соответствии с требованиями Заказчиков.

1.3 Минимальный состав технических средств

1.3.1 Устройство под управлением процессоров Intel разработки ООО «Новые платформы».

1.3.2 Для применения дополнительных функций необходима их аппаратная поддержка.

1.4 Требования к персоналу

Персонал должен иметь минимум среднее техническое образование и изучить комплект эксплуатационной документации на устройство и БСВВ.

Изм.	Лист	№ докум.	Подп.	Дата

2 СТРУКТУРА ПРОГРАММЫ

2.1 Сведения о структуре программы

БСВВ состоит из одного бинарного файла и не имеет составных частей.

2.2 Сведения о связях с другими программами

В процессе работы БСВВ самодостаточна.

Бинарный файл БСВВ формируется с помощью скрипта сборщика, алгоритм работы которого приведен в описании программы RU.ФЛФБ.000001-01 13 01.

БСВВ в ходе своей работы инициализирует загрузчик ОС с внутренних или внешних накопителей, или с использованием сетевой загрузки PXE.

Примечание – Для функционирования БСВВ наличие установленной на платформе ОС не требуется.

Изм.	Лист	№ докум.	Подп.	Дата

3 КОМПИЛЯЦИЯ И СБОРКА ПРОГРАММЫ

3.1 Среда сборки программы

Среда сборки программы представляет собой совокупность скриптов и фалов, участвующих в сборке программы.

Примечание – среда сборки программы находится в соответствующей директории на сервере компании.

3.2 Состав среды сборки программы

Среда сборки программы, согласно рисунку 1, состоит из следующих директорий и скриптов:

- папка Common – содержит общие файлы среды сборки;
- папка Custom – содержит индивидуальные файлы для каждой конкретной сборки программы;
- папки NP-1x0 и NP-2xxx – содержит сборочные файлы серий платформ;
- папка Result – содержит итоговые файлы для установки программы;
- папка Patches – содержит общие файлы для сборочной машины;
- папка Scripts – содержит файлы скриптов сборочной машины;
- папка Sources – содержит файлы источников;
- папка Updates – содержит файлы обновления сборочной машины;
- файл скрипта build-ext.sh – основной скрипт сборки программы;
- файл скрипта sboot.sh – файл скрипта сборки программы с редактируемыми параметрами;

Изм.	Лист	№ докум.	Подп.	Дата

Левая панель					Файл		Команда		Настройки		Правая панель	
/home/coreboot/Coreboot/Coreboot-ВМ-02/coreboot-4.16-build-ext												
.и												
Имя												
Размер												
Время												
правки												
/..												
-ВВЕРХ-												
ноя 28 15:32												
/Common												
4096												
окт 3 15:15												
/Custom												
4096												
ноя 2 16:11												
/NP-1x0												
4096												
июл 5 12:39												
/NP-2xxx												
4096												
июн 28 18:38												
/Result												
4096												
ноя 28 15:33												
/patches												
4096												
мая 24 2022												
/scripts												
4096												
июл 12 18:13												
/sources												
4096												
июл 25 11:44												
/updates												
4096												
июл 12 12:41												
1												
3												
сен 29 12:49												
*build-ext.sh												
21870												
июл 25 12:07												
*cboot.sh												
501												
ноя 2 16:15												

Рисунок 1 – Среда разработки программы

3.3 Работа с файлом скрипта cboot.sh

Создание новой сборки программы производится посредством выполнения скрипта cboot.sh.

Скрипт представляет собой набор параметров, которые необходимо конфигурировать в зависимости от требований к программе и аппаратной платформе, на которую она будет устанавливаться.

Состав файла скрипта указан на рисунке 2.

```
#!/bin/bash
#####
PLFRM="2000i"
VAR="0"
VERSION="1.0.4"
BVERSION="1003"
RESULT_DIR="/home/coreboot/Coreboot/Coreboot-ВМ-02/coreboot-4.16-build-ext/Result/FT"

IMAGE_VERSION=${BVERSION} \
PLATFORM=${PLFRM} \
VER=${VERSION} \
TYPE="BIOS" \
PXE="1" \
AST="1" \
HT="1" \
EFISHELL="0" \
SBSPLASH="$(pwd)/Custom/FT/1024x768-dev.jpg" \
SBB0="$(pwd)/Custom/FT/bootorder.dev" \
VARIANT=${VAR} \
RES_DIR=${RESULT_DIR} \
UNLOCK="0" \
./build-ext.sh

exit 0
```

Рисунок 2 – Файл скрипта cboot.sh

Изм.	Лист	№ докум.	Подп.	Дата

3.4 Параметры файл скрипта

PLFRM – вводится название платформы;

VAR – указываются варианты включения массивов двоичных данных Gbe;

VERSION – указывается строка версии для SMBUS;

BVERSION – указывается версия BCBB;

RESULT_DIR – указывается расположение собранного файла программы;

TYPE – параметр, отвечающий за тип собранной программы. При значении «BIOS» - собирается версия BIOS – LEGACY, при значении «UEFI» - версия UEFI;

PXE – параметр, отвечающий за поддержку программой среды загрузки системы с помощью сетевой карты. Для включения вводится значение «1», для отключения – «0»;

AST – параметр, отвечающий за поддержку программой видеоадаптера. Для включения вводится значение «1», для отключения – «0»;

HT – параметр, отвечающий за поддержку многопоточности hyper-threading. Для включения вводится значение «1», для отключения – «0»;

EFISHELL – параметр, отвечающий за поддержку программой командной оболочки UEFI. Для включения вводится значение «1», для отключения – «0». Только для UEFI;

SBSPLASH – указывается расположение графического файла для загрузочного экрана;

SBBO – указывается расположение файла bootorder.dev, определяющего порядок загрузки программы;

UNLOCK – параметр, отвечающий за блокировку регионов BIOS. Для включения блокировки вводится значение «0», для отключения – «1»;

./build-ext.sh – выполнение основного скрипта сборки программы;

Изм.	Лист	№ докум.	Подп.	Дата

3.5 Запуск скрипта сборки программы

После конфигурирования файла скрипта, следует запустить его путем ввода следующей команды:

```
./cboot.sh
```

После завершения работы скрипта, итоговые файлы будут расположены в папке «Result», в среде сборки программы.

Изм.	Лист	№ докум.	Подп.	Дата

4 УСТАНОВКА ПРОГРАММЫ

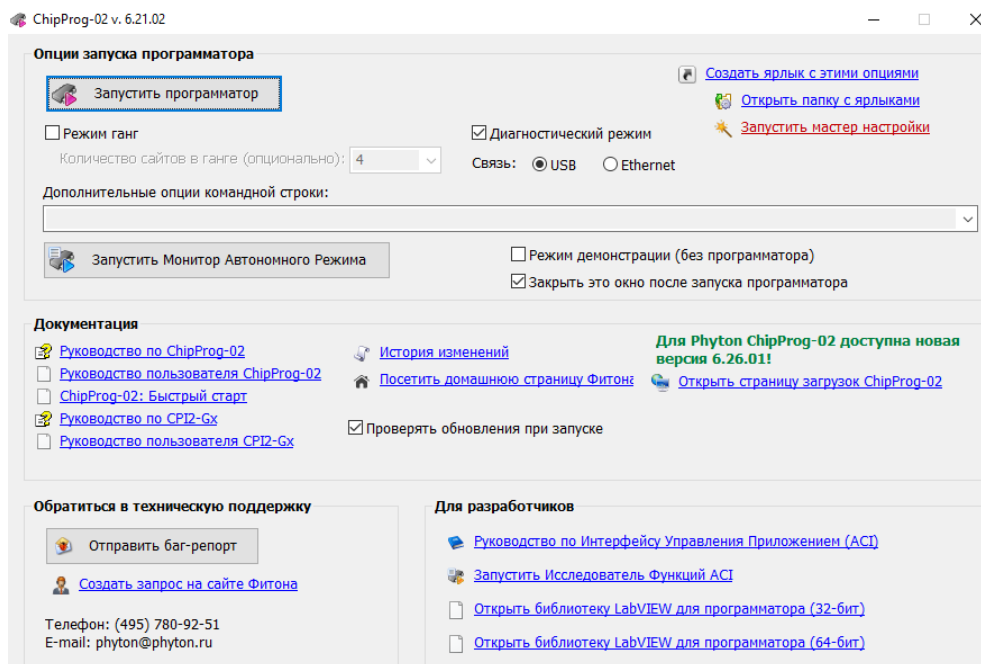
4.1 Аппаратная прошивка

4.1.1 Методика аппаратной прошивки применяется для первичной и повторной записи БСВВ в энергонезависимую память устройства.

4.1.2 Для записи БСВВ необходимо использовать SPI-программатор.

Запись БСВВ происходит следующим образом:

- подключить программатор к разъему для прошивки на плате аппаратной платформы согласно технологической документации;
- с рабочего стола технологического компьютера запустить программу ChipProg-02 6.24.00 и в открывшемся окне нажать «Запустить программатор» согласно рисунку 3;

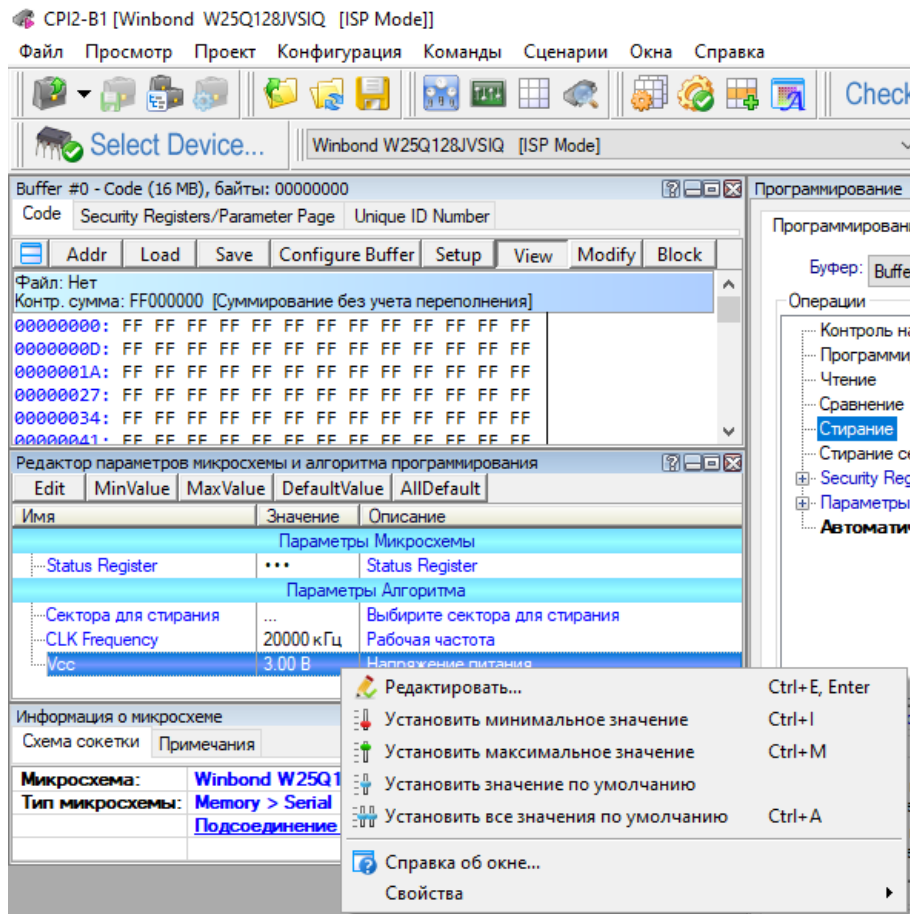


Р и с у н о к 3 – Окно программы ChipProg-02 6.24.00


- в левой части открывшегося окна в разделе «Параметры Алгоритма» щелкнуть на строку «Vcc» правой клавишей мыши для вызова контекстного меню и нажать «редактировать», в появившемся окне ввести значение напряжения, подаваемого на микросхему в размере 3.3 вольта (рисунок 4);

Изм.	Лист	№ докум.	Подп.	Дата

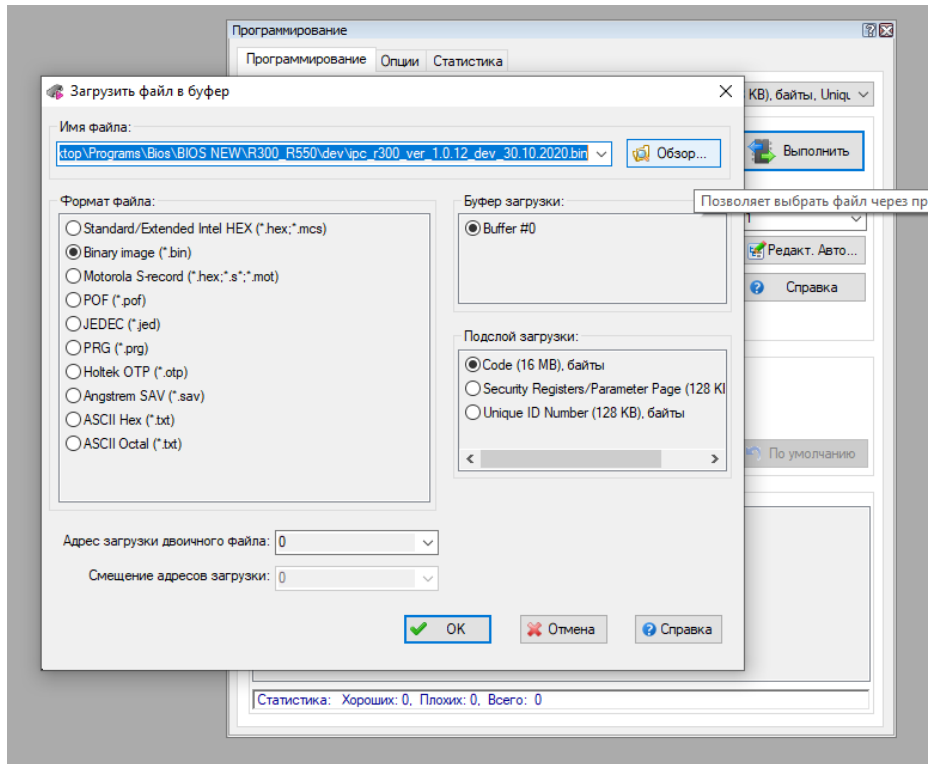
Примечание – Для корректной работы необходимо удостовериться, что в строке «Select Device» выставлена правильная микросхема.



Р и с у н о к 4 – Настройка напряжения

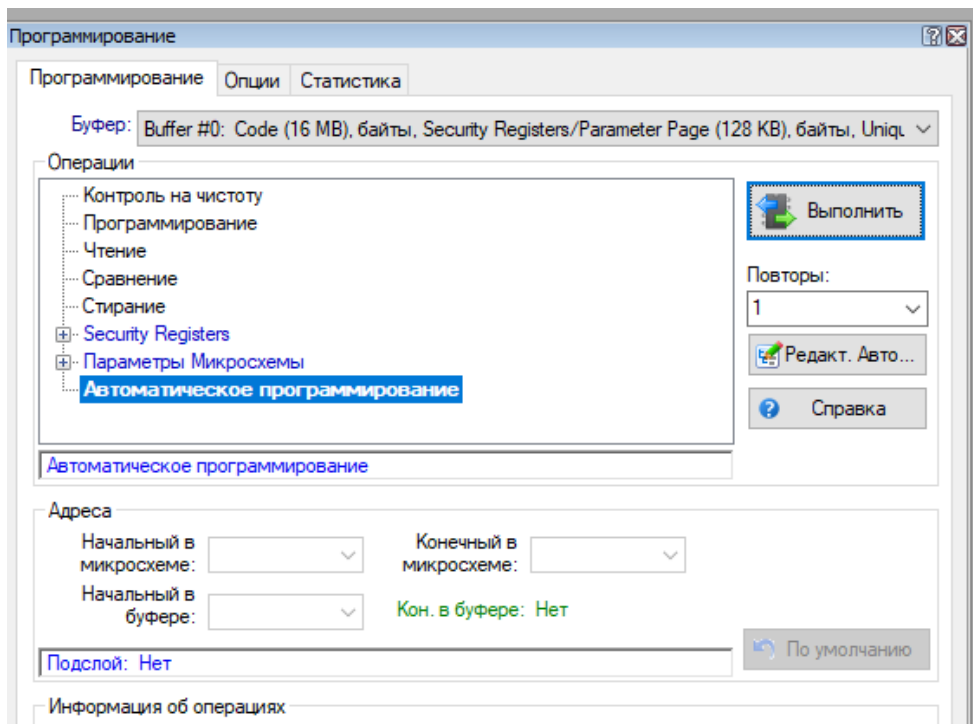
- в окне программы выбрать иконку  (загрузить файл);
- в новом окне нажать «Обзор» согласно рисунку 5 и выбрать необходимый бинарный файл прошивки и нажать «Ок»;

Изм.	Лист	№ докум.	Подп.	Дата



Р и с у н о к 5 – Окно загрузки файла в буфер

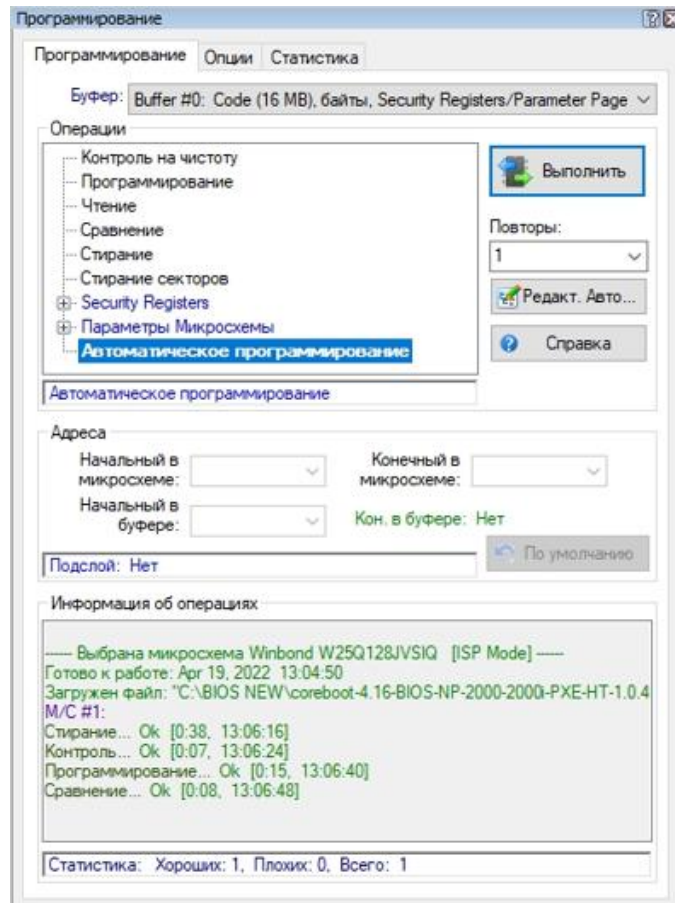
– в окне программы нажать на строчку «Автоматическое программирование», как показано на рисунке 6.



Р и с у н о к 6 - Окно программирования

Изм.	Лист	№ докум.	Подп.	Дата

- нажать «Выполнить» для запуска программирования;
- при успешно выполненной прошивке в окне программирования у операций будет показатель «Ок» и в графе статистика прибавится единица в пункте «Хороших» в соответствии с рисунком 7, иначе повторить операцию программирования.



Р и с у н о к 7 – Окно информации об операциях

4.2 Программная прошивка

4.2.1 Программная прошивка происходит только уже на записанную в энергонезависимую память БСВВ.

4.2.2 Программная прошивка возможна только при условии, что уже записанная в память БСВВ имеет открытые для записи необходимые регионы. Если же необходимые регионы закрыты, то обновление возможно только при помощи программатора.

Изм.	Лист	№ докум.	Подп.	Дата

4.2.3 Метод программной прошивки БСВВ происходит следующим образом:

- подать питание на аппаратную платформу и дождаться включения ОС.
- после запуска ОС необходимо получить права суперпользователя с помощью команды «sudo -s» (рисунок 8).

```
tester@tester:~$ sudo -s  
[sudo] password for tester:  
root@tester:/home/tester# █
```

Р и с у н о к 8 – Получение прав суперпользователя

- скопировать бинарный файл с USB накопителя в предварительно выбранный каталог, из которого далее будет производиться переустановка/обновление БСВВ.

- производим обновление репозитория командой «apt update» и устанавливаем пакет утилиты «flashrom» командой «apt install flashrom».

- далее в каталоге с бинарным файлом нового БСВВ вводим команду «flashrom -p internal -w ./наименование бинарного файла» и утилита производит переустановку/обновление БСВВ (рисунок 9).

Примечание – метод программной прошивки БСВВ применим исключительно для аппаратных платформ типа NP-2XXXi.

```
root@tester:/home/tester/coreboot# flashrom -p internal -w ./coreboot-4.15-NP-2000i-SeaBIOS-HT-VGA.rom  
flashrom v1.2 on Linux 5.4.0-104-generic (x86_64)  
flashrom is free software, get the source code at https://flashrom.org  
  
using clock_gettime for delay loops (clk_id: 1, resolution: 1ns).  
coreboot table found at 0x89c50000.  
Found chipset "Intel C246".  
This chipset is marked as untested. If you are using an up-to-date version  
of flashrom *and* were (not) able to successfully update your firmware with it,  
then please email a report to flashrom@flashrom.org including a verbose (-v) log.  
Thank you!  
Enabling flash write... warning: Setting Bios control at 0xdc from 0x8b to 0x89 failed.  
New value is 0x8b.  
SPI configuration is locked down.  
OK.  
Found Programmer flash chip "Opaque flash chip" (16384 kB, Programmer-specific) mapped at physical address 0x0000000000000000.  
Reading old flash chip contents... done.  
Erasing and writing flash chip... Erase/write done.  
Verifying flash... VERIFIED.  
root@tester:/home/tester/coreboot# █
```

Р и с у н о к 9 – Переустановка/обновление БСВВ

Изм.	Лист	№ докум.	Подп.	Дата

5 ВЫПОЛНЕНИЕ ПРОГРАММЫ

5.1 Загрузка и запуск

Загрузка и запуск БСВВ происходит автоматически после включения питания или «сброса» устройства, на которое она установлена.

5.2 Выполнение программы

5.2.1 Выполнение функций инициализация аппаратной платформы

Выполнение функции инициализации аппаратной платформы происходит автоматически при запуске устройства с установленной БСВВ в процессе обнаружения и настройки внутренних узлов.

5.2.2 Выполнение функции выбора устройства для загрузки ОС

Выполнение функции выбора устройства загрузки ОС может происходить автоматически или самостоятельным выбором пользователя.

5.2.2.1 Выполнение загрузки ОС с устройства «по умолчанию» происходит автоматически по таймауту.

5.2.2.2 Выбор устройства загрузки отличный от «по умолчанию» будет предоставлен пользователю, если было вызвано меню загрузки (boot-меню).

Примечания:

1) В режиме Legacy вызов меню загрузки (boot-меню) происходит при нажатии клавиши «ESC» на клавиатуре при появлении соответствующего сообщения.

2) В режиме UEFI вызов меню загрузки (boot-меню) происходит при выборе соответствующего раздела в меню БСВВ.

5.2.3 Выполнение функции запуска загрузчика ОС

Вне зависимости от выбранного устройства загрузки, если на нем установлена ОС, то БСВВ автоматически запускает и передает управление загрузчику данной ОС.

5.2.4 Настройка порядка загрузки устройств

Изм.	Лист	№ докум.	Подп.	Дата

Выполнение настройки порядка загрузки устройств задается в процессе формирования бинарного файла БСВВ с помощью скрипта сборки.

5.2.5 Поддержки видеointерфейса

Активация видеointерфейса на целевой платформе происходит при его наличии автоматически, при условии включения данной функции на этапе сборки БСВВ.

5.2.6 Выполнение функции поддержки консольного порта

Возможность поддержки реализуется на этапе сборки БСВВ

5.2.7 Функция загрузки через сеть Ethernet (PXE)

Выполнение функции загрузки через сеть Ethernet происходит посредством меню загрузки (boot-меню) и реализуется на этапе сборки БСВВ.

5.2.8 Выполнение функции вывода изображения

Выполнение функции вывода на дисплей «стартовой» загрузки (splash-изображения) в формате BMP/JPEG происходит автоматически при наличии видеointерфейса. При запуске платформы также в текстовой консоли выводится задаваемое на этапе сборки БСВВ изображение текстового баннера.

5.2.9 Поддержка оперативной памяти с функцией коррекцией ошибок

БСВВ позволяет применять в устройствах оперативную память с поддержкой функции коррекции ошибок. Данная функция обеспечивается при сборке БСВВ.

5.2.10 Функция поддержки АПМДЗ

Возможность поддержки реализуется на этапе сборки БСВВ на основании специальных модулей расширения ПМДЗ или вызовом PCI Opinion ROM устройства АПМДЗ.

5.2.11 Функция поддержки EFI Shell

Включение и отключение функции EFI Shell происходит на этапе сборки БСВВ.

Изм.	Лист	№ докум.	Подп.	Дата

5.2.12 Функция поддержки многопоточности

Включение и отключение функции многопоточности процессора происходит на этапе сборки БСВВ.

5.2.13 Поддержка вывода дополнительной информации

При сборке БСВВ при необходимости возможна настройка вывода дополнительной информации, содержащей версию БСВВ, контрольную сумму БСВВ, дату, время и т.д., перед формированием меню загрузки (boot-меню).

5.2.14 Функция поддержки аппаратной виртуализации

При сборке БСВВ возможно включение или отключение по умолчанию возможностей аппаратной виртуализации ЦП.

Изм.	Лист	№ докум.	Подп.	Дата

6 НАСТРОЙКА ПРОГРАММЫ

БСВВ не требует дополнительных настроек для технических и программных средств.

Изм.	Лист	№ докум.	Подп.	Дата

7 ПРОВЕРКА ПРОГРАММЫ

7.1 Описание способов проверки

7.1.1 После успешного запуска устройства должны пройти проверки работы узлов платформы и отобразится меню загрузки (boot-меню) БСВВ.

7.2 Методы прогона

Проверка работоспособности БСВВ проводится следующим образом:

- включить устройство;
- запуск считается успешным, если прошли проверки, появилось меню загрузки (boot-меню) и после выбора устройства загрузки началась загрузка ОС.

Изм.	Лист	№ докум.	Подп.	Дата

8 СООБЩЕНИЯ

При запуске устройства с записанным в память БСВВ происходит проверка аппаратных частей как показано на рисунке 8.

```

0.000000] KERNEL supported cpus:
0.000000] Intel GenuineIntel
0.000000] AMD AuthenticAMD
0.000000] Hygon HygonGenuine
0.000000] Centaur CentaurHauls
0.000000] zhaoxin Shanghai
0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
0.000000] x86/fpu: Supporting XSAVE feature 0x008: 'MPX bounds registers'
0.000000] x86/fpu: Supporting XSAVE feature 0x010: 'MPX CSR'
0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
0.000000] x86/fpu: xstate_offset[3]: 832, xstate_sizes[3]: 64
0.000000] x86/fpu: xstate_offset[4]: 896, xstate_sizes[4]: 64
0.000000] x86/fpu: Enabled xstate features 0x1f, context size is 960 bytes, using 'compacted' format.
0.000000] BIOS-provided physical RAM map:
0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbfff] usable
0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
0.000000] BIOS-e820: [mem 0x000000000000f000-0x000000000000ffff] reserved
0.000000] BIOS-e820: [mem 0x0000000000010000-0x0000000000089c06fff] usable
0.000000] BIOS-e820: [mem 0x0000000000089c07000-0x000000000008f7ffff] reserved
0.000000] BIOS-e820: [mem 0x000000000e000000-0x000000000effffff] reserved
0.000000] BIOS-e820: [mem 0x00000000fc000000-0x00000000fc000fff] reserved
0.000000] BIOS-e820: [mem 0x00000000fe000000-0x00000000fe00ffff] reserved
0.000000] BIOS-e820: [mem 0x00000000fed10000-0x00000000fed17fff] reserved
0.000000] BIOS-e820: [mem 0x00000000fed80000-0x00000000fed83fff] reserved
0.000000] BIOS-e820: [mem 0x00000000fed90000-0x00000000fed91fff] reserved
0.000000] BIOS-e820: [mem 0x00000000feda0000-0x00000000feda1fff] reserved
0.000000] BIOS-e820: [mem 0x0000000100000000-0x0000000086efffff] usable
0.000000] NX (Execute Disable) protection: active
0.000000] SMBIOS 3.0 present.
0.000000] DMI: newplatforms IPC-R1000/IPC-R1000, BIOS 4.15-cc66b56c80862a59117a4582abc8d59f092ac59c 02/20/2022
0.000000] tsc: Detected 3800.000 MHz processor
0.001022] tsc: Detected 3799.900 MHz TSC
0.001026] last_pfn = 0x86e800 max_arch_pfn = 0x400000000
0.001291] x86/PAT: Configuration [0-7]: WB WC UC- UC WB WP UC- WT
0.001626] last_pfn = 0x89c07 max_arch_pfn = 0x400000000
0.007190] check: Scanning 1 areas for low memory corruption
0.007193] Using GB pages for direct mapping
0.007415] RAMDISK: [mem 0x2def5000-0x32f71fff]
0.007420] ACPI: Early table checksum verification disabled
0.007422] ACPI: RSDP 0x000000000000f62b0 000024 (v02 COREV4)
0.007424] ACPI: XSDT 0x00000000089c2c0e0 000064 (v01 COREV4 COREBOOT 00000000 CORE 20210331)
0.007428] ACPI: FACP 0x00000000089c2e1d0 000114 (v06 COREV4 COREBOOT 00000000 CORE 20210331)
0.007431] ACPI: DSDT 0x00000000089c2c280 001f50 (v02 COREV4 COREBOOT 20220108 INTL 20210331)
0.007433] ACPI: FACS 0x00000000089c2c240 000040
0.007435] ACPI: FACS 0x00000000089c2c240 000040
0.007436] ACPI: SSDT 0x00000000089c2e2f0 0013bc (v02 COREV4 COREBOOT 0000002a CORE 20210331)
0.007438] ACPI: MCFG 0x00000000089c2f6b0 00003c (v01 COREV4 COREBOOT 00000000 CORE 20210331)
0.007440] ACPI: LPIT 0x00000000089c2f6f0 000094 (v00 COREV4 COREBOOT 0000002a CORE 20210331)
0.007442] ACPI: APIC 0x00000000089c2f790 0000b2 (v03 COREV4 COREBOOT 00000000 CORE 20210331)
0.007444] ACPI: NHLT 0x00000000089c2f850 000025 (v05 COREV4 COREBOOT 00000000 CORE 00000000)
0.007445] ACPI: DMAR 0x00000000089c2f880 000088 (v01 COREV4 COREBOOT 00000000 CORE 20210331)
0.007447] ACPI: HPET 0x00000000089c2f910 000038 (v01 COREV4 COREBOOT 00000000 CORE 20210331)
0.007449] ACPI: Reserving FACP table memory at [mem 0x89c2e1d0-0x89c2e2e3]
0.007450] ACPI: Reserving DSDT table memory at [mem 0x89c2c280-0x89c2e1cf]
0.007451] ACPI: Reserving FACS table memory at [mem 0x89c2c240-0x89c2c27f]
0.007451] ACPI: Reserving FACS table memory at [mem 0x89c2c240-0x89c2c27f]
0.007452] ACPI: Reserving SSDT table memory at [mem 0x89c2e2f0-0x89c2f6ab]
0.007452] ACPI: Reserving MCFG table memory at [mem 0x89c2f6b0-0x89c2f6eb]
0.007453] ACPI: Reserving LPIT table memory at [mem 0x89c2f6f0-0x89c2f783]
0.007453] ACPI: Reserving APIC table memory at [mem 0x89c2f790-0x89c2f841]
0.007454] ACPI: Reserving NHLT table memory at [mem 0x89c2f850-0x89c2f874]
0.007455] ACPI: Reserving DMAR table memory at [mem 0x89c2f880-0x89c2f907]
0.007455] ACPI: Reserving HPET table memory at [mem 0x89c2f910-0x89c2f947]
    
```

Р и с у н о к 8 – Проверка аппаратных частей устройства

Изм.	Лист	№ докум.	Подп.	Дата

