

УТВЕРЖДЕН
ТВГИ.00406-01-УД

ПРОГРАММА НАЧАЛЬНОГО СТАРТА МП «ЭЛЬБРУС-2С3»

Руководство оператора

Листов 113

2023

Литера

АННОТАЦИЯ

Настоящий документ является руководством оператора по работе с программой начального старта МП «Эльбрус-2С3», предназначенной для первоначальной инициализации вычислительных комплексов и вычислительных модулей на базе микропроцессора Е2С3, выполнения доверенной загрузки (если доверенная загрузка необходима) и запуска операционной системы (далее по тексту — ОС). Доверенная загрузка реализована посредством передачи управления аппаратно-программному модулю доверенной загрузки АПМДЗ-И/Э КБДЖ.468243.173 (далее по тексту — АПМДЗ).

СОДЕРЖАНИЕ

1. Назначение программы.....	5
2. Условия выполнения программы, выполнение программы и сообщения оператору.....	6
2.1. Условия выполнения программы.....	6
2.2. Выполнение программы.....	6
2.3. Сообщения оператору.....	7
2.3.1. Общие сведения.....	7
2.3.2. Схемы запуска целевых программ.....	8
2.3.3. Диалог.....	8
2.4. Тесты, исполняемые на этапе начальной загрузки.....	10
3. Условия применения.....	11
3.1. Требования к техническим (аппаратным) средствам.....	11
3.2. Требования к программным средствам, требования и условия организационного, технического и технологического характера.....	11
4. Список команд верхней консоли.....	12
4.1 Команды верхней консоли.....	12
4.2. Подробное описание команд верхней консоли.....	13
5. Расширенный список команд верхней консоли.....	15
5.1. Подробное описание расширенного списка команд верхней консоли.....	16
6. Список команд нижней консоли.....	25
6.1. Подробное описание команд нижней консоли.....	25
7. Список команд программного модуля специальных настроек.....	30
7.1. Подробное описание команд программного модуля специальных настроек...36	
7.1.1. Команда вывода справочной информации по командам настроек.....	36
7.1.2. Команда перехода в меню загрузки ОС.....	37
7.1.3. Команды определения устройства загрузки.....	37
7.1.4. Команды настройки ввода/вывода сообщений ПНС через последовательный порт.....	37
7.1.5. Команды конфигурирования и вывода настроек ОЗУ.....	39

<u>7.1.6. Команды настройки доверенной загрузки.....</u>	<u>39</u>
<u>7.1.7. Команды установки времени.....</u>	<u>43</u>
<u>7.1.8. Команды настройки IDE-интерфейсов и SATA-интерфейсов.....</u>	<u>49</u>
<u>7.1.9. Команды настройки онбордного Ethernet-канала.....</u>	<u>51</u>
<u>7.1.10. Команды настройки загрузки ОС по сети.....</u>	<u>56</u>
<u>7.1.11. Команды настройки вывода на монитор.....</u>	<u>58</u>
<u>7.1.12. Команды настройки КПИ.....</u>	<u>61</u>
<u>7.1.13. Команды настройки USB-контролеров.....</u>	<u>62</u>
<u>7.1.14. Команды конфигурирования NVME.....</u>	<u>65</u>
<u>7.1.15. Команды запуска тестов из ПНС.....</u>	<u>66</u>
<u>7.1.16. Команды идентификации материнской платы и установи FRUID.....</u>	<u>67</u>
<u>7.1.17. Команды настроек чипсета.....</u>	<u>70</u>
<u>7.1.18. Команды конфигурирования интерфейса I2C.....</u>	<u>106</u>
<u>7.1.19. Команды дополнительных настроек.....</u>	<u>109</u>
<u>7.1.20. Команда Перезапуск.....</u>	<u>110</u>
<u>7.1.21. Команда для перехода в программный модуль специальных тестов.....</u>	<u>111</u>
<u>Перечень сокращений.....</u>	<u>112</u>

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

Программа начального старта МП «Эльбрус-2С3» (далее по тексту — ПНС) предназначена для:

- первоначального тестирования и инициализации вычислительных систем на базе микропроцессора E2C3;
- доверенной загрузки и запуска ОС;
- проверок и настроек вычислительных систем на базе микропроцессора E2C3, предшествующих эксплуатации.

Выполнение любых действий с ПНС, не описанных в настоящем документе, запрещено.

П р и м е ч а н и е . Ограничение доступа пользователя к функциям ПНС, предназначенным исключительно для администратора безопасности информации, возлагается на администратора безопасности информации, осуществляющего обслуживание вычислительных систем на базе микропроцессора E2C3.

2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ, ВЫПОЛНЕНИЕ ПРОГРАММЫ И СООБЩЕНИЯ ОПЕРАТОРУ

2.1. Условия выполнения программы

ПНС выполняется на вычислительных системах, построенных на базе микропроцессора E2C3.

ПНС предоставляет разные возможности администратору безопасности информации и пользователю.

Пользователю необходимо убедиться, что АПМДЗ находится в режиме управления в соответствии с руководством КБДЖ.468243.173РЭ1, в противном случае на экране появится сообщение:

```
WARNING: APMDZ module is Off or init-mode
```

```
Would you like to run APMDZ Linux ? (y/n)
```

не нажимая кнопок, пользователю следует обратиться к администратору безопасности информации.

Администратору безопасности информации для получения сведений по настройке АПМДЗ и работе с ним необходимо ознакомиться с руководством КБДЖ.468243.173РЭ2.

2.2. Выполнение программы

ПНС запускается при включении питания или перезапуске вычислительных систем, построенных на базе микропроцессора E2C3

Схема работы ПНС, запускаемой с АПМДЗ, находящимся в режиме управления, состоит в следующем:

- старт загрузочного процессора в одноядерном режиме;
- выполнение необходимых подпрограмм самотестирования оборудования;

-инициализация устройств, необходимых для загрузки программного кода АПМДЗ (оперативной памяти, других ядер загрузочного процессора и других процессоров);

-передача управления АПМДЗ для реализации возможности доверенной загрузки сконфигурированной ОС.

ПНС, запускаемая с АПМДЗ, находящимся в режиме инициализации, позволяет:

-изменить параметры ПНС (подробная информация по параметрам приведена в п.2.3.3 настоящего документа);

-вывести на экран текущие параметры ПНС;

-запустить файл, в котором может быть сохранена конфигурация ПНС (boot.conf).

2.3. Сообщения оператору

2.3.1. Общие сведения

ПНС выводит информационные сообщения в последовательный порт, а также, с момента инициализации видеоадаптера — на подключенный к нему видеомонитор, если он присутствует (далее обозначаются как «внешний терминал»).

Для обмена информацией по последовательному порту к вычислительной системе, построенной на базе микропроцессора E2C3 должен быть подключен знаковый терминал или эмулятор терминала со следующими настройками:

-скорость — 115200 бит/сек;

-кадр — 8 бит;

-четность — нет;

-стоп-бит — 1;

-регулировка потока - нет.

По мере тестирования устройств на внешний терминал выдаются информационные сообщения.

После проведения необходимых для запуска АПМДЗ операций произойдет передача управления дальнейшей загрузкой АПМДЗ или безусловная загрузка ОС (в зависимости от настроек, сделанных администратором безопасности).

2.3.2. Схемы запуска целевых программ

Порядок загрузки целевых программ (ОС) определяется администратором безопасности информации.

2.3.3. Диалог

Возможность изменения параметров загрузки и старта программ пользователем не предусмотрена.

Администратору безопасности информации при условии запуска ПНС с АПМДЗ, находящимся в режиме инициализации, доступны следующие команды:

«r» или «s» — загрузка из энергозависимой памяти и старт выбранного конфигурационного файла с кодом загружаемой программы (ОС);

«c» — режим изменения параметров загрузки и старта;

«u» — вывод существующих на данный момент параметров загрузки и старта на экран;

«d» — вывод подключенных к панели дисков (устройств для загрузки) и их разделов на монитор;

«m» — сохранение выбранных параметров загрузки и старта в энергонезависимую память;

«b» — режим старта с использованием конфигурационного файла boot.conf;

«w» — вывод на экран краткой информации об аппаратной конфигурации устройства, на котором запущена ПНС;

«P» — установка режима старта устройства, на котором запущена ПНС, по кнопке включения питания;

«B» — установка режима старта устройства, на котором запущена ПНС, по подаче питания (без использования кнопки включения питания);

«T» — запуск теста индикаторов устройства, на котором запущена ПНС;

«t» — переход в режим тестирования и отладки;

«W» — включение/выключение ограничения по времени, которое дается на полное выполнение ПНС;

«K» — включение/выключение компенсационного режима питания и температуры;

«M» — изменение формата вывода информации на экран;

«`», «~» — переход в режим расширенного списка команд (см. раздел 3 приложения 3 к настоящему документу);

«N» — установка следующего включения без очистки NVRAM.

К параметрам ПНС относятся следующие переменные:

`drive_number` — номер устройства загрузки;

`drive_label` — идентификатор устройства загрузки;

`partition_number` — номер раздела на выбранном устройстве загрузки;

`file system id` — идентификатор файловой системы на выбранном устройстве загрузки, однозначно определяющий номер раздела (`uuid`);

`command_string` — командная строка, передаваемая ядру ОС;

`filename` — имя файла загружаемой программы;

`initrdfilename` — имя файла `initrd*` (необязательный параметр, может использоваться для старта ядра ОС).

* `initrd` — Initial RAM Disk — образ виртуального диска, который может использоваться для загрузки ОС, например, для установки ОС с CD/DVD-ROM.

Для изменения параметра необходимо ввести его значение (строку для параметров `command_string`, `filename` и `initrdfilename` и число для прочих параметров) и нажать клавишу «Enter». Нажатие клавиши «Enter» без ввода

значения устанавливает параметру пустое значение или ноль, соответственно. Нажатие клавиши «Esc» оставляет прежнее значение параметра.

Для облегчения ввода параметров filename и initrdfilename поддерживается поиск по начальным буквам имени файла.

Параметры drive_label и file system id выводятся в качестве информационных, возможность их изменить не поддерживается.

2.4. Тесты, исполняемые на этапе начальной загрузки

В процессе загрузки автоматически производится тестирование аппаратного обеспечения, необходимого для работы изделия:

- работоспособности кэш-памяти процессора;
- работоспособности контроллеров оперативной памяти;
- работоспособности связей «процессор-процессор» и «процессор-контроллер периферийных интерфейсов»;
- работоспособности АПМДЗ;
- работоспособности периферийного оборудования.

При неудовлетворительном результате тестирования, не позволяющем дальнейшую загрузку системы, на внешний терминал выдаются диагностические сообщения об ошибке, и процесс загрузки останавливается.

3. УСЛОВИЯ ПРИМЕНЕНИЯ

3.1. Требования к техническим (аппаратным) средствам

ПНС выполняется на однопроцессорных, двухпроцессорных и четырехпроцессорных вычислительных устройствах на основе многоядерного микропроцессора архитектуры «Эльбрус» отечественной разработки.

3.2. Требования к программным средствам, требования и условия организационного, технического и технологического характера

ПНС функционирует самостоятельно.

Ограничение доступа непривилегированного пользователя к функциям ПНС, предназначенным исключительно для администратора безопасности информации (АПМДЗ в режиме инициализации), возлагается на администратора безопасности информации, осуществляющего обслуживание вычислительного устройства или системы, построенной на его базе, на объекте эксплуатации.

4. СПИСОК КОМАНД ВЕРХНЕЙ КОНСОЛИ

4.1 Команды верхней консоли

Команды верхней консоли становятся доступны оператору, если сразу после включения питания ВК, или сразу после перезапуска ВК, зажать на штатной клавиатуре или на клавиатуре, подключённой к ВК по последовательному порту ЭВМ, клавишу «h» в латинском регистре.

В случае нажатия клавиши «h» будет выведен список возможных команд.

Список приведён в таблице 1.

Т а б л и ц а 1 — Список команд

Горячая клавиша	Команда
ESC	ESC BOOT: Set Default Params
m	m Mem: Set Default Params
`	` CPU: Set Limit Freq
v	v Verbose y/n flag (is_watch)
s	s Silence y/n flag
t	t Threads y/n Multi Threading
x	x eXtra Set DDR4/PHY,EIOH,MB/NB,MGA,IPCC ...
h	h Help Show this list of commands
.	Please press any known key (except 'h') to Accept or '.' to Cancel

4.2. Подробное описание команд верхней консоли

Т а б л и ц а 2 — Описание команд

Горячая клавиша	Описание
<i>ESC</i>	Set Default Params
	Установка значений инженерных параметров ПНС по умолчанию.
	Нажать клавишу [Esc] CMOS kpi.verbose_flag:0 Invert ? (y or n)
m	Mem: Set Default Params
	Установка/Отмена значений параметров ОЗУ по умолчанию.
	Нажать клавишу [m]
`	CPU: Set Limit Freq
	Установка верхней границы тактовой частоты CPU
	Нажать клавишу [`]
v	Verbose y/n flag (is_watch)
	Вкл/Выкл режима более детальной выдачи информационных сообщений, поступающих от ПНС
	Нажать клавишу [v]
s	Silence y/n flag
	Вкл/Выкл режима «Нет выдачи сообщений от ПНС»
	Нажать клавишу [s]
t	Threads y/n Multi Threading
	Вкл/Выкл режима «Параллельная инициализация ядер»
	Нажать клавишу [t]
x	eXtra Set DDR4/PHY, EIOH, MB/NB, MGA, IPCC ...
	Вывод расширенного списка команд верхней консоли для установки параметров
	Нажать клавишу [x]

Окончание таблицы 2

h	Help Show this list of commands
	Выдача списка команд верхней консоли с кратким пояснением
	Нажать клавишу [h]
.	Please press any known key (except 'h') to Accept or '.' to Cancel
	Выйти из верхней консоли
	Нажать клавишу [.]

5. РАСШИРЕННЫЙ СПИСОК КОМАНД ВЕРХНЕЙ КОНСОЛИ

Т а б л и ц а 3 — Расширенный список команд

Горячая клавиша	Описание
0	y/n Verbose Advanced HW Diagnos
1	y/n ETH Wait: Speed, Looking, AOE
2	y/n X86EMU_VIDEO_INIT En
3	y/n HW_CTRL_MGA25_PRIO_REQ_DSBL
4	y/n PMC_regs_Keep_Wr En
5	Set PWM (FANs)
6	Sel MB ID
7	y/n IP-Links ReInit
8	Sel SPI Freq for FLASH & CMOS
	DDR4, PHY
a	Set HMU_MCEN (Slots per CPUs)
b	Calc CPU MC Freq (PLL MC)
c	y/n PHY Skip Train Input
d	y/n PHY Print Debug MSGs
e	y/n PHY Retention En Input (S3)
f	y/n PHY Init Sequence Num
g	y/n Increased Mem BandWidth
h	y/n Data Bus Inverse
i	y/n PHY Initiated Update
j	y/n PHY Update Threshold
k	y/n VRefDQ Change En
l	y/n DDR Eye Optimize
m	Set DRAM Scrub Time
n	y/n DRAM Gear Down
o	y/n PHY yIs2Ttiming[0]
p	y/n MC/PHY BroadCast
q	y/n Write CRC
r	y/n C/A Parity
	MB/NB, SYSTEM
A	y/n MIN Conf (CPUs=1 ...)

Окончание таблицы 3

L	Set Limit PCIM,IO,IOAPIC,MLO area
Z	Exit Menu

5.1. Подробное описание расширенного списка команд верхней консоли

Т а б л и ц а 4— Подробное описание расширенного списка команд

Команда	Описание
0	y/n Verbose Advanced HW Diagnos
	Вкл/Выкл режима вывода в лог расширенной диагностики
	Нажать клавишу [0] (ноль) Hit any Menu-key: 0 CMOS kpi.verbose_flag:0 Invert ? (y or n)
1	y/n ETH Wait: Speed, Looking, AOE
	Вкл/Выкл режима ожидания установки скорости базового Ethernet-интерфейса и поиска AoE сервера
	Нажать клавишу [1] Hit any Menu-key: 1 CMOS kpi.wait_eth:0 Invert ? (y or n)
2	2 y/n X86EMU_VIDEO_INIT En
	Установка/отмена эмуляции видеобиоса X86
	Нажать клавишу [2] Hit any Menu-key: 2 Current x86emu_video_init_enable Flag = 1 Invert ? (y or n)

Продолжение таблицы 4

3	<p>y/n HW_CTRL_MGA25.PRIO_REQ_DSBL</p> <p>Вкл/Выкл бита HW_CTRL_MGA25.PRIO_REQ_DSBL</p> <p>Нажать клавишу [3]</p> <p>Hit any Menu-key: 3 No sense to change MGA25.PRIO_REQ_DSBL bit since MGA2XX is not enabled Current HW_CTRL_MGA25.PRIO_REQ_DSBL: 0</p> <p>Invert ? (y or n)</p>
4	<p>y/n PMC_regs_Keep_Wr En</p> <p>Установка/отмена режима сохранения регистров PMC</p> <p>Нажать клавишу [4]</p> <p>Hit any Menu-key: 4 CMOS nbsr.pmc_regs_wren_flag: 1</p> <p>Invert ? (y or n)</p>
5	<p>Set PWM (FANs)</p> <p>Вызов диалога установки скорости вращения вентиляторов</p> <p>Нажать клавишу [5]</p> <p>Hit any Menu-key: 5 Press Enter for Dflt CPUs Mask 0xF or type 0 to finish or type hex mask of CPUs : 0xf Press Enter for Dflt FANs Mask 3 or type 0 to finish or type hex mask of FANs : 0x3 Enter PWM mode: 'd' for "Disabled", 'f' for "Fixed value", 'l' for "LUT": f Fixed value Enter fixed PWM value = 0xf Fixed PWM value = 0xF (11.7 %) saved in CMOS for selected CPUs and FANs</p> <p>Other Tune ? (y or n)</p>

Продолжение таблицы 4

6	<p>Sel MB ID</p> <hr/> <p>Установить ID материнской платы</p> <hr/> <p>Нажать клавишу [6]</p> <pre>Hit any Menu-key: 6 MB Version:0x60 MB_DETECT Change ? (y or n) y MB Version: 0 MB_DETECT 1 MB_1E16_uATX 2 MB_2E16_MSCTX 3 MB_2E16C_... 4 MB_4E16C-XL 5 MB_E16C_eATX_FRAM Set MB Version:4 Exit ? (y or n)</pre>
7	<p>y/n IP-Links ReInit</p> <hr/> <p>Установка/отмена режима повторной перетренировки</p> <hr/> <p>Нажать клавишу [7]</p> <pre>Hit any Menu-key: 7 CMOS nbsr.ipl_reini_flag: 1 Invert ? (y or n)</pre>
8	<p>Sel SPI Freq for FLASH & CMOS DDR4, PHY</p> <hr/> <p>Установить частоту SPI для взаимодействия с FLASH & CMOS DDR4, PHY</p> <hr/> <p>Нажать клавишу [8]</p> <pre>Hit any Menu-key: 8 CPU0.0: MB_show_spi_freq MB_show_spi_freq Cmos=2 Dflt=3 User= !present Change ? (y or n) y SPI Freq: 0 50.0 MHz 1 25.0 MHz 2 12.5 MHz 3 6.25 MHz 4 5.00 MHz 5 2.94 MHz 6 2.00 MHz 7 1.00 MHz 8 !present Set SPI Freq:1 Exit ? (y or n)</pre>

Продолжение таблицы 4

a	Set HMU_MCEN (Slots per CPUs)											
	<p>Установить маскирование каналов памяти</p> <p>Нажать клавишу [a]</p> <p>Hit any Menu-key: a Type CPU# 0-3: 0 Current HMU_MCEN.[63:0]=0x101010101010101 Type Slots# 1-2: 1 Hit ENTER for DFLT ChMsk0xFF or type ChMsk: 0xff CPU0 Slots:1 ChMsk=0xFF</p> <p>Save Tune ? (y or n)</p>											
b	Calc CPU MC Freq (PLL MC)											
	<p>Установить маскирование каналов памяти</p> <p>Нажать клавишу [b]</p> <p>Hit any Menu-key: b E.g: 350000000-1350, 400000000-1600, 450000000-1800, 525000000-2100, 600000000-2400 650000000-2600, 675000000-2700, 700000000-2800, 725000000-2900, 750000000-3000, 800000000-3200 Hit Enter to DFLT MC Freq 337500000 or 0 to finish or type Freq in Hz : 337500000</p> <table border="1"> <thead> <tr> <th>#</th> <th>NF</th> <th>NR</th> <th>OD</th> <th>Fvco</th> <th>error</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0x3600000000</td> <td>1</td> <td>8</td> <td>2.700000e+09</td> <td>+0000.000000</td> </tr> </tbody> </table> <p>Calc tunes: clkf 0x3600000000 , clkr 0 , clkod 7 Press Enter for Dflt CPUs Mask 0xF type 0 to finish type hex CPUs mask: 0x0 You've entered empty CPUs mask ...</p> <p>Other Tune ? (y or n)</p>	#	NF	NR	OD	Fvco	error	1	0x3600000000	1	8	2.700000e+09
#	NF	NR	OD	Fvco	error							
1	0x3600000000	1	8	2.700000e+09	+0000.000000							
c	y/n PHY Skip Train Input											
	<p>Вкл/Выкл режим тренировки физ.уровней памяти</p> <p>Нажать клавишу [c]</p> <p>Hit any Menu-key: c CMOS phy.skip_train_flag:0</p> <p>Invert ? (y or n)</p>											

Продолжение таблицы 4

d	y/n PHY Print Debug MSGs
	Вкл/Выкл режима печати отладочных сообщений
	Нажать клавишу [d] Hit any Menu-key: d CMOS phy.debug_flag:0 Invert ? (y or n)
e	y/n PHY Retention En Input (S3)
	Вкл/Выкл режима восстановления S3
	Нажать клавишу [e] Hit any Menu-key: e CMOS phy.yRetEn:0 Invert ? (y or n)
f	y/n PHY Init Sequence Num
	Вкл/Выкл режима инициализации только одного физ.уровня
	Нажать клавишу [f] Hit any Menu-key: f CMOS phy.PhyInitSequenceNum:0 Invert ? (y or n)
g	y/n Increased Mem BandWidth
	Вкл/Выкл режим увеличенной пропускной способности Mem
	Нажать клавишу [g] Hit any Menu-key: g CMOS: increased mem bandwidth: 1 Invert ? (y or n)
h	y/n Data Bus Inverse
	Вкл/Выкл режима DBI (Data Bus Inversion)
	Нажать клавишу [h] Hit any Menu-key: h CMOS mem.yDataBusInvers:0 Invert ? (y or n)

Продолжение таблицы 4

i	y/n PHY Initiated Update
	Вкл/Выкл режима PHY Initiated Update
	Нажать клавишу [I] Hit any Menu-key: i CMOS mem.yPHY_InitUpdate:1 Invert ? (y or n)
j	y/n PHY Update Threshold
	Вкл/Выкл режима PHY Update Threshold
	Нажать клавишу [j] Hit any Menu-key: j CMOS mem.yPHY_UpdateThreshold:0 Invert ? (y or n)
k	y/n VRefDQ Change En
	Вкл/Выкл возможности изменения VRefDQ
	Нажать клавишу [j] Hit any Menu-key: k CMOS mem.yDDR4_VREFDQ_CHANGE_EN:0 Invert ? (y or n)
l	y/n DDR Eye Optimize
	Вкл/Выкл режима DDR Eye Optimization
	Нажать клавишу [L] Hit any Menu-key: l CMOS mem.yDDR_EyeOptimiz:1 Invert ? (y or n)

Продолжение таблицы 4

m	Set DRAM Scrub Time
	Установка параметра DRAM Scrub Time
	Нажать клавишу [M] Hit any Menu-key: m CMOS DRAM Scrub Time:0 Change ? (y or n) y Opt DST: 0 Disabled 1 1 hour 2 4 hours 3 8 hours 4 16 hours 5 24 hours 6 48 hours Set your DST:1 Exit ? (y or n)
n	y/n DRAM Gear Down
	Вкл/Выкл режима DRAM Gear Down
	Нажать клавишу [m] Hit any Menu-key: n CMOS mem.yGearDown:0 Invert ? (y or n)
o	y/n PHY yIs2Ttiming[0]
	Установка/отмена режима Timing2T
	Нажать клавишу [m] Hit any Menu-key: o CMOS phy.yIs2Ttiming:0 Invert ? (y or n)

Продолжение таблицы 4

p	y/n MC/PHY BroadCast
	Вкл/Выкл режима BroadCast (широковещательно)
	Нажать клавишу [p] Hit any Menu-key: p CMOS mem.yBC:0 Invert ? (y or n)
q	y/n Write CRC
	Вкл/Выкл режима Write CRC
	Нажать клавишу [q] Hit any Menu-key: q CMOS mem.yWrCRC:0 Invert ? (y or n)
r	y/n C/A Parity MB/NB, SYSTEM
	Вкл/Выкл режима C/A Parity
	Нажать клавишу [p] Hit any Menu-key: r CMOS mem.yCAPM:0 Invert ? (y or n)

Окончание таблицы 4

A	y/n MIN Conf (CPUs=1 ...)
	Вкл/Выкл режима минимальной конфигурации оборудования
	Нажать клавишу [A] Hit any Menu-key: A Exit ? (y or n)
L	Set Limit PCIM, IO, IOAPIC, MLO area
	Установить ограничение области PCIM, IO, IOAPIC, MLO
	Нажать клавишу [L] Hit any Menu-key: L Curr.assigned [PCIM, PCIIO, IOAPIC, MLO] space area's Plan = 3 Change ? (y or n) y Please select [PCIM, PCIIO, IOAPIC, MLO] space areas Plan 0 - MAX_FOR_MLO_MODE 1 - MAX_FOR_PCI_MODE 2 - BIG_FOR_MLO_MODE 3 - BIG_FOR_PCI_MODE 4 - USER_FOR_PCI_MLO_MODE Press the your selected type: 0 Exit ? (y or n)
Z	Exit Menu
	Выход из программного модуля верхней консоли
	Нажать клавишу [Z]

6. СПИСОК КОМАНД НИЖНЕЙ КОНСОЛИ

BOOT SETUP

Press command letter, or press 'h' to get help

:h

```

                                HELP
'p' or 's'   - Load and Start file
'O'          - Load and Start micro-OS-proxy
'I'          - Show Boot-Info & FRUID Board Info Area
'D'          - Download file from Disk to Flash:
'U'          - Upload file from Flash to Disk:
'l'          - Set IP-Links Poweron Mask. 7-All-On, 0-All-Off
'd'          - show Disks and partitions
'u'          - show cUrrent params
'c'          - change boot params
'm'          - save params to NVRAM
'b'          - start Boot.conf menu
't'          - run basic test suite
'T'          - enter debug tests menu
'F'          - enable CPU auto Thermo-Control (with floating divF)
'f'          - disable CPU auto Thermo-Control (fix divF_limit_lo)
'w',        - show green zone : watch summary of the Boot's operation
'e', '`', '~' - enter enhanced cmd mode

```

6.1. Подробное описание команд нижней консоли

h

Комментарий: Press command letter, or press 'h' to get help

Описание: повторный вызов перечня команд нижней консоли

Пример: Нажатие клавиши «h»

p

Комментарий: Load and Start file

Описание: Пуск (Старт) ОС

Пример: Нажатие клавиши «p»

s

Комментарий: Load and Start file

Описание: Пуск (Старт) ОС

Пример: Нажатие клавиши «р»

О

Комментарий: Load and Start micro-OS-proxy

Описание: Пуск (Старт) Мини ОС

Пример: Нажатие клавиши «О»

I

Комментарий: Show Boot-Info & FRUID Board Info Area

Описание: Пуск (Старт) Мини ОС

Пример: Нажатие клавиши «I»

D

Комментарий: Download file from Disk to Flash:

Описание: запись файла с диска в раздел на флэш

Пример: Нажатие клавиши «D»

U

Комментарий: Upload file from Flash to Disk:

Описание: сохранение содержимого раздела флэши в файл на диск

Пример: Нажатие клавиши «U»

1

Комментарий: *Set IP-Links Poweron Mask. 7-All-On, 0-All-Off*

Описание: задание маски *IP-Links*

Пример: *Нажатие клавиши «l»*

d

Комментарий: *show Disks and partitions*

Описание: вывод перечня подключённых внешних носителей

Пример: *Нажатие клавиши «d»*

u

Комментарий: *show cUrrrent params*

Описание: вывод текущих настроек

Пример: *Нажатие клавиши «u»*

c

Комментарий: *change boot params*

Описание: переход к установке/изменению настроек загрузки ОС

Пример: *Нажатие клавиши «c»*

m

Комментарий: *save params to NVRAM*

Описание: сохранение настроек загрузки ОС в *NVRAM*

Пример: *Нажатие клавиши «m»*

b

Комментарий: *-start Boot.conf menu*

Описание: команда на выполнение загрузки ОС

Пример: *Нажатие клавиши «b»*

t run basic test suite

Комментарий: *run basic test suite*

Описание: команда на выполнение системного теста

Пример: *Нажатие клавиши «t»*

T

Комментарий: *enter debug tests menu*

Описание: вызов меню настройки системных тестов

Пример: *Нажатие клавиши «T»*

F

Комментарий: *enable CPU auto Thermo-Control (with floating divF)*

Описание: установка защиты от перегрева CPU

Пример: *Нажатие клавиши «F»*

f

Комментарий: *disable CPU auto Thermo-Control (fix divF_limit_lo)*

Описание: отмена защиты от перегрева CPU

Пример: *Нажатие клавиши «f»*

w

Комментарий: *вывод итогов конфигурирования ВК перед загрузкой ОС*

Описание: *show green zone : watch summary of the Boot's operation*

Пример: *Нажатие клавиши «w»*

e

Комментарий: *enter enhanced cmd mode*

Описание: *переход в программный модуль специальных настроек*

Пример: *Нажатие клавиши «e»*

7. СПИСОК КОМАНД ПРОГРАММНОГО МОДУЛЯ СПЕЦИАЛЬНЫХ НАСТРОЕК

Существуют команды расширенного списка, использование которых допускается исключительно уполномоченными лицами до передачи изделия с ПНС на объект эксплуатации. Ниже приведен перечень и описание этих команд.

Т а б л и ц а 5— Встроенная подсказка программного модуля специальных настроек

Ключевые слова, мнемоника команд и комментарии	
help	Brief help. WARN: Fields enabled to set in CPU registers are marked with an *
boot	- Enter boot menu
set	- Allows to set parameters:
boot_device	- Boot device
sp	- Serial ports settings:
output_dis	- Serial port output disable flag
input_dis	- Serial port input disable flag
memory	- Memory:
limit	- Limited size
csum	- Control sum settings:
calc	- Calculate and print control sums, when loading files
check	- Check loaded file
save	- Save control sum of next loaded file to NVRAM
kernel_check	- Kernel check loaded file
apmdz	- Turn on/off usage of "Echelon-E" APMDZ
rtc	- real time clock settings:
time	- set/show current time
date	- set/show current date
weekday	- set/show current day of the week
password	- Enter password menu:
enable	- Allows to enable new password
disable	- Allows to disable password
change	- Allows to set new password
mode	- Password check mode (load/modify or modify only)
serial_num	- Machine serial number
drive_init_delay	- Delay in seconds, before any drive init-starts
mb_version	- Machine version magic (arch_type)
cpu_type	- CPU type
mc	- Memory controller parameters:
freq	- MC Freq Tunes
MemAlertEn	- DDR4PHY MemAlertEn
odti0	- DDR4PHY ODTImpedance[0]

Продолжение таблицы 5

	txi0	- DDR4PHY TxImpedance[0]
	DisDynAdrTri	- DDR4PHY DisDynAdrTri
	mc_pll	- Tune MC Frequency by PLL settings (clkf,clkr,clkod)
	ahci	- AHCI configuration:
	controller	- tune AHCI controller:
	enable	- enable controller or port
	disable	- disable controller or port
	max_speed	- AHCI speed limit
	ide	- IDE configuration:
	mcst	- MCST IDE configuration:
	maxdma	- The highest DMA mode
	controller	- tune IDE controller:
	enable	- enable controller or drive
	disable	- disable controller or drive
	eth	- Ethernet Controllers configuration:
	100	- ETH100 configuration:
	mac	- MAC address base
	speed	- Data speed
	mode	- Normal/Loopback
	full	- Init Block Full-Bit
	1000	- ETH1000 configuration:
	mac	- MAC address base
	speed	- Data speed
	mode	- Normal/Loopback
	full	- Init Block Full-Bit
	controller	- tune ETH controller:
	enable	- enable ETH controller
	disable	- disable ETH controller
	aoe	- ATA over Ethernet (AoE) settings:
	server	- initial requests settings:
	mac	- destination MAC address
	major	- destination AoE major address
	minor	- destination AoE minor address
	timeout	- timeout settings:
	break	- timeout (ms) to break exchange with AoE server
	video	- tune video controllers:
	output	- choose video output
	dc	- tune display controller
	dc_mode	- tune display controller
	dc_lvds	- tune display controller for lvds out
	dc_lvds_mode	- tune display controller for lvds out
	vga	- deprecated parameter, use set/get video

Продолжение таблицы 5

	pcie	- tune PCIe bridges:
	baud	- configure PCIe baud
	eioh	- tune EIOH:
	slink	- Conf SLink Freq

Продолжение таблицы 5

	usb	- USB configuration:
	xhci	- USB XHCI (USB 3) configuration:
	port_power_delay	- delay after switching ports power on
	controller	- usb controller control:
	enable	- enable usb controller
	disable	- disable usb controller
	ehci	- USB EHCI (USB 2.0) configuration:
	port_power_delay	- delay after switching ports power on
	controller	- usb controller control:
	enable	- enable usb controller
	disable	- disable usb controller
	ohci	- USB OHCI (USB 1.1) configuration:
	port_power_delay	- delay after switching ports power on
	controller	- usb controller control:
	enable	- enable usb controller
	disable	- disable usb controller
	nvme	- NVME configuration:
	memory_page_size	- memory page size
	controller	- nvme controller control:
	enable	- enable nvme controller
	disable	- disable nvme controller
	test_mem	- Set Test MEM Flag
	fruid	- FRUID configuration:
	data	- setup FRUID
	unlock	- unlock FRUID area
	mb_name	- set motherboard product name
	auto_install	- Set auto install flag
	chipset	- Set/Get Chipset Config Settings:
	pm1r	- CPU IPCC3 Link Params:
	amplitude	- Transmitter Amplitude
	preemp	- Pre-Emphasise Value
	rx_eq	- RX Equalizer Value
	cpu_pci	- Set/Get RT_PCICFGED Mask of Embedded Devs
	cpu_rt	- Set/Get PCI[MEM,IO],IOAPIC,MLO Space Areas
	cpu_cores	- Set/Get cores masks for all CPUs
	prepic_linp	- Set/Get PREPIC_LINP_N register value
	hmu_ctrl0	- Set/Get HMU_CTRL0 register value
	hmu_ctrl1	- Set/Get HMU_CTRL1 register value
	hmu_mic	- Set/Get HMU_MIC register value
	hmu_mcen	- Set/Get HMU_MCEN values (RankEn for all CPU->Ch)

hmu_imask	- Set/Get HMU_IMASK register value
freq_core_float	- Set/Get PMC_FREQ_CORE_FLOAT register value
freq_ocn_float	- Set/Get PMC_FREQ_OCN_FLOAT register value
ocn_hmuil	- Set/Get OCN_HMUIL register value
ocn_l3il	- Set/Get OCN_L3IL register value
ocn_l3en	- Set/Get OCN_L3EN register value

Продолжение таблицы 5

mc_ecc	- Set/Get MC_ECC fields for any CPU->Ch
cpu_mopma	- Set/Get MC_MOPMa fields for need CPU
cpu_mopmb	- Set/Get MC_MOPMb fields for need CPU
cpu_mopmc	- Set/Get MC_MOPMc fields for need CPU
cpu_tim1	- Set/Get MC_TIM1 fields for need CPU
cpu_tim2	- Set/Get MC_TIM2 fields for need CPU
cpu_perf0	- Set/Get MC_PERF0 fields for need CPU
cpu_perf1	- Set/Get MC_PERF1 fields for need CPU
cpu_cfg	- Set/Get MC_CFG fields for need CPU
cpu_ctl	- Set/Get MC_CTL fields for need CPU
cpu_odt	- Set/Get MC_ODT fields for need CPU
cpu_am0	- Set/Get MC_AM0 fields for need CPU
cpu_am1	- Set/Get MC_AM1 fields for need CPU
cpu_ax0	- Set/Get MC_AX0 fields for need CPU
cpu_ax1	- Set/Get MC_AX1 fields for need CPU
cpu_ax2	- Set/Get MC_AX2 fields for need CPU
cpu_errrec	- Set/Get MC_ERRREC fields for need CPU
hc_ctrl	- Set/Get HC_CTRL fields for need CPU
hc_int_mask	- Set/Get HC_INT_MASK fields for need CPU
l3_imsk	- Set/Get L3_IMSK register value
kpi_mask	- Set/Get EIOHs Mask
map	- Get (print) NVRAM map for HW CMOS parameters
i2cb	- I2C 8-bit Access; Param: bus, adr, reg, [data]
i2cw	- I2C 32-bit Access; Param: bus, adr, reg, [data]
i2ch	- I2C 16-bit Access; Param: bus, adr, reg, [data]
i2cl	- I2C 64-bit Access; Param: bus, adr, reg, [data]
get	- Allows to get parameters:
boot_device	- Boot device
version	- Prints info:
boot	- Version of boot program
sp	- Serial ports settings:
output_dis	- Serial port output disable flag
input_dis	- Serial port input disable flag
memory	- Memory:
regions	- Regions
limit	- Limited size
csum	- Control sum settings:
calc	- Calculate and print control sums, when loading files
check	- Check loaded file
kernel_check	- Kernel check loaded file

apmdz	- Turn on/off usage of "Echelon-E" APMDZ
rtc	- real time clock settings:
time	- set/show current time
date	- set/show current date
weekday	- set/show current day of the week
password	- Enter password menu:

Продолжение таблицы 5

status	- Password status
serial_num	- Machine serial number
drives	- Show current drives and partitions
drive_init_delay	- Delay in seconds, before any drive init-starts
mb_version	- Machine version magic (arch_type)
cpu_type	- CPU type
mc	- Memory controller parameters:
freq	- MC Freq Tunes
MemAlertEn	- DDR4PHY MemAlertEn
odti0	- DDR4PHY ODTImpedance[0]
txi0	- DDR4PHY TxImpedance[0]
DisDynAdrTri	- DDR4PHY DisDynAdrTri
ecc	- Show ECC's
dump	- Show MC's
mc_pll	- Tune MC Frequency by PLL settings (clkf,clkr,clkod)
ahci	- AHCI configuration:
controller	- tune AHCI controller
max_speed	- AHCI speed limit
ide	- IDE configuration:
mcst	- MCST IDE configuration:
maxdma	- The highest DMA mode
controller	- tune IDE controller
eth	- Ethernet Controllers configuration:
100	- ETH100 configuration:
mac	- MAC address base
speed	- Data speed
mode	- Normal/Loopback
full	- Init Block Full-Bit
1000	- ETH1000 configuration:
mac	- MAC address base
speed	- Data speed
mode	- Normal/Loopback
full	- Init Block Full-Bit
controller	- tune ETH controller
aoe	- ATA over Ethernet (AoE) settings:
server	- initial requests settings:
mac	- destination MAC address
major	- destination AoE major address
minor	- destination AoE minor address
timeout	- timeout settings:

break	- timeout (ms) to break exchange with AoE server
video	- tune video controllers:
output	- choose video output
dc	- get video mode for display controller
dc_lvds	- get lvds out video mode
mode	- get video mode num

Продолжение таблицы 5

vga	- deprecated parameter, use set/get video
pcie	- tune PCIe bridges:
baud	- configure PCIe baud
eioh	- tune EIOH:
slink	- Conf SLink Freq
usb	- USB configuration:
xhci	- USB XHCI (USB 3) configuration:
port_power_delay	- delay after switching ports power on
controller	- usb controller control
ehci	- USB EHCI (USB 2.0) configuration:
port_power_delay	- delay after switching ports power on
controller	- usb controller control
ohci	- USB OHCI (USB 1.1) configuration:
port_power_delay	- delay after switching ports power on
controller	- usb controller control
nvme	- NVME configuration:
memory_page_size	- memory page size
controller	- nvme controller control
test_mem	- Get Test MEM Flag: 1/0 - ON/OFF
fruid	- FRUID configuration:
data	- print FRUID info
unlock	- print FRUID area lock status
mb_name	- get motherboard product name
auto_install	- Get auto install flag: 1/0 - ON/OFF
chipset	- Set/Get Chipset Config Settings:
pm1r	- CPU IPCC3 Link Params:
amplitude	- Transmitter Amplitude
preemp	- Pre-Emphasise Value
rx_eq	- RX Equalizer Value
cpu_pci	- Set/Get RT_PCICFGED Mask of Embeded Devs
cpu_rt	- Set/Get PCI[MEM,IO],IOAPIC,MLO Space Areas
cpu_cores	- Set/Get cores masks for all CPUs
prepic_linp	- Set/Get PREPIC_LINP_N register value
hmu_ctrl0	- Set/Get HMU_CTRL0 register value
hmu_ctrl1	- Set/Get HMU_CTRL1 register value
hmu_mic	- Set/Get HMU_MIC register value
hmu_mcen	- Set/Get HMU_MCEN values (RankEn for all CPU->Ch)
hmu_imask	- Set/Get HMU_IMASK register value
hmu_sig0	- Get HMU_DIR_BIST_SIG0 register value
hmu_sig1	- Get HMU_DIR_BIST_SIG1 register value

freq_core_float	- Set/Get PMC_FREQ_CORE_FLOAT register value
freq_ocn_float	- Set/Get PMC_FREQ_OCN_FLOAT register value
ocn_hmuil	- Set/Get OCN_HMUIL register value
ocn_l3il	- Set/Get OCN_L3IL register value
ocn_l3en	- Set/Get OCN_L3EN register value
mc_ecc	- Set/Get MC_ECC fields for any CPU->Ch

Окончание таблицы 5

cpu_mopma	- Set/Get MC_MOPMa fields for need CPU
cpu_mopmb	- Set/Get MC_MOPMb fields for need CPU
cpu_mopmc	- Set/Get MC_MOPMc fields for need CPU
cpu_tim1	- Set/Get MC_TIM1 fields for need CPU
cpu_tim2	- Set/Get MC_TIM2 fields for need CPU
cpu_perf0	- Set/Get MC_PERF0 fields for need CPU
cpu_perf1	- Set/Get MC_PERF1 fields for need CPU
cpu_cfg	- Set/Get MC_CFG fields for need CPU
cpu_ctl	- Set/Get MC_CTL fields for need CPU
cpu_odt	- Set/Get MC_ODT fields for need CPU
cpu_am0	- Set/Get MC_AM0 fields for need CPU
cpu_am1	- Set/Get MC_AM1 fields for need CPU
cpu_ax0	- Set/Get MC_AX0 fields for need CPU
cpu_ax1	- Set/Get MC_AX1 fields for need CPU
cpu_ax2	- Set/Get MC_AX2 fields for need CPU
cpu_errrec	- Set/Get MC_ERRREC fields for need CPU
hc_ctrl	- Set/Get HC_CTRL fields for need CPU
hc_int_mask	- Set/Get HC_INT_MASK fields for need CPU
l3_imsk	- Set/Get L3_IMSK register value
kpi_mask	- Set/Get EIOHs Mask
map	- Get (print) NVRAM map for HW CMOS parameters
i2cb	- I2C 8-bit Access; Param: bus, adr, reg, [data]
i2cw	- I2C 32-bit Access; Param: bus, adr, reg, [data]
i2ch	- I2C 16-bit Access; Param: bus, adr, reg, [data]
i2cl	- I2C 64-bit Access; Param: bus, adr, reg, [data]
reset	- Init Hard reset signal
tests	- Enter tests menu

7.1. Подробное описание команд программного модуля специальных настроек

7.1.1. Команда вывода справочной информации по командам настроек

```
Help
```

Вывод полного списка инженерных команд ПНС.

```
#help
```

7.1.2. Команда перехода в меню загрузки ОС

```
Boot [ Esc ]
```

иди нажать клавишу

Войти в загрузочное меню

```
#boot
```

7.1.3. Команды определения устройства загрузки

```
set boot_device
```

Установить индекс устройства загрузки

```
# set boot_device
Usage:
    set boot_device <drive>
        <drive> - Drive number (Range: 0 - 255)
# set boot_device 3
#
```

```
get boot_device
```

Получить индекс устройства загрузки

```
# get boot_device
3
#
```

7.1.4. Команды настройки ввода/вывода сообщений ПНС через последовательный порт

```
set sp output_dis
```

Включить/Выключить вывод на последовательный порт

```
# set sp output_dis
Usage:
    set sp output_dis <port> <value>
        <port> - Serial port number (0/1)
        <value> - enable/disable (0/1)
# set sp output_dis 0 0
#
```

```
get sp output_dis
```

Вывести значение флага Включить/Выключить вывод на последовательный порт

```
# get sp output_dis
Usage:
    get sp output_dis <port>
        <port> - Serial port number (0/1)
# get sp output_dis 0
0
#
```

```
set sp input_dis
```

Включить/Выключить ввод с последовательного порта

```
# set sp input_dis
Usage:
    set sp input_dis <port> <value>
        <port> - Serial port number (0/1)
        <value> - enable/disable (0/1)
# set sp input_dis 1 1
#
```

```
get sp input_dis
```

Вывести значение флага Включить/Выключить ввод с последовательного порта

```
# get sp input_dis
Usage:
    get sp input_dis <port>
    <port> - Serial port number (0/1)
# get sp input_dis 0
#
```

7.1.5. Команды конфигурирования и вывода настроек ОЗУ

```
set memory limit
```

Установить ограничение размера ОЗУ

```
# set memory limit 1024
USAGE:
set memory limit <size>
    <size> - the limited memory size on each node (in MiB) you want
to set.
    1. <size> equal to 0 means all of memory.
    2. Else <size> must be more or equal to 4096 and less
or equal to 4194304.
EXAMPLE:
set memory limit 1024
NOTE:
You must restart the machine after setting the memory limit.
#
```

```
get memory limit
```

Вывести заданное ограничение размера ОЗУ

```
# get memory limit
All of memor
#
```

```
get memory regions
```

Вывод значения размеров ОЗУ по регионам

```
# get memory regions
Usage:
    get memory (<node>)
    <node> - Optional: Node number (Range: 0 - 3)
# get memory regions 0
Node#00: Memory map:
Region [0]: 0 MB - 1024 MB (1024 MB)
Region [1]: 33792 MB - 65536 MB (31744 MB)
```

#

7.1.6. Команды настройки доверенной загрузки

```
set csum calc
```

Включить/Выключить вычисление и выдачу контрольных сумм системных файлов при загрузке ОС

```
# set csum calc
USAGE:
set csum calc <0|1>
1 - Control sum of every loaded file will be calculated and printed
0 - Control sum calculation disabled
# set csum calc 0
#
```

```
get csum calc
```

Вывести значение флага Вычислать/Не вычислять контрольные суммы системных файлов ОС.

```
# get csum calc
USAGE:
get csum calc
# get csum calc
0
#
```

```
set csum check
```

Включить/Выключить проверку контрольных сумм при загрузке системных файл ОС

```
# set csum check
USAGE:
set csum check <0|1>
1 - Control sum of main target (linux kernel, lintel etc.) will be
calculated and checked to previously saved value in NVRAM
0 - Control sum check disabled
# set csum check 0
#
```

```
get csum check
```

Вывод значение флага Включить/Выключить проверку контрольных сумм при загрузке системных файл ОС

```
# get csum check
```



```

USAGE:
get csum check
# get csum check
0
#

```

```
set csum save
```

Сохранить контрольную сумму следующего загруженного файла в NVRAM

```

# set csum save
USAGE:
set csum save <0|1>
1 - Control sum of next loaded file will be calculated and saved in
NVRAM
0 - Nothing mentioned will be done
# set csum save 0
#

```

```
set csum kernel_check
```

Включить/Выключить проверку контрольных сумм системных файлов ОС.

```

# set csum kernel_check
USAGE:
set csum kernel_check <0|1>
1 - Control sum of main target (linux kernel, lintel etc.) will be
calculated and checked to previously saved value in NVRAM
0 - Control sum check disabled
# set csum kernel_check 0
#

```

```
get csum kernel_check
```

Вывести значение флага Проверить/Не проверять контрольные суммы загружаемых файлов ядра ОС.

```

# get csum kernel_check
USAGE:
get csum kernel_check
# get csum kernel_check
0
#

```

```
set apmdz
```

Включить/выключить использование АПМДЗ "Эшелон-Э"

```

# set apmdz
USAGE:
set apmdz <0|1>
1 - Turn on usage of "Echelon-E" APMDZ

```

ТВГИ.00406-01 34 01

```
0 - Turn off usage of "Echelon-E" APMDZ
# set apmdz 0
#
```

```
get apmdz
```

**Вывод значение флага Включить/выключить использование АПМДЗ
"Эшелон-Э"**

```
# get apmdz
USAGE:
get apmdz
# get apmdz
0
#
```

```
set password enable
```

Включить доступ в систему по паролю

```
# set password enable
Warning: Password is not set
Enter new password      : *****
Confirm new password    : *****
Password succesfully enabled
#
```

```
set password disable
```

Выключить доступ в систему по паролю

```
# set password disable
Enter current password: *****
Password succesfully disabled
#
```

```
set password change
```

Поменять пароль доступа в систему

```
# set password change
Enter new password      : *****
Confirm new password    : *****
Password succesfully changed
#
```

```
get password status
```

Вывод текущего статуса пароля

```
# get password status
Status      : Disabled
Check mode: Modify
```

#

set password mode

Режим проверки пароля**(0 – запрос пароля только при изменении параметров)****(1 - запрос пароля только при изменении параметров и при включении****БК)**

set password mode

Usage:

mode <mode>

<mode> - Password check stage:

0 - Check password on modify stage

1 - Check password on load (and modify) stage

Password mode succesfully changed

set password mode 0

#

7.1.7. Команды установки времени

set rtc time

Установка времени

set rtc time

USAGE:

set rtc time XX:YY:ZZ

XX - Hours

YY - Minutes

XX - Seconds

set rtc time 18:45:00

#

get rtc time

Вывод текущего значения времени.

get rtc time

USAGE:

get rtc time

get rtc time

Current time: 20:09:32

#

set rtc date

Установка даты

```
# set rtc date
USAGE:
set rtc date XX.YY.ZZ
XX - Day
YY - Month
ZZ - Year
# set rtc date 08.02.23
#
```

```
get rtc date
```

Вывод текущего значения даты

```
# get rtc date
USAGE:
get rtc time
# get rtc date
Current date: 18.02.23
#
```

```
set rtc weekday
```

Установка дня недели

```
# set rtc weekday
USAGE:
set rtc weekday X
X - Number of day in a week
# set rtc weekday 3
#
```

```
Get rtc weekday
```

Вывод текущего дня недели

```
# get rtc weekday
USAGE:
get rtc weekday
# get rtc weekday
Current day of the week: 4
#
```

```
set cpu_type
```

Установить тип процессора

```
# set cpu_type
Usage:
    set cpu_type <type>
```

ТВГИ.00406-01 34 01

```

                <type> - CPU type (Range: 0 - 64)
# set cpu_type 11
#

```

```
set mc freq
```

Установка параметров freq

```

# set mc freq
Usage:
    set mc MemAlertEn <node> <ch> <val>
                <node> - Node num (Range: 0 - 3)
                <ch> - MC ch num (Range: 0 - 7)
                <val> - Param val (Range: 0 - 1)
# set mc MemAlertEn 0 0 0x0
#

```

```
get mc freq
```

Выдача установленных параметров freq

```

# get mc freq
Usage:
    get mc_freq <node>
                <node> is CPU number of Range: 0-3
# get mc freq 0
CPU0  MC0 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0  MC0 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0  MC1 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0  MC1 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0  MC2 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0  MC2 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0  MC3 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0  MC3 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0  MC4 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0  MC4 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0  MC5 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0  MC6 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0  MC6 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0  MC7 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0  MC7 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
#

```

```
set mc MemAlertEn
```

Установка параметров DDR4PHY MemAlertEn.

```

# set mc MemAlertEn
Usage:

```

ТВГИ.00406-01 34 01

```

set mc MemAlertEn <node> <ch> <val>
    <node> - Node num (Range: 0 - 3)
    <ch> - MC ch num (Range: 0 - 7)
    <val> - Param val (Range: 0 - 1)
# set mc MemAlertEn 0 0 0x0
#

```

```
get mc MemAlertEn
```

Выдача установленных параметров DDR4PHY MemAlertEn

```

# get mc MemAlertEn
Usage:
    get mc MemAlertEn <node> <ch>
        <node> - Node number (Range: 0 - 3)
        <ch> - MC ch num (Range: 0 - 7)
# get mc MemAlertEn 0 0
0x0
#

```

```
set mc odti0
```

Установка параметров DDR4PHY ODTImpedance[0].

```

# set mc odti0
Usage:
    set mc odti0 <node> <ch> <val>
        <node> - Node num (Range: 0 - 3)
        <ch> - MC ch num (Range: 0 - 7)
        <val> - Param val (Range: 0 - 240)
# set mc odti0 0 0 48
#

```

```
get mc odti0
```

Выдать установленные параметры DDR4PHY ODTImpedance[0]

```

# get mc odti0
Usage:
    get mc odti0 <node> <ch>
        <node> - Node number (Range: 0 - 3)
        <ch> - MC ch num (Range: 0 - 7)
# get mc odti0 0 0
0x30
#

```

```
set mc txi0
```

Установка параметров DDR4PHY TxImpedance[0].

```

# set mc txi0
Usage:
    set mc txi0 <node> <ch> <val>
        <node> - Node num (Range: 0 - 3)

```

ТВГИ.00406-01 34 01

```

                <ch> - MC ch num (Range: 0 - 7)
                <val> - Param val (Range: 0 - 240)
# set mc txi0 0 0 40
#

```

```
get mc txi0
```

Выдать установленные параметры DDR4PHY TxImpedance[0]

```

# get mc txi0
Usage:
    get mc txi0 <node> <ch>
                <node> - Node number (Range: 0 - 3)
                <ch> - MC ch num (Range: 0 - 7)
# get mc txi0 0 0
0x28
#

```

```
set mc DisDynAdrTri
```

Установка параметров DDR4PHY DisDynAdrTri.

```

# set mc DisDynAdrTri
Usage:
    set mc DisDynAdrTri <node> <ch> <val>
                <node> - Node num (Range: 0 - 3)
                <ch> - MC ch num (Range: 0 - 7)
                <val> - Param val (Range: 0 - 1)
# set mc DisDynAdrTri 0 0 1
#

```

```
get mc DisDynAdrTri
```

Выдать установленные параметры DDR4PHY DisDynAdrTri

```

# get mc DisDynAdrTri
Usage:
    get mc DisDynAdrTri <node> <ch>
                <node> - Node number (Range: 0 - 3)
                <ch> - MC ch num (Range: 0 - 7)
# get mc DisDynAdrTri 0 0
0x1
#

```

```
set mc_pll
```

Настройка частоты MC с помощью настроек PLL (clkf,clkr,clkod)

```

# set mc_pll
Usage:
    set mc_pll <node> <clkf> <clkr> <clkod>
                <node> is CPU number of Range: 0-3
                <clkf> <clkr> <clkod> are PLL tunes for MC Freq
# set mc_pll 0 0x34 0 3
CPU0 MC0 Freq: 0.0 MHz calc for fRef: 100.0 MHz, clkf:0x34, clkr:0,
clkod:3

```

ТВГИ.00406-01 34 01

```

CPU0 MC1 Freq: 0.0 MHz calc for fRef: 100.0 MHz, clkf:0x34, clkr:0,
clkod:3
CPU0 MC2 Freq: 0.0 MHz calc for fRef: 100.0 MHz, clkf:0x34, clkr:0,
clkod:3
CPU0 MC3 Freq: 0.0 MHz calc for fRef: 100.0 MHz, clkf:0x34, clkr:0,
clkod:3
CPU0 MC4 Freq: 0.0 MHz calc for fRef: 100.0 MHz, clkf:0x34, clkr:0,
clkod:3
CPU0 MC5 Freq: 0.0 MHz calc for fRef: 100.0 MHz, clkf:0x34, clkr:0,
clkod:3
CPU0 MC6 Freq: 0.0 MHz calc for fRef: 100.0 MHz, clkf:0x34, clkr:0,
clkod:3
CPU0 MC7 Freq: 0.0 MHz calc for fRef: 100.0 MHz, clkf:0x34, clkr:0,
clkod:3
Tunes will take Effect only after Reset
#

```

```
get mc_pll
```

**Выдать текущие настройки частоты MC через параметры PLL
(clkf,clkr,clkod)**

```

# get mc_pll
Usage:
    get mc_freq <node>
    <node> is CPU number of Range: 0-3
# get mc_pll 0
CPU0 MC0 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0 MC0 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0 MC1 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0 MC1 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0 MC2 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0 MC2 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0 MC3 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0 MC3 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0 MC4 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0 MC4 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0 MC5 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0 MC6 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0 MC6 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
CPU0 MC7 Freq, get from PLLMC_CTRL0-3: 650.0 MHz
CPU0 MC7 Freq, calc from NVRAM: 650.0 MHz (FRef: 100.0 MHz,
clkf:0x3400000000 clkr:0 clkod:3)
#

```

```
get ecc
```

7.1.8. Вбывать значение ECC

```

# get mc ecc
Usage:

```


ТВГИ.00406-01 34 01

```

get mc ecc <node>
    <node> - Node number (Range: 0 - 3)
# get mc ecc 0
Rd MC0 MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode =0*[ 0:
0]ee =1
Rd MC1 MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode =0*[ 0:
0]ee =1
Rd MC2 MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode =0*[ 0:
0]ee =1
Rd MC3 MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode =0*[ 0:
0]ee =1
Rd MC4 MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode =0*[ 0:
0]ee =1
Rd MC5 MC_ECC =0 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode =0*[ 0:
0]ee =0
Rd MC6 MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode =0*[ 0:
0]ee =1
Rd MC7 MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode =0*[ 0:
0]ee =1
#

```

```
get dump
```

Выдать параметры МС

```

# get mc dump
Usage:
    get mc info (<node>)
    <node> - Optional: Node number (Range: 0 - 3)
# get mc dump 0

```

7.1.8. Команды настройки IDE-интерфейсов и SATA-интерфейсов

```
get drivers
```

Выдать перечень подключённых носителей и их портиций

```

# get drives
Usage:
    get drives
# get drives
CPU0.0: Drive [0]: SATA - NODE[0] IOLINK[0] PCI BUS[1]:DEV[3]:FUNC[0],
MCST SATA COMBINED Port [0] - SATA SSD
CPU0.0:      Partition [0]: fs: Linux EXT2;
           U:7aae224f-ed1a-41ec-9a00-b1d083d6a565 L:""
CPU0.0:      Partition [1]: Linux
#

```

```
set ahci controller enable
```

Включить контроллер или порт AHCI.

```
# set ahci controller
```

ТВГИ.00406-01 34 01

```

enable - enable controller or port
disable - disable controller or port
# set ahci controller enable
SUCCESS: cmos updated
#

```

```
set ahci controller disable
```

Выключить контроллер или порт АНСИ.

```

# set ahci controller
enable - enable controller or port
disable - disable controller or port
# set ahci controller disable
SUCCESS: cmos updated
#

```

```
get ahci controller
```

Выдать состояние контроллера АНСИ.

```

# get ahci controller
Usage:
    get ahci controller [<node> <link> <bus> <slot> <func>]
# get ahci controller
AHCI Controller on (core=0,link=0,bus=1,slot=3,func=0) enabled
#

```

```
set ahci controller max_speed
```

Установить ограничение скорости АНСИ

```

# set ahci max_speed
Usage:
    set ahci max_speed <speed>
    <speed> - speed limit to negotiate:
                0 - no restrictions
                1 - Gen.1
                2 - Gen.2
# set ahci max_speed 0
#

```

```
get achi max_speed
```

Выдать установленное ограничение скорости АНСИ

```

# get ahci max_speed
Usage:
    get ahci max_speed
Output: 0 - no restrictions
        1 - Gen.1
        2 - Gen.2
# get ahci max_speed
0 (no restrictions)
#

```

```
set ide mcst maxdma
```

Установка скорости обмена IDE по DMA.

```
# set ide mcst maxdma
Usage: set ide mcst maxdma pio|mwdmaN|udmaN
      N - mode number
EXAMPLE: set ide mcst maxdma udma2
# set ide mcst maxdma udma2
#
```

```
get ide mcst maxdma
```

Выдать установленную скорость обмена IDE по DMA.

```
# get ide mcst maxdma
Usage: get ide mcst maxdma
# get ide mcst maxdma
  Ultra DMA 2
#
```

```
set ide controller enable
```

Включить IDE контроллер или диск

```
# set ide controller
enable           - enable controller or drive
disable         - disable controller or drive
# set ide controller enable
SUCCESS: cmos updated
#
```

```
set ide controller disable
```

Выключить IDE контроллер или диск

```
# set ide controller
enable           - enable controller or drive
disable         - disable controller or drive
# set ide controller disable
SUCCESS: cmos updated
#
```

```
get ide controller
```

Выдать установленное состояние контроллера IDE

```
# get ide controller
Usage:
  get ide controller [<node> <link> <bus> <slot> <func>]
# get ide controller
  disable
```

#

7.1.9. Команды настройки онбордного Ethernet-канала

```
set eth 100 mac
```

Установка базового MAC-адреса для встроенных Ethernet-интерфейсов 100Mb

```
# set eth 100 mac
Usage:
    set eth <100|1000> mac 01-02-03-04-05-06
    or
    set eth <100|1000> mac 01:02:03:04:05:06
# set eth 100 mac AB:F0:C7:13:55:77
#
```

```
get eth 100 mac
```

Выдать установленный базовый MAC-адрес для встроенных Ethernet-интерфейсов 100Mb

```
# get eth 100 mac
Usage:
    get eth <100|1000> mac
# get eth 100 mac
    12:34:56:AB:CD:EF
#
```

```
set eth 100 speed
```

Установить скорость обмена для базового Ethernet-интерфейса 100Mb

```
# set eth 100 speed
    1000half | 1000full | 100half | 100full | 10half |
10full
# set eth 100 speed 100half
#
```

```
get eth 100 speed
```

Выдать установленную скорость обмена для базового Ethernet-интерфейса 100Mb

```
# get eth 100 speed
Usage:
  get eth <100|1000> speed
# get eth 100 speed
  AUTO
#
```

```
set eth 100 mode
```

Установить режим — Нормальный/Петля для базового Ethernet-интерфейса 100Mb

```
# set eth 100 mode
normal | loop_ext | loop_int
# set eth 100 mode normal
#
```

```
get eth 100 mode
```

Выдать установленный режим — Нормальный/Петля для базового Ethernet-интерфейса 100Mb

```
# get eth 100 mode
Usage:
  get eth <100|1000> mode
# get eth 100 mode
  Normal
#
```

```
set eth 100 full
```

Установить Full-Bit для базового Ethernet-интерфейса 100Mb

```
# set eth 100 full
0 | 1
# set eth 100 full 1
#
```

```
get eth 100 full
```

Выдать установленный Full-Bit для базового Ethernet-интерфейса 100Mb

```
# get eth 100 full
Usage:
  get eth <100|1000> full
# get eth 100 full
  0
#
```

```
set eth 1000 mac
```

Установка базового MAC-адреса для встроенных Ethernet-интерфейсов 1000Mb

```
# set eth 1000 mac
Usage:
  set eth <100|1000> mac 01-02-03-04-05-06
  or
  set eth <100|1000> mac 01:02:03:04:05:06
# set eth 1000 mac AB:F0:C7:13:55:77
#
```

```
get eth 1000 mac
```

Выдать установленный базовый MAC-адреса для встроенных Ethernet-интерфейсов 1000Mb

```
# get eth 1000 mac
Usage:
  get eth <100|1000> mac
# get eth 1000 mac
12:34:56:AB:CD:EF
#
```

```
set eth 1000 speed
```

Установить скорость обмена для базового Ethernet-интерфейса 1000Mb

```
# set eth 1000 speed
1000half | 1000full | 100half | 100full | 10half | 10full
# set eth 1000 speed 1000half
#
```

```
get eth 1000 speed
```

Выдать установленную скорость обмена для базового Ethernet-интерфейса 1000Mb

```
# get eth 1000 speed
Usage:
  get eth <100|1000> speed
# get eth 1000 speed
AUTO
#
```

```
set eth 1000 mode
```

Установить режим — Нормальный/Петля для базового Ethernet-интерфейса 1000Mb

```
# set eth 1000 mode
```

ТВГИ.00406-01 34 01

```
normal | loop_ext | loop_int
# set eth 1000 mode normal
#
```

```
get eth 1000 mode
```

Выдать установленный режим — Нормальный/Петля для базового Ethernet-интерфейса 1000Mb

```
# get eth 1000 mode
Usage:
  get eth <100|1000> mode
# get eth 1000 mode
Normal
#
```

```
set eth 1000 full
```

Установить Full-Bit для базового Ethernet-интерфейса 1000Mb

```
# set eth 1000 full
0 | 1
# set eth 1000 full 1
#
```

```
get eth 1000 full
```

Выдать установленный Full-Bit для базового Ethernet-интерфейса 1000Mb

```
# get eth 1000 full
Usage:
  get eth <100|1000> full
# get eth 1000 full
0
#
```

```
set eth controller enable
```

Включить онбордный Ethernet контроллер

```
# set eth controller
enable - enable ETH controller
disable - disable ETH controller
# set eth controller enable
SUCCESS: cmos updated
#
```

```
set eth controller disable
```

Отключить онбордный Ethernet контроллер

```
# set eth controller
  enable - enable ETH controller
  disable - disable ETH controller
# set eth controller disable
SUCCESS: cmos updated
#
```

```
get eth ontroller
```

**Выдать состояние базового Ethernet-контроллера.
За словом идёт следующее ключевое слово.**

```
# get eth controller
Usage:
  get eth controller
  get eth controller <node> <link> <bus> <slot> <func>
# get eth controller
# get eth controller 0 0 0 0 0
FAILED: wrong pci address or unknown controller

#
```

7.1.10. Команды настройки загрузки ОС по сети

```
set aoe server mac
```

MAC-адрес AoE сервера

```
# set aoe server mac
USAGE:
  set aoe server mac <address>
  <address> -
  ethernet address the initial searching requests
  will be transmitted to. The form of <address> is
  XX:XX:XX:XX:XX:XX or XX-XX-XX-XX-XX-XX
  where X is a hexadecimal digit.
  The leading zero within a one octet can be omitted.
  The default is FF:FF:FF:FF:FF:FF (broadcast requests).
EXAMPLE:
  set aoe server mac 12:34:56:78:9A:BC

# set aoe server mac FF:EE:CC:77:06:66
#
```

```
get aoe server mac
```

Выдать установленный MAC-адрес AoE-сервера

*В ПНС E16C команды с ключевым словом aoe
не реализованы.*


```
# get aoe server mac
  USAGE:
    get aoe server mac
```

```
# get aoe server mac
  FF:FF:FF:FF:FF:FF
#
```

set aoe server major

Задать основной адрес AoE назначения

```
# set aoe server major
  USAGE:
    set aoe server major <address>
    <address> -
      AoE major address used in the initial searching requests.
      <address> is a number between 0 and 65535 inclusive.
      The default is 65535 (0xFFFF), which means broadcast.
```

```
EXAMPLE:
  set aoe server major 12345
```

```
# set aoe server major 00666
#
```

get aoe server major

Выдать установленный основной адрес AoE-сервера назначения

```
# get aoe server major
  USAGE:
    get aoe server major
```

```
# get aoe server major
  65535 (0xFFFF)
#
```

set aoe server minor

Задать второстепенный адрес AoE назначения

```
# set aoe server minor
  USAGE:
    set aoe server minor <address>
    <address> -
      AoE minor address used in the initial searching requests.
      <address> is a number between 0 and 255 inclusive.
      The default is 255 (0xFF), which means broadcast.
```

```
EXAMPLE:
  set aoe server minor 123
```

```
# set aoe server minor 13
#
```

```
get aoe server minor
```

Выдать установленный адрес AoE-сервер назначения

```
# get aoe server minor
  USAGE:
    get aoe server minor

# get aoe server minor
  255 (0xFF)
#
```

```
set aoe timeout break
```

Задать таймаут (мс) для прерывания обмена с AoE сервером

```
# set aoe timeout break 0777
#
```

```
get aoe timeout reak
```

Выдача таймаута (мс) прекращения поиска AoE-сервера

```
# get aoe timeout break
  USAGE:
    get aoe timeout break

# get aoe timeout break
  5000 (0x1388)
```

7.1.11. Команды настройки вывода на монитор

```
set video output
```

Выбор видеовыхода

```
set video output <number>
```

EXAMPLE:

```
set video output 0xfa
```

VGA List:

```
-----
[0x00] - 00:00:02:00.0, [0x126f/0x0768]
[0xf0] - none
[0xfa] - auto(last found)
-----
```

ТВГИ.00406-01 34 01

```
# set video output 0xfa
Auto(last found) video output is used.
```

```
#
```

```
get video output
```

Выдать номер выбранного видеовыхода

```
# get video output
USAGE:
    get video output
```

```
# get video output
Auto(last found) video output is used.
#
```

```
set video dc
```

настроить контроллер дисплея

```
# set video dc
Usage:
    set video dc <num> <en> <vmode>
    <num> - Display controller number (0/1)
    <en> - disable(0)/enable(0x1..0xF - mask out num)
    0x2 - HDMI0, 0x4 - HDMI1, 0x6 all HDMIx, 0x8 - only lvds zoom
mode)
    <vmode> - drm video mode 1..88)
```

```
# set video dc 0 1 1
#
```

```
get video dc
```

Выдать установленные настройки контроллера дисплея

```
# get video dc
Usage:
    get video dc <num>
    <num> - Display controller number (0/1)
```

```
# get video dc 0
DC:=0:disabled:
#
```

```
set video dc_mode
```

Настройка контроллера дисплея

```
# set video dc_mode
Usage:
    set video dc_mode <num> <en> <hdisplay> <vdisplay> <refresh>
    <num> - Display controller number (0/1)
```

ТВГИ.00406-01 34 01

```

<en> - disable(0)/enable(0x1..0xF - mask out num)
0x2 - HDMI0, 0x4 - HDMI1, 0x6 all HDMIx, 0x8 - only lvds zoom mode)
  <hdisplay> - Display hdisplay
  <vdisplay> - Display vdisplay
  <refresh> - Display vrefresh
# set video dc_mode 0 1 1024 768 60
Set video mode num:=16:1024x768:60
#

```

```
set video dc_lvds
```

Настроить LVDS канал контроллера дисплея

```

dc_lvds_mode - tune display controller for lvds out
# set video dc_lvds
Usage:
    set video dc_lvds <num> <en> <panel>
    <num> - Display controller number (0/1)
    <en> - disable(0)/enable(1 - not zoom,2 - zoom mode)/
    <panel> - panel num max=9

```

```

out    dc_lvds - tune display controller for lvds
out    dc_lvds_mode - tune display controller for lvds
# set video dc_lvds 0 2 0
LCD:=0,LCD Default:VMODE[1024,768]@60,1024x768
#

```

```
get video dc_lvds
```

Выдать настройки контроллера дисплея для выхода lvds

```

# get video dc_lvds
Usage:
    get video dc_lvds <num>
    <num> - Display controller number (0/1)
# get video dc_lvds 0
DC:=0:out=disabled:zoom=dis
#

```

```
set video dc_lvds_mode
```

Настроить разрешение LVDS канала контроллера дисплея

```

# set video dc_lvds_mode
Usage:
    set video dc_lvds_mode <num> <en> <hdisplay> <vdisplay>
<refresh>
    <num> - Display controller number (0/1)
    <en> - disable(0)/enable(1 - not zoom,2 - zoom mode)
    <hdisplay> - Display hdisplay
    <vdisplay> - Display vdisplay
    <refresh> - Display vrefresh
# set video dc_lvds_mode 0 1 1024 768 60

```

```
Not set panel mode
#
```

```
get video mode
```

Выдать настройки контроллера дисплея

```
# get video mode
Usage:
    get video mode <hdisplay> <vdisplay> <refresh>
        <hdisplay> - Display hdisplay
        <vdisplay> - Display vdisplay
        <refresh> - Display vrefresh

# get video mode 1024 768 60
Find video mode num:=16:1024x768:
#
```

7.1.12. Команды настройки КПИ

```
set pcie baud
```

настроить скорость передачи данных PCIe

```
# set pcie baud
You can use only 1, 2, 4, 8 or 16 values!

USAGE:
    set pcie baud <value>
You can use only 1, 2, 4, 8 or 16 values!
EXAMPLE:
    set pcie baud 16
# set pcie baud 16
PCIe baud x16 is selected!

#
```

```
get pcie baud
```

Выдать установленную скорость передачи данных PCIe

```
# get pcie baud
USAGE:
    get pcie baud

# get pcie baud
PCIe baud x16 is selected!
#
```

```
set eioh slink
```

Конфигурирование частоты slink

```
# set eioh slink
Parameter in cmos is incorrect!

USAGE:
    set eioh slink <freq>
    <freq> -
        The frequency of SLink in MHz.
EXAMPLE:
    set eioh slink 133
# set eioh slink 133
Parameter in cmos is incorrect!

USAGE:
    set eioh slink <freq>
    <freq> -
        The frequency of SLink in MHz.
EXAMPLE:
    set eioh slink 133

#
```

```
get eioh link
```

Выдать установленную конфигурацию частоты slink

```
# get eioh slink
USAGE:
    get eioh slink

# get eioh slink
Parameter in cmos is incorrect
```

7.1.13. Команды настройки USB-контроллеров

```
set usb xhci port_power_delay
```

Установить задержку обращения к портам USB-3 после включения питания

```
# set usb xhci port_power_delay
Usage:
    set usb <hci> port_power_delay <delay>
    <hci> - xhci | ehci | ohci
    <delay> - delay after switching ports power on (in ms)
              range: 100-10000
# set usb xhci port_power_delay 1000
#
```

```
get usb xhci port_power_delay
```

Выдать установленную задержку после включения

питания портов USB XHCI (USB 3)

```
# get usb xhci port_power_delay
Usage:
  get usb <hci> port_power_delay
      <hci>    - xhci | ehci | ohci
# get usb xhci port_power_delay
1000 ms
#
```

```
set usb xhci controller enable
```

Включить USB контроллер XHCI (USB 3):

```
# set usb xhci controller
enable          - enable usb controller
disable        - disable usb controller
# set usb xhci controller enable
#
```

```
set usb xhci controller disable
```

Выключить USB контроллер XHCI (USB 3):

```
# set usb ehci controller
enable          - enable usb controller
disable        - disable usb controller
# set usb ehci controller disable
#
```

```
get usb xhci ontroller
```

Ключевое слово к выдаче состоянию USB контроллера и XHCI (USB 3).

```
# get usb xhci controller
Usage:
  get usb <hci> controller
      <hci>    - xhci | ehci | ohci
# get usb xhci controller
enabled
#
```

```
set usb ehci port_power_delay
```

Установить задержку обращения к портам USB-2.0 после включения питания

```
# set usb ehci port_power_delay
Usage:
  set usb <hci> port_power_delay <delay>
      <hci>    - xhci | ehci | ohci
      <delay> - delay after switching ports power on (in ms)
```

ТВГИ.00406-01 34 01

```

                                range: 100-10000
# set usb ehci port_power_delay 2000
#

```

```
get usb ehci ort_power_delay
```

Выдать установленную задержку после включения питания портов USB XHCI (USB 3)

```

# get usb ehci port_power_delay
Usage:
  get usb <hci> port_power_delay
  <hci>    - xhci | ehci | ohci
# get usb ehci port_power_delay
2000 ms
#

```

```
set usb ehci controller enable
```

Включить контроллер USB-2.0

```

# set usb ehci controller
  enable          - enable usb controller
  disable         - disable usb controller
# set usb ehci controller enable
#

```

```
set usb ehci controller disable
```

Выключить USB контроллер EHCI (USB 2.0):

```

# set usb ehci controller
  enable          - enable usb controller
  disable         - disable usb controller
# set usb ehci controller disable
#

```

```
get usb ehci conctroller
```

Выдача текущего состояния контроллера USB EHCI (USB 2.0).

```

# get usb ehci controller
Usage:
  get usb <hci> controller
  <hci>    - xhci | ehci | ohci
# get usb ehci controller
enabled
#

```

```
set usb ohci port_power_delay
```

Установить задержку обращения к портам USB-1.1 после включения питания

ТВГИ.00406-01 34 01

```
# set usb ohci port_power_delay
Usage:
  set usb <hci> port_power_delay <delay>
  <hci>    - xhci | ehci | ohci
  <delay>  - delay after switching ports power on (in ms)
             range: 100-10000
# set usb ohci port_power_delay 3000
#
```

```
get usb ohci ort_power_delay
```

Выдать установленную задержку после включения питания портов USB OHCI (USB 1.1)

```
# get usb ohci port_power_delay
Usage:
  get usb <hci> port_power_delay
  <hci>    - xhci | ehci | ohci
# get usb ohci port_power_delay
100 ms
#
```

```
set usb ohci controller enable
```

Включить контроллер USB 1.1

```
# set usb ohci controller
enable           - enable usb controller
disable         - disable usb controller
# set usb ohci controller enable
#
```

```
set usb ohci controller disable
```

Выключить контроллер USB 1.1

```
# set usb ohci controller
enable           - enable usb controller
disable         - disable usb controller
# set usb ohci controller disable
```

```
get usb ohci controller
```

Выдать состояние USB контроллера EHCI (USB 2.0):

```
# get usb ohci controller
Usage:
  get usb <hci> controller
  <hci>    - xhci | ehci | ohci
# get usb ohci controller
enabled
#
```

7.1.14. Команды конфигурирования NVME

```
set nvme memory_page_size
```

Установить размер страницы памяти NVME

```
# set nvme memory_page_size
Usage:
    set nvme memory_page_size <mps>
    <mps> - 0 for 4K, 1 for 8K, ..., 15 for 128M
# set nvme memory_page_size 15
#
```

```
get nvme memory_page_size
```

Выдать размер страницы памяти NVME

```
# get nvme memory_page_size
0
# get nvme memory_page_size
Usage:
    get nvme memory_page_size
    0 means 4K, 1 means 8K, ..., 15 means 128M
# get nvme memory_page_size
0
#
```

```
set nvme conctroller enable
```

Включить онбордный контроллер NVME

```
# set nvme controller
    enable - enable nvme controller
    disable - disable nvme controller
# set nvme controller enable
#
```

```
set nvme conctroller disable
```

Выключить онбордный контроллер NVME

```
# set nvme controller
    enable - enable nvme controller
    disable - disable nvme controller
# set nvme controller disable
#
```

```
get nvme conctroller
```

Выдать состояние онбордного контроллера NVME

```
# get nvme controller
Usage:
    get nvme controller
# get nvme controller
enabled
#
```

7.1.15. Команды запуска тестов из ПНС

```
set test_mem
```

Включить/Выключить тестирование памяти при запуске ВК

```
# set test_mem
Usage:
    set test_mem 0/1
# set test_mem 0
#
```

```
get test_mem
```

Выдать значение флага Включить/Выключить тестирование памяти при запуске ВК

```
# get test_mem
Usage:
    get test_mem
# get test_mem
0
#
```

7.1.16. Команды идентификации материнской платы и установи FRUID

```
set serial_num
```

Установить серийный номер материнской платы

ТВГИ.00406-01 34 01

```
# set serial_num 123456789
Usage: set serial_num <works_num>
        works_num - 8 decimal digits
EXAMPLE: set serial_num 12345678
```

```
# set serial_num 12345678
# get serial_num
345
#
```

```
get serial_num
```

Вывести серийный номер материнской платы

```
# set serial_num 123456789
Usage: set serial_num <works_num>
        works_num - 8 decimal digits
EXAMPLE: set serial_num 12345678
```

```
# set serial_num 12345678
# get serial_num
345
#
```

```
set mb_version
```

Установить версию материнской платы

```
# set mb_version
Usage:
        set mb_version <version>
        <version> - Motherboard version (Range: 0 - 255)
# set mb_version 77
```

```
#
```

```
get mb_version
```

Вывести установленную версию материнской платы

```
# get mb_version
Usage:
        get mb_version
# get mb_version
77
```

```
#
```

```
set mb_name
```

Задать название материнской платы

```
# set mb_name
Usage:
        set mb_name <name>
        <name> - Motherboard product name
# set mb_name ELBRUS_16C
```

#

get mb_name

Выдать установленное название материнской платы

```
# get mb_name
Usage:
    get mb_name
# get mb_name
ELBRUS_16C
#
```

get version boot

Позволяет получить версию загрузочной программы

```
# get version boot
BOOT          E16C
VERSION:      testing ::::::: ( /branches/l_e16c at revision 9455M )
BUILT BY      egorov_s
TARGET:
    ON         Feb 17 2023
    AT         16:28:10
COMPILER:     lcc:1.25.26:Nov-22-2022:e2k-v2-linux.cross:x86_64-linux
              gcc (GCC)
```

set fruid unlock

Разрешить запись FRUID

```
# set fruid unlock
Usage:
    set fruid unlock
# set fruid unlock
FRUID area unlocked.
#
```

get fruid unlock

Разрешить запись FRUID

```
# get fruid unlock#
Usage:
    get fruid unlock
# get fruid unlock
FRUID area currently unlocked
#
```

set fruid data

Передача данных FRUID по протоколу Xmodem

ТВГИ.00406-01 34 01

```
# set fruud data
Usage:
    set fruud data [port]
    [port] - COM port number for Xmodem receiver (Range: 0 - 1).
    Default value is 0.
# set fruud data 0
Press spacebar to start Xmodem receiver on com port 0.CCCCCCCCCCCCCCCC
```

```
# get fruud data
```

Выдать данне FRUID

```
# get fruud data
Reading FRUID from I2C EEPROM:
Reading FRUID:
Header version: 01

Internal Use Area:
Version: 01
mac_addr: 98A7B0071B20
mb_ver: 00

Board Info Area:
Version: 01
Mfg date: 24:00 Mon 11 Dec 2022
Board manufacturer: -----
Board product name: 1E2C3-TmITX
Board serial number: 0006-12.22 PAN
Board part number: TVGI.469555.480
FRU file ID: 85711881
Product version: 00
#
```

7.1.17. Команды настроек чипсета

```
set cpu_type
```

Задать тип процессора

```
# set cpu_type
Usage:
    set cpu_type <type>
    <type> - CPU type (Range: 0 - 64)
# set cpu_type 33
#
```

```
get cpu_type
```

Выдать установленный тип процессора

```
# get cpu_type
Usage:
    get cpu_type
# get cpu_type
33
#
```

```
set chipset p1mr amplitude
```

Установить амплитуду передатчика

```
# set chipset p1mr amplitude
Usage:
    set chipset p1mr amplitude <node> <val>
    <node> - Node num (Range: 0 - 3)
    <val> - Param val (Range: 0 - 7)
# set chipset p1mr amplitude 0 0
#
```

```
get cipset mp1r mplitude
```

Выдать амплитуда передатчика

```
# get chipset p1mr amplitude
Usage:
    get chipset p1mr amplitude <node>
    <node> - Node number (Range: 0 - 3)
# get chipset p1mr amplitude 0
0x0
#
```

```
set chipset p1mr preemp
```

Установить значение предварительного акцента

```
# set chipset p1mr preemp
Usage:
    set chipset p1mr preemp <node> <val>
    <node> - Node num (Range: 0 - 3)
    <val> - Param val (Range: 0 - 15)
# set chipset p1mr preemp 0 0
#
```

```
get chipset pmr1 preemp
```

Выдать значение предварительного акцента

```
# get chipset p1mr preemp
```

ТВГИ.00406-01 34 01

```
Usage:
  get chipset pmlr preemp <node>
    <node> - Node number (Range: 0 - 3)
# get chipset pmlr preemp 0
0x0
#
```

```
set chipset pmlr rx_eq
```

Установить значение эквалайзера RX

```
# set chipset pmlr rx_eq
Usage:
  set chipset pmlr rx_eq <node> <val>
    <node> - Node num (Range: 0 - 3)
    <val> - Param val (Range: 0 - 7)
# set chipset pmlr rx_eq 0 0
#
```

```
get chipset pmlr rx_eq
```

Выдать значение эквалайзера RX

```
# get chipset pmlr rx_eq
Usage:
  get chipset pmlr rx_eq <node>
    <node> - Node number (Range: 0 - 3)
# get chipset pmlr rx_eq 0
0x1
#
```

```
set cpu_pci
```

Установить маску RT_PCICFGED для встроенных устройств

```
# set cpu_pci
Usage:
```

```
  set cpu_pci <node> <mask>

    <node> is CPU number of Range: 0-3

    <mask> is hex-digit of Range: 0-0xffffffff
```

```
Example: set cpu_pci 0 0xf8000000 /// Disable MGA 2.5
```

```
Example: set cpu_pci 0 0xfc000000 /// Enable all devs
```

```
Alas! Disable MGA 2.5 is needed to connect discrete VGA
```

```
# set cpu_pci 0 0xfc000000
```

```
  NVRAM RT_PCICFGED.value = 0xFC000000 .MGA: 1 .GPU: 1 .ENI: 1 .DCI:
```

```
.ENG: 1 .DCG: 1
```

```
Tunes will take Effect only after Reset
```

```
#
```

```
get cpu_pci
```


Выдать установленную маску RT_PCICFGED для встроенных устройств

```
# get cpu_pci
Usage:
    get cpu_pci <node>

    <node> is CPU number of Range: 0-3
# get cpu_pci 0
Final NVRAM RT_PCICFGED.value = 0 .MGA: 0 .GPU: 0 .ENI: 0 .DCI:
0 .ENG: 0 .DCG: 0
#
```

```
set cpu_rt
```

**Установить/получить области пространства
PCI[MEM,IO],IOAPIC,MLO**

```
# set cpu_rt
Usage:
    set cpu_rt <node> <mask>

    <node> is CPU number of Range: 0-3

    <mask> is hex-digit of Range: 0-0xffffffffffffffff

Byte0:mem_bg byte1:mem_end byte2:io_bg byte3:io_end byte4:ioapic_bg
byte5:mlo_bg byte6:mlo_end byte7:0
# set cpu_rt 0 0x55aa55aa
CPU0 MLO[0:0] PCI_IO[[aa:55]
PCI_MEM[[aa:55] IOAPIC.bgn:0
Tunes will take Effect only after Reset
#
```

```
get cpu_rt
```

**Выдать установленные значение области пространства
PCI[MEM,IO],IOAPIC,MLO**

```
# get cpu_rt
Usage:
    get cpu_rt <node>

    <node> is CPU number of Range: 0-3
# get cpu_rt 0
CPU0 MLO[0:3] PCI_IO[[0:3]
PCI_MEM[[10:13] IOAPIC.bgn:0
#
```

```
set cpu_cores
```

Установить/получить маски ядер для всех процессоров

```
# set cpu_cores
USAGE:
    set cpu_cores <masks>
```

ТВГИ.00406-01 34 01

EXAMPLE:

```
set cpu_cores FFFF:B:555:C1
```

```
FORM of <mask> :
```

```
  CPU      :      0 1 2 3
  Core Mask :      X:X:X:X
```

```
NOTE: X is a hexadecimal digit.
```

```
The first zeros in each mask may be omitted.
```

```
Null mask for BSP CPU will be replaced with default.
```

```
# set cpu_cores EEEE:C:444:C1
```

```
The setting will take effect only after Restarting.
```

```
#
```

```
get cpu_cores
```

Выдать установленные маски ядер для всех процессоров

```
# get cpu_cores
```

```
  USAGE:
```

```
    get cpu_cores
```

```
# get cpu_cores
```

```
Masks of cores for each CPU:
```

```
  CPU 0 cores mask = 0xffff
```

```
  CPU 1 cores mask = 0xffff
```

```
  CPU 2 cores mask = 0xffff
```

```
  CPU 3 cores mask = 0xffff
```

```
WARN: CPU Off in CONF
```

```
WARN: CPU Off in CONF
```

```
WARN: CPU Off in CONF
```

```
#
```

```
set prepic_linp
```

Установить значение регистра PREPIC_LINP_N

```
# set prepic_linp
```

```
Usage:
```

```
set prepic_linp <num> <node> <msb> <lsb> <value>
```

```
<num> - reg number of range: 0-5
```

```
<node> - CPU number
```

```
<msb> - most significant bit number of range: 0-31
```

```
<lsb> - low significant bit number of range: 0-31 <=
```

```
msb
```

```
<value> - value in [msb:lsb] : number of range: 0-
```

```
0xFFFFFFFF
```

```
# set prepic_linp 2 1 0 0 0
```

```
CPU0.0: @ core0 of CPU1 is out of CNF
```

```
CMOS prepic_linp_2[cpu1 ] new Value: 0
```

```
*[ 9: 0] .vect 0
```

```
  [12:12] .stat 0
```

```
*[15:13] .dlvm 0
```

```
*[16:16] .mask 0
```

```
*[29:20] .dst 0
```

```
Save Tune ? (y or n) y
```

```
Settings will take Effect only after Reset
```

```
#
```

```
get repci_linp
```

Выдать установленное значение регистра PREPIC_LINP_N

```
# get prepic_linp
Usage:
    get prepic_linp <num> <node>
        <num>  is reg number of range: 0-5
        <node>  is CPU number
# get prepic_linp 0 0
PREPIC_LINP_0 : CPU0  addr 0x110008C00
*[31: 0] = 0x10000
*[ 9: 0] .vect 0
 [12:12] .stat 0
*[15:13] .dlvm 0
*[16:16] .mask 1
*[29:20] .dst  0
#
```

```
set hmu_ctrl0
```

Установить/получить значение регистра HMU_CTRL0

```
# set hmu_ctrl0
Usage:
    set hmu_ctrl0 <cpu> <msb> <lsb> <val>
        <cpu>:  node#    0-3
        <msb>:  hex|dec 0-31
        <lsb>:  hex|dec 0-31
        <val>:  hex|dec 0-0xffffffff
Result in NVRAM: hmu_ctrl0.[msb:lsb] = val
# set hmu_ctrl0 0 0 0 0
 [31: 0] HMU_CTRL0=0x1078B2FE
*[ 3: 0] diren=0xE
*[ 7: 4] dmadiren=0xF
 [ 9: 8] i2ri=2
 [10:10] cl2i=0
 [11:11] ev_null=0
 [12:12] pf_null=1
 [13:13] decc_corr=1
 [14:14] decc_memcorr=0
 [16:15] wnp_rsn=1
 [18:17] wait_wdat=0
 [19:19] rns_cmp=1
 [20:20] sopc_en=1
 [21:21] rd_exc=1
 [22:22] srq_rnc=1
 [23:23] hllc_sn=0
 [24:24] rlb_rsrv_on=0
*[30:25] rlb_rsrv=8
 [31:31] prior_arb=0

Save Tune ? (y or n) y
Tune effect after reset
#
```

```
hmu_ctrl0
```

Выдать установленное значение регистра HMU_CTRL0

```
# get hmu_ctrl0
Usage:
    get hmu_ctrl0 <cpu>
        <cpu>: node# 0-3
# get hmu_ctrl0 0
[31: 0] HMU_CTRL0=0x1078B2FF
*[ 3: 0] diren=0xF
*[ 7: 4] dmadiren=0xF
[ 9: 8] i2ri=2
[10:10] cl2i=0
[11:11] ev_null=0
[12:12] pf_null=1
[13:13] decc_corr=1
[14:14] decc_memcorr=0
[16:15] wnp_rsn=1
[18:17] wait_wdat=0
[19:19] rns_cmp=1
[20:20] sopc_en=1
[21:21] rd_exc=1
[22:22] srq_rnc=1
[23:23] hllc_sn=0
[24:24] rlb_rsrv_on=0
*[30:25] rlb_rsrv=8
[31:31] prior_arb=0
```

#

```
set hmu_ctrl1
```

Установить/получить значение регистра HMU_CTRL1

```
# set hmu_ctrl1
Usage:
    set hmu_ctrl1 <cpu> <msb> <lsb> <val>
        <cpu>: node# 0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff
Result in NVRAM: hmu_ctrl1.[msb:lsb] = val
# set hmu_ctrl1 0 0 0 0
[31: 0] HMU_CTRL1=2
[ 0: 0] dlben=0
[ 1: 1] dwrb_wren=1
[ 2: 2] llock_prot_en=0
[ 3: 3] dlock_prot_en=0
[ 4: 4] no_pass_ind=0
[ 5: 5] no_pass_nb=0
[ 6: 6] no_pass_3nb=0
[ 7: 7] sc_dlrq_oturn_en=0
[ 8: 8] cl_snri=0
[ 9: 9] dwrb_off=0
[10:10] wait_cdat=0
[11:11] rlb_llock_prot_en=0
[13:12] rlb_llock_prot_age=0
[14:14] rlb_llock_prot_mode=0
[15:15] rlb_way_hit_en=0
*[16:16] i_snri=0
[17:17] rlb_prebusy_en=0
```

ТВГИ.00406-01 34 01

```

[18:18] hmu_fault_mask=0
[19:19] mc0_fault_mask=0
[20:20] mc1_fault_mask=0
*[21:21] coh_ns_on=0
[31:22] NoUse=0

```

```

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
hmu_ctrl1
```

Выдать установленное значение регистра HMU_CTRL1

```

# set hmu_ctrl1
Usage:
    set hmu_ctrl1 <cpu> <msb> <lsb> <val>
        <cpu>: node#    0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff

```

Result in NVRAM: hmu_ctrl1.[msb:lsb] = val

```

# set hmu_ctrl1 0 0 0 0
[31: 0] HMU_CTRL1=2
[ 0: 0] dlben=0
[ 1: 1] dwrb_wren=1
[ 2: 2] llock_prot_en=0
[ 3: 3] dlock_prot_en=0
[ 4: 4] no_pass_ind=0
[ 5: 5] no_pass_nb=0
[ 6: 6] no_pass_3nb=0
[ 7: 7] sc_dlrq_oturn_en=0
[ 8: 8] cl_snri=0
[ 9: 9] dwrb_off=0
[10:10] wait_cdat=0
[11:11] rlb_llock_prot_en=0
[13:12] rlb_llock_prot_age=0
[14:14] rlb_llock_prot_mode=0
[15:15] rlb_way_hit_en=0
*[16:16] i_snri=0
[17:17] rlb_prebusy_en=0
[18:18] hmu_fault_mask=0
[19:19] mc0_fault_mask=0
[20:20] mc1_fault_mask=0
*[21:21] coh_ns_on=0
[31:22] NoUse=0

```

```

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
set hmu_mic
```

Установить/получить значение регистра HMU_MIC

```

# set hmu_mic
Usage:
    set hmu_mic <cpu> <msb> <lsb> <val>

```

ТВГИ.00406-01 34 01

```

        <cpu>: node#    0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff
Result in NVRAM: hmu_mic.[msb:lsb] = val
# set hmu_mic 0 0 0 0
  [31: 0] HMU_MIC=0x249248
  *[ 5: 0] mcil_bit0=8
  *[11: 6] mcil_bit1=9
  *[17:12] mcil_bit2=9
  *[23:18] mcil_bit3=9
  [31:24] mcen=0

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get hmu_mic
```

Выдать установленное значение регистра HMU_MIC

```

# get hmu_mic
Usage:
    get hmu_mic <cpu>
    <cpu>: node# 0-3
# get hmu_mic 0
  [31: 0] HMU_MIC=0x249249
  *[ 5: 0] mcil_bit0=9
  *[11: 6] mcil_bit1=9
  *[17:12] mcil_bit2=9
  *[23:18] mcil_bit3=9
  [31:24] mcen=0
#

```

```
set hmu_mcen
```

Установить/получить значения HMU_MCEN
(RankEn для всех CPU→Ch)

```

# set hmu_mcen
Usage:
    set hmu_mcen <node> <field value>
    <node> is CPU number of Range: 0-3
    <field value> is hex or dec number of Range: 0-
0x2020202020202020
    Ex:          CPU MC 0 1 2 3 4 5 6 7
                set hmu_mcen 0 0x0202020202020202
# set hmu_mcen 0 0x0101010101010101
HMU_MCEN .[63:0]=0x101010101010101
OK hmu.mcen[7]=0x1
OK hmu.mcen[6]=0x1
OK hmu.mcen[5]=0x1
OK hmu.mcen[4]=0x1
OK hmu.mcen[3]=0x1

```

ТВГИ.00406-01 34 01

```
OK hmu.mcen[2]=0x1
OK hmu.mcen[1]=0x1
OK hmu.mcen[0]=0x0
```

```
Save Tune ? (y or n) y
Settings will take Effect only after Reset
#
```

```
get mu_mcen
```

Выдать установленное значение HMU_MCEN (RankEn для всех CPU→Ch)

```
# get hmu_mcen
Usage:
    get hmu_mcen <node>
    <node> - is CPU number of Range: 0-3
    Ex: get hmu_mcen 0

# get hmu_mcen 0
HMU_MCEN .[63:0]=0x10101010101010101
OK hmu.mcen[7]=0x1
OK hmu.mcen[6]=0x1
OK hmu.mcen[5]=0x1
OK hmu.mcen[4]=0x1
OK hmu.mcen[3]=0x1
OK hmu.mcen[2]=0x1
OK hmu.mcen[1]=0x1
OK hmu.mcen[0]=0x1
#
```

```
set hmu_imask
```

Установить значение регистра HMU_IMASK

```
# set hmu_imask
Usage:
    set hmu_imask <cpu> <msb> <lsb> <val>
    <cpu>: node#    0-3
    <msb>: hex|dec 0-31
    <lsb>: hex|dec 0-31
    <val>: hex|dec 0-0xffffffff

Result in NVRAM: hmu_imask.[msb:lsb] = val
# set hmu_imask 0 0 0 0
[31: 0] HMU_IMASK=0
*[0 : 0] decc1_mask=0
*[1 : 1] decc2_mask=0
*[2 : 2] irq_mask=0
*[3 : 3] wdt_mask=0
*[4 : 4] ack_mask=0
*[5 : 5] rls_mask=0
*[6 : 6] rdt0_mask=0
*[7 : 7] rdt1_mask=0
*[8 : 8] rpos_mask=0
*[9 : 9] dpos_mask=0
*[10:10] rfifo_of_mask=0
*[11:11] wfifo0_of_mask=0
*[12:12] wfifo1_of_mask=0
*[13:13] dir_rw_mask=0
```

ТВГИ.00406-01 34 01

```

*[14:14] dir_corr_mask=0
*[15:15] rlb_nway_mask=0
*[16:16] n2h0_irq_par_mask=0
*[17:17] n2h1_irq_par_mask=0
*[18:18] n2h0_ack_par_mask=0
*[19:19] n2h1_ack_par_mask=0
*[20:20] n2h0_rls_par_mask=0
*[21:21] n2h1_rls_par_mask=0
*[22:22] n2h0_dat_bdy_par_mask=0
*[23:23] n2h1_dat_bdy_par_mask=0
*[24:24] n2h0_dat_hdr_par_mask=0
*[25:25] n2h1_dat_hdr_par_mask=0
*[26:26] m2h0_dat_par_mask=0
*[27:27] m2h1_dat_par_mask=0
*[28:28] mc0_mask=0
*[29:29] mc1_mask=0
[31: 2] Reserved=0

```

```

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get hmu_imask
```

Выдать установленное регистра HMU_IMASK

```

# get hmu_imask
Usage:
    get hmu_imask <cpu>
        <cpu>: node# 0-3
# get hmu_imask 0
[31: 0] HMU_IMASK=0
*[0 : 0] decc1_mask=0
*[1 : 1] decc2_mask=0
*[2 : 2] irq_mask=0
*[3 : 3] wdt_mask=0
*[4 : 4] ack_mask=0
*[5 : 5] rls_mask=0
*[6 : 6] rdt0_mask=0
*[7 : 7] rdt1_mask=0
*[8 : 8] rpos_mask=0
*[9 : 9] dpos_mask=0
*[10:10] rfifo_of_mask=0
*[11:11] wfifo0_of_mask=0
*[12:12] wfifo1_of_mask=0
*[13:13] dir_rw_mask=0
*[14:14] dir_corr_mask=0
*[15:15] rlb_nway_mask=0
*[16:16] n2h0_irq_par_mask=0
*[17:17] n2h1_irq_par_mask=0
*[18:18] n2h0_ack_par_mask=0
*[19:19] n2h1_ack_par_mask=0
*[20:20] n2h0_rls_par_mask=0
*[21:21] n2h1_rls_par_mask=0
*[22:22] n2h0_dat_bdy_par_mask=0
*[23:23] n2h1_dat_bdy_par_mask=0
*[24:24] n2h0_dat_hdr_par_mask=0
*[25:25] n2h1_dat_hdr_par_mask=0
*[26:26] m2h0_dat_par_mask=0
*[27:27] m2h1_dat_par_mask=0
*[28:28] mc0_mask=0

```



```
*[29:29] mc1_mask=0
[31: 2] Reserved=0
#
```

```
get hmu_sig0
```

Выдать значение регистра HMU_DIR_BIST_SIG0

```
# get hmu_sig0
Usage:
    get hmu_sig0 <node>
        <node> - is CPU: 0-3
# get hmu_sig0 0
HMU_DIR_BIST_SIG04299088768
.reserv4299088800
.du0_err4299088832
.du1_err4299088864
.du2_err4299088896
.du3_err4299088928
.du4_err4299088960
.du5_err4299088992
.du6_err4299089024
.du7_err4299089056
.lru_err4299089088
.du_total_err4299089120
.du_max_err_lsb4299089152
HMU_DIR_BIST_SIG04299088768
.reserv4299088800
.du0_err4299088832
.du1_err4299088864
.du2_err4299088896
.du3_err4299088928
.du4_err4299088960
.du5_err4299088992
.du6_err4299089024
.du7_err4299089056
.lru_err4299089088
.du_total_err4299089120
.du_max_err_lsb4299089152
HMU_DIR_BIST_SIG04299088768
.reserv4299088800
.du0_err4299088832
.du1_err4299088864
.du2_err4299088896
.du3_err4299088928
.du4_err4299088960
.du5_err4299088992
.du6_err4299089024
.du7_err4299089056
.lru_err4299089088
.du_total_err4299089120
.du_max_err_lsb4299089152
HMU_DIR_BIST_SIG04299088768
.reserv4299088800
.du0_err4299088832
.du1_err4299088864
.du2_err4299088896
.du3_err4299088928
.du4_err4299088960
.du5_err4299088992
.du6_err4299089024
```

ТВГИ.00406-01 34 01

```
.du7_err4299089056
.lru_err4299089088
.du_total_err4299089120
.du_max_err_lsb4299089152
#
```

```
get hmu_sig1
```

Выдать значение регистра HMU_DIR_BIST_SIG1

```
# get hmu_sig1
Usage:
    get hmu_sig1 <node>
        <node> - is CPU: 0-3
# get hmu_sig1 0
HMU_DIR_BIST_SIG04299088768
.reserv4299088800
.du0_err4299088832
.du1_err4299088864
.du2_err4299088896
.du3_err4299088928
.du4_err4299088960
.du5_err4299088992
.du6_err4299089024
.du7_err4299089056
.lru_err4299089088
.du_total_err4299089120
.du_max_err_lsb4299089152
HMU_DIR_BIST_SIG04299088768
.reserv4299088800
.du0_err4299088832
.du1_err4299088864
.du2_err4299088896
.du3_err4299088928
.du4_err4299088960
.du5_err4299088992
.du6_err4299089024
.du7_err4299089056
.lru_err4299089088
.du_total_err4299089120
.du_max_err_lsb4299089152
HMU_DIR_BIST_SIG04299088768
.reserv4299088800
.du0_err4299088832
.du1_err4299088864
.du2_err4299088896
.du3_err4299088928
.du4_err4299088960
.du5_err4299088992
.du6_err4299089024
.du7_err4299089056
.lru_err4299089088
.du_total_err4299089120
.du_max_err_lsb4299089152
HMU_DIR_BIST_SIG04299088768
.reserv4299088800
.du0_err4299088832
.du1_err4299088864
.du2_err4299088896
.du3_err4299088928
.du4_err4299088960
```

ТВГИ.00406-01 34 01

```
.du5_err4299088992
.du6_err4299089024
.du7_err4299089056
.lru_err4299089088
.du_total_err4299089120
.du_max_err_lsb4299089152
```

#

set freq_bnd

Установить значение регистра PMC_FREQ_BND

set freq_bnd

Usage:

```
set freq_bnd <node> <msb> <lsb> <value>
               <node> - CPU number of range: 0-0
```

```
               <msb> - most significant bit number of range: 0-31
```

```
               <lsb> - low significant bit number of range: 0-31 <=
```

msb

```
               <value> - value in [msb:lsb] : number of range: 0-0x2F
```

set freq_bnd 0 0 0 0

new Val: 0x80242424

*[5: 0] .core_divF 0x24

*[13: 8] .ocn_divF 0x24

*[21:16] .graph_divF 0x24

[31:31] .rmwen 1

Save Tune ? (y or n) y

Note: PMC_FREQ_CORE_0 mode is 5

Tunes will take Effect only after Reset

#

get freq_bnd

Выдать значение регистра PMC_FREQ_BND

get freq_bnd

Usage:

```
get freq_bnd <node>
               <node> is CPU number
```

get freq_bnd 0

CMOS pmc_freq_bnd[CPU0]

*[31: 0] = 0x80242424

*[5: 0] .core_divF 0x24

*[13: 8] .ocn_divF 0x24

*[21:16] .graph_divF 0x24

[31:31] .rmwen 1

PMC_FREQ_BND: cpu0 addr 0x11000110C *[31: 0] = 0x80242424

*[5: 0] .core_divF 0x24

*[13: 8] .ocn_divF 0x24

*[21:16] .graph_divF 0x24

[31:31] .rmwen 1

#

set freq_core_float

ТВГИ.00406-01 34 01

```

        <msb> - most significant bit num of range: 0-31
        <lsb> - low significant bit num of range: 0-31 <= msb
        <val> - val in [msb:lsb] : num of range: 0-0xFFFFFFFF
# set freq_ocn_float 0 0 0 0
new Val: 0
*[ 8: 0] T_lo_float=0
*[20:12] T_hi_float=0
[31:31] rmwen=0

Save Tune ? (y or n) y
Note: PMC_FREQ_OCN_FLOAT mode is 0
Set will take Effect only after Reset
#

```

```
get req_ocn_float
```

Выдать значение регистра PMC_FREQ_OCN_FLOAT

```

# get freq_ocn_float
Usage:
    get freq_ocn_float <node>
    <node> is CPU number of Range: 0-3
# get freq_ocn_float 0
CMOS pmc_freq_ocn_float[ CPU0 ]
*[31: 0] = 0
*[ 8: 0] T_lo_float=0
*[20:12] T_hi_float=0
[31:31] rmwen=0
PMC_FREQ_OCN_FLOAT: CPU0  addr 0x110001114 *[31: 0] = 0x801EF1F9
*[ 8: 0] T_lo_float=505
*[20:12] T_hi_float=495
[31:31] rmwen=1
#

```

```
set ocn_hmuil
```

Установить значение регистра OCN_HMUIL

```

# set ocn_hmuil
Usage:
    set ocn_hmuil <cpu> <msb> <lsb> <val>
    <cpu>: node# 0-3
    <msb>: hex|dec 0-31
    <lsb>: hex|dec 0-31
    <val>: hex|dec 0-0xffffffff
Result in NVRAM: ocn_hmuil.[msb:lsb] = val
# set ocn_hmuil 0 0 0 0
[31: 0] OCN_HMUIL=0xB28A
*[ 5 : 0] OCN_HMUIL_bit0=0xA
*[11: 6] OCN_HMUIL_bit1=0xA
*[17:12] OCN_HMUIL_bit2=0xB
[27:18] OCN_HMUIL_NoUse=0
[31:28] OCN_HMUIL_en=0

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get ocn_hmuil
```

Выдать значение регистра OCN_HMUIL

```
# get ocn_hmuil
Usage:
    get ocn_hmuil <cpu>
        <cpu>: node# 0-3
# get ocn_hmuil 0
[31: 0] OCN_HMUIL=0xB28A
*[5 : 0] OCN_HMUIL_bit0=0xA
*[11: 6] OCN_HMUIL_bit1=0xA
*[17:12] OCN_HMUIL_bit2=0xB
[27:18] OCN_HMUIL_NoUse=0
[31:28] OCN_HMUIL_en=0
#
```

```
set ocn_l3il
```

Установить значение регистра OCN_L3IL

```
# set ocn_l3il
Usage:
    set ocn_l3il <cpu> <msb> <lsb> <val>
        <cpu>: node# 0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff
Result in NVRAM: ocn_l3il.[msb:lsb] = val
# set ocn_l3il 0 0 0 0
[31: 0] OCN_L3IL=0x2CA208
*[5 : 0] l3il_bit0=8
*[11: 6] l3il_bit1=8
*[17:12] l3il_bit2=0xA
*[23:18] l3il_bit3=0xB
[31:24] Reserved=0

Save Tune ? (y or n) y
Tune effect after reset
#
```

```
get ocn_l3il
```

Выдать значение регистра OCN_L3IL

```
# get ocn_l3il
Usage:
    get ocn_l3il <cpu>
        <cpu>: node# 0-3
# get ocn_l3il 0
[31: 0] OCN_L3IL=0x2CA209
*[5 : 0] l3il_bit0=9
*[11: 6] l3il_bit1=8
*[17:12] l3il_bit2=0xA
*[23:18] l3il_bit3=0xB
[31:24] Reserved=0
#
```

```
set ocn_l3en
```

Установить значение регистра OCN_L3EN

```
# set ocn_l3en
Usage:
    set ocn_l3en <cpu> <msb> <lsb> <val>
    <cpu>: node# 0-3
    <msb>: hex|dec 0-31
    <lsb>: hex|dec 0-31
    <val>: hex|dec 0-0xffffffff

Result in NVRAM: ocn_l3en.[msb:lsb] = val
# set ocn_l3en 0 0 0 0
  [31: 0] OCN_L3EN=0
  *[15: 0] l3_msk=0
  [31:16] Reserved=0

Save Tune ? (y or n) y
Tune effect after reset
#
```

```
get ocn_l3en
```

Выдать значение регистра OCN_L3EN

```
# get ocn_l3en
Usage:
    get ocn_l3en <cpu>
    <cpu>: node# 0-3

# get ocn_l3en 0
  [31: 0] OCN_L3EN=0
  *[15: 0] l3_msk=0
  [31:16] Reserved=0
#
```

```
set mc_ecc
```

Установить/получить поля MC_ECC для любого CPU->Ch

```
# set mc_ecc
Usage:
    set mc_ecc <node> <field value>
    <node> - is CPU number of Range: 0-3
    <field value> -is hex or dec digit of Range: 0-
0x3030303030303030
    Ex:      CPU MC 0 1 2 3 4 5 6 7
           set mc_ecc 0 0x0001020301000302

# set mc_ecc 0 0xabcd
MC_ECC.[63:0] = 0xabcd
ERR ecc.mc[7]=0xcd > 3
ERR ecc.mc[6]=0xab > 3
OK ecc.mc[5]  MC_ECC =0 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =0
OK ecc.mc[4]  MC_ECC =0 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =0
OK ecc.mc[3]  MC_ECC =0 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
```

ТВГИ.00406-01 34 01

```
=0*[ 0: 0]ee =0
  OK ecc.mc[2]  MC_ECC =0 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =0
  OK ecc.mc[1]  MC_ECC =0 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =0
  OK ecc.mc[0]  MC_ECC =0 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =0
#
```

```
get mc_ecc
```

Выдать установленные значения поля MC_ECC для любого CPU→Ch

```
# get mc_ecc
Usage:
    get mc_ecc <node>
        <node> - is CPU number of Range: 0-3
        Ex: get mc_ecc 0

# get mc_ecc 0
MC_ECC.[63:0] = 0x101010101010101
OK ecc.mc[7]  MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =1
  OK ecc.mc[6]  MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =1
  OK ecc.mc[5]  MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =1
  OK ecc.mc[4]  MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =1
  OK ecc.mc[3]  MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =1
  OK ecc.mc[2]  MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =1
  OK ecc.mc[1]  MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =1
  OK ecc.mc[0]  MC_ECC =1 [31:16]secnt =0 [15: 2]uecnt =0*[ 1: 1]dmode
=0*[ 0: 0]ee =1
#
```

```
set cpu_mopma
```

Установка/вывод полей MC_MOPMa для требуемого ЦП

```
# set cpu_mopma
Usage:
    set cpu_mopma <cpu> <msb> <lsb> <val>
        <cpu>: node#    0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff

Result in NVRAM: cpu_mopma.[msb:lsb] = val
# set cpu_mopma 0 0 0 0
MC_MOPMa      [31:0 ]=0x1000000
BL            [ 1:0 ]=0
CL_2_2       [ 2:0 ]=0
BT           [ 3:0 ]=0
CL_6_4       [ 6:4 ]=0
TM           [ 7:7 ]=0
DR           [ 8:8 ]=0
```


ТВГИ.00406-01 34 01

```

twr_11_9          [11:9 ]=0
CL_12_12          [12:12]=0
twr_13            [13:13]=0
NoUse1            [15:14]=0
DE                [16:16]=0
dic               *[18:17]=0
tal               [20:19]=0
NoUse2            [22:21]=0
LEVEL             [23:23]=0
RTT               *[26:24]=1
TDQS              [27:27]=0
QOFF              [28:28]=0
NoUse3            [29:29]=0
yes_dic           >[30:30]=0
yes_RTT           >[31:31]=0

```

```

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_mopma
```

Выдать установленные значения полей MC_MOPMA для требуемого

ЦП

```

# get cpu_mopma
Usage:
    get cpu_mopma <cpu>
    <cpu>: node# 0-3
# get cpu_mopma 0
MC_MOPMa          [31:0 ]=0x1000000
BL                [ 1:0 ]=0
CL_2_2            [ 2:0 ]=0
BT                [ 3:0 ]=0
CL_6_4            [ 6:4 ]=0
TM                [ 7:7 ]=0
DR                [ 8:8 ]=0
twr_11_9          [11:9 ]=0
CL_12_12          [12:12]=0
twr_13            [13:13]=0
NoUse1            [15:14]=0
DE                [16:16]=0
dic               *[18:17]=0
tal               [20:19]=0
NoUse2            [22:21]=0
LEVEL             [23:23]=0
RTT               *[26:24]=1
TDQS              [27:27]=0
QOFF              [28:28]=0
NoUse3            [29:29]=0
yes_dic           >[30:30]=0
yes_RTT           >[31:31]=0

```

```
#
```

```
set cpu_mopmb
```

Установить/получить поля MC_MOPMb для требуемого ЦП

ТВГИ.00406-01 34 01

```

# set cpu_mopmb
Usage:
    set cpu_mopmb <cpu> <msb> <lsb> <val>
                <cpu>: node#    0-3
                <msb>: hex|dec 0-31
                <lsb>: hex|dec 0-31
                <val>: hex|dec 0-0xffffffff
Result in NVRAM: cpu_mopmb.[msb:lsb] = val
# set cpu_mopmb 0 0 0 0
MC_MOPMb      [31:0 ]=0x200
NoUse1       [ 2:0 ]=0
CWL          [ 5:3 ]=0
LP_ASR       [ 7:6 ]=0
NoUse2       [ 8:8 ]=0
Rtt_WR       *[10:9 ]=1
NoUse3       [11:11]=0
WRCRC        *[12:12]=0
TRRCID       [15:13]=0
MPRSEL       [17:16]=0
MPRO         [18:18]=0
GDM          [19:19]=0
PDA          [20:20]=0
TSR          [21:21]=0
FGRM        [24:22]=0
WCL          [26:25]=0
MPRRF        [28:27]=0
NoUse4       [30:29]=0
yes_Rtt_WR   >[31:31]=0

Save Tune ? (y or n) y
Tune effect after reset
#

get cpu_mopmb

```

Выдать установленные значения полей MC_MOPMB для требуемого

ЦП

```

# get cpu_mopmb
Usage:
    get cpu_mopmb <cpu>
                <cpu>: node# 0-3
# get cpu_mopmb 0
MC_MOPMb      [31:0 ]=0x200
NoUse1       [ 2:0 ]=0
CWL          [ 5:3 ]=0
LP_ASR       [ 7:6 ]=0
NoUse2       [ 8:8 ]=0
Rtt_WR       *[10:9 ]=1
NoUse3       [11:11]=0
WRCRC        *[12:12]=0
TRRCID       [15:13]=0
MPRSEL       [17:16]=0
MPRO         [18:18]=0
GDM          [19:19]=0
PDA          [20:20]=0
TSR          [21:21]=0
FGRM        [24:22]=0
WCL          [26:25]=0

```

ТВГИ.00406-01 34 01

```

MPRRF      [28:27]=0
NoUse4     [30:29]=0
yes_Rtt_WR >[31:31]=0
#

```

```
set cpu_mopmc
```

Установка/получение полей MC_MOPMc для требуемого ЦП

```

# set cpu_mopmc
Usage:
    set cpu_mopmc <cpu> <msb> <lsb> <val>
                <cpu>: node# 0-3
                <msb>: hex|dec 0-31
                <lsb>: hex|dec 0-31
                <val>: hex|dec 0-0xffffffff
Result in NVRAM: cpu_mopmc.[msb:lsb] = val
# set cpu_mopmc 0 0 0 0
MC_MOPMc      [31:0 ]=0
NoUse1        [ 0:0 ]=0
MPDM          [ 1:1 ]=0
TCRR          [ 2:2 ]=0
TCRM          [ 3:3 ]=0
IVM           [ 4:4 ]=0
NoUse2        [ 5:5 ]=0
CStoCMDL     [ 8:6 ]=0
SRA           [ 9:9 ]=0
RPTM          [10:10]=0
RDP           [11:11]=0
WRP           [12:12]=0
NoUse3        [15:13]=0
CAPM          [18:16]=0
CRCEC        [19:19]=0
CAPES        [20:20]=0
ODTIBPD      [21:21]=0
RTTPARK      [24:22]=0
CAPPE        *[25:25]=0
DM           [26:26]=0
WDBI         *[27:27]=0
RDBI         *[28:28]=0
NoUse4        [30:29]=0
yes_CAPPE    >[31:31]=0

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_mopmc
```

ЦП

Выдать установленные значения полей MC_MOPMc для требуемого

```

# get cpu_mopmc
Usage:
    get cpu_mopmc <cpu>
                <cpu>: node# 0-3
# get cpu_mopmc 0
MC_MOPMc      [31:0 ]=0

```

ТВГИ.00406-01 34 01

```

NoUse1      [ 0:0 ]=0
MPDM        [ 1:1 ]=0
TCRR        [ 2:2 ]=0
TCRM        [ 3:3 ]=0
IVM         [ 4:4 ]=0
NoUse2      [ 5:5 ]=0
CStoCMDL    [ 8:6 ]=0
SRA         [ 9:9 ]=0
RPTM        [10:10]=0
RDP         [11:11]=0
WRP         [12:12]=0
NoUse3      [15:13]=0
CAPM        [18:16]=0
CRCEC       [19:19]=0
CAPES       [20:20]=0
ODTIBPD     [21:21]=0
RTTPARK     [24:22]=0
CAPPE       * [25:25]=0
DM          [26:26]=0
WDBI        * [27:27]=0
RDBI        * [28:28]=0
NoUse4      [30:29]=0
yes_CAPPE   > [31:31]=0
#

```

```
set cpu_tim1
```

Установить/получить поля MC_TIM1 для требуемого ЦП

```

# set cpu_tim1
Usage:
    set cpu_tim1 <cpu> <msb> <lsb> <val>
                <cpu>: node#    0-3
                <msb>: hex|dec 0-31
                <lsb>: hex|dec 0-31
                <val>: hex|dec 0-0xffffffff
Result in NVRAM: cpu_tim1.[msb:lsb] = val
# set cpu_tim1 0 0 0 0
MC_TIM1     [31:0 ]=0
trrd_l      [ 3:0 ]=0
tfaw        [ 8:4 ]=0
trc         [14:9 ]=0
twtr_s      [17:15]=0
trrd_s      [20:18]=0
rcven_del   * [23:21]=0
tWrWr       * [27:24]=0
tRdRd       * [30:28]=0
trc_h       [31:31]=0

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_tim1
```

Выдать поля MC_TIM1 для требуемого ЦП

```

# get cpu_tim1
Usage:

```

ТВГИ.00406-01 34 01

```

get cpu_tim1 <cpu>
      <cpu>: node# 0-3
# get cpu_tim1 0
MC_TIM1 [31:0 ]=0
trrd_l [ 3:0 ]=0
tfaw [ 8:4 ]=0
trc [14:9 ]=0
twtr_s [17:15]=0
trrd_s [20:18]=0
rcven_del *[23:21]=0
tWrWr *[27:24]=0
tRdRd *[30:28]=0
trc_h [31:31]=0
#

```

```
set cpu_tim2
```

Установить/получить поля MC_TIM2 для нужного ЦП

```

# set cpu_tim2
Usage:
    set cpu_tim2 <cpu> <msb> <lsb> <val>
                <cpu>: node# 0-3
                <msb>: hex|dec 0-31
                <lsb>: hex|dec 0-31
                <val>: hex|dec 0-0xffffffff
Result in NVRAM: cpu_tim2.[msb:lsb] = val
# set cpu_tim2 0 0 0 0
MC_TIM2 [31:0 ]=0
trfc [ 7:0 ]=0
trfc_dlr [15:8 ]=0
tCCD_dlr [18:16]=0
tWrRd *[22:19]=0
3ds_fast [23:23]=0
trrd_s_h [24:24]=0
zqcs_period [31:25]=0

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_tim2
```

Выдать поля MC_TIM2 для нужного ЦП

```

# get cpu_tim2
Usage:
    get cpu_tim2 <cpu>
                <cpu>: node# 0-3
# get cpu_tim2 0
MC_TIM2 [31:0 ]=0
trfc [ 7:0 ]=0
trfc_dlr [15:8 ]=0
tCCD_dlr [18:16]=0
tWrRd *[22:19]=0
3ds_fast [23:23]=0
trrd_s_h [24:24]=0
zqcs_period [31:25]=0

```

#

set cpu_perf0

Установить/получить поля MC_PERF0 для требуемого ЦП

set cpu_perf0

Usage:

```

    set cpu_perf0 <cpu> <msb> <lsb> <val>
                <cpu>: node# 0-3
                <msb>: hex|dec 0-31
                <lsb>: hex|dec 0-31
                <val>: hex|dec 0-0xffffffff

```

Result in NVRAM: cpu_perf0.[msb:lsb] = val

set cpu_perf0 0 0 0 0

MC_PERF0 [31:0]=0x2806684

reg_num [0:0]=0

pbmask [1:1]=0

arp_en *[2:2]=1

flt_rdpr_type *[3:3]=0

flt_rdpr_sign *[5:4]=0

flt_brop [6:6]=0

cmdpack *[7:7]=1

sldrd_fast *[8:8]=0

rd_weight *[11:9]=3

flt_prio *[12:12]=0

apen *[13:13]=1

pt *[14:14]=1

rdpr_l *[20:15]=0

rdpr_h *[26:21]=0x14

rd_prio_rsv *[31:27]=0

Save Tune ? (y or n) y

Tune effect after reset

#

get cpu_perf0

Установить/получить поля MC_PERF0 для требуемого ЦП

get cpu_perf0

Usage:

```

    get cpu_perf0 <cpu>
                <cpu>: node# 0-3

```

get cpu_perf0 0

MC_PERF0 [31:0]=0x2806684

reg_num [0:0]=0

pbmask [1:1]=0

arp_en *[2:2]=1

flt_rdpr_type *[3:3]=0

flt_rdpr_sign *[5:4]=0

flt_brop [6:6]=0

cmdpack *[7:7]=1

sldrd_fast *[8:8]=0

rd_weight *[11:9]=3

flt_prio *[12:12]=0

apen *[13:13]=1

pt *[14:14]=1

rdpr_l *[20:15]=0

ТВГИ.00406-01 34 01

```
rdpr_h    *[26:21]=0x14
rd_prio_rsv  *[31:27]=0
#
```

```
set cpu_perf1
```

Установить/получить поля MC_PERF1 для требуемого ЦП

```
# set cpu_perf1
Usage:
    set cpu_perf1 <cpu> <msb> <lsb> <val>
        <cpu>: node#    0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff
Result in NVRAM: cpu_perf1.[msb:lsb] = val
# set cpu_perf1 0 0 0 0
MC_PERF1    [31:0 ]=0xFF1E
reg_num     [ 0:0 ]=0
ap_mgn     *[ 6:1 ]=0xF
NoUse1     [ 7:7 ]=0
sldrd     *[15:8 ]=0xFF
sldwr     *[23:16]=0
NoUse2     [31:24]=0

Save Tune ? (y or n) y
Tune effect after reset
#
```

```
cpu_perf1
```

Установить/получить поля MC_PERF1 для требуемого ЦП

```
# get cpu_perf1
Usage:
    get cpu_perf1 <cpu>
        <cpu>: node# 0-3
# get cpu_perf1 0
MC_PERF1    [31:0 ]=0xFF1E
reg_num     [ 0:0 ]=0
ap_mgn     *[ 6:1 ]=0xF
NoUse1     [ 7:7 ]=0
sldrd     *[15:8 ]=0xFF
sldwr     *[23:16]=0
NoUse2     [31:24]=0
#
```

```
set cpu_cfg
```

Установка/получение полей MC_CFG для требуемого ЦП

```
# set cpu_cfg
Usage:
    set cpu_cfg <cpu> <msb> <lsb> <val>
        <cpu>: node#    0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
```

ТВГИ.00406-01 34 01

```

                                <val>: hex|dec 0-0xffffffff
Result in NVRAM: cpu_cfg.[msb:lsb] = val
# set cpu_cfg 0 0 0 0
MC_CFG      [31:0 ]=0
ct0         [ 2:0 ]=0
ct1         [ 5:3 ]=0
pbm0        [ 7:6 ]=0
pbm1        [ 9:8 ]=0
rm          [10:10]=0
mode_3ds    [12:11]=0
mirr        [13:13]=0
sf          [16:14]=0
mt          *[17:17]=0
nosra       [18:18]=0
ptrr_mode   [20:19]=0
mcrc        [21:21]=0
odt_ext     [23:22]=0
pbswap      [24:24]=0
dqw         [26:25]=0
pda_sel     [31:27]=0

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_cfg
```

Вывод полей MC_CFG для требуемого ЦП

```

# get cpu_cfg
Usage:
        get cpu_cfg <cpu>
        <cpu>: node# 0-3

# get cpu_cfg 0
MC_CFG      [31:0 ]=0
ct0         [ 2:0 ]=0
ct1         [ 5:3 ]=0
pbm0        [ 7:6 ]=0
pbm1        [ 9:8 ]=0
rm          [10:10]=0
mode_3ds    [12:11]=0
mirr        [13:13]=0
sf          [16:14]=0
mt          *[17:17]=0
nosra       [18:18]=0
ptrr_mode   [20:19]=0
mcrc        [21:21]=0
odt_ext     [23:22]=0
pbswap      [24:24]=0
dqw         [26:25]=0
pda_sel     [31:27]=0
#

```

```
set cpu_ctl
```

Установить/получить поля MC_CTL для нужного ЦП

```

# set cpu_ctl
Usage:

```


ТВГИ.00406-01 34 01

```

set cpu_ctl <cpu> <msb> <lsb> <val>
      <cpu>: node# 0-3
      <msb>: hex|dec 0-31
      <lsb>: hex|dec 0-31
      <val>: hex|dec 0-0xffffffff
Result in NVRAM: cpu_ctl.[msb:lsb] = val
# set cpu_ctl 0 0 0 0
MC_CTL [31:0 ]=0x40002000
mcen [ 0:0 ]=0
phyupd [ 1:1 ]=0
mcinitreq [ 2:2 ]=0
phyinitreq [ 3:3 ]=0
mc_ps [ 4:4 ]=0
lpreq [ 5:5 ]=0
lpwkup [ 9:6 ]=0
upd0_en *[10:10]=0
parint_en [11:11]=0
phyint_en [12:12]=0
mi_bg *[13:13]=1
mrs_en [14:14]=0
dfi_freq [19:15]=0
phyreset_cfg [20:20]=0
phy_reset [21:21]=0
trwm [24:22]=0
tdly [27:25]=0
dmemint_en [28:28]=0
bridgeint_en [29:29]=0
mcln *[30:30]=1
mcstart [31:31]=0

```

```

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_ctl
```

Выдать поля MC_CTL для нужного ЦП

```

# get cpu_ctl
Usage:
      get cpu_ctl <cpu>
      <cpu>: node# 0-3
# get cpu_ctl 0
MC_CTL [31:0 ]=0x40002000
mcen [ 0:0 ]=0
phyupd [ 1:1 ]=0
mcinitreq [ 2:2 ]=0
phyinitreq [ 3:3 ]=0
mc_ps [ 4:4 ]=0
lpreq [ 5:5 ]=0
lpwkup [ 9:6 ]=0
upd0_en *[10:10]=0
parint_en [11:11]=0
phyint_en [12:12]=0
mi_bg *[13:13]=1
mrs_en [14:14]=0
dfi_freq [19:15]=0
phyreset_cfg [20:20]=0
phy_reset [21:21]=0
trwm [24:22]=0
tdly [27:25]=0

```

ТВГИ.00406-01 34 01

```

dmemint_en      [28:28]=0
bridgeint_en    [29:29]=0
mc1n            *[30:30]=1
mcstart        [31:31]=0
#

```

```
set cpu_odt
```

Установка/получение полей MC_ODT для требуемого ЦП

```

# set cpu_odt
Usage:
    set cpu_odt <cpu> <msb> <lsb> <val>
                <cpu>: node#    0-3
                <msb>: hex|dec 0-31
                <lsb>: hex|dec 0-31
                <val>: hex|dec 0-0xffffffff

Result in NVRAM: cpu_odt.[msb:lsb] = val
# set cpu_odt 0 0 0 0
MC_ODT      [31:0 ]=0x95650000
rdodt0     *[ 3:0 ]=0
rdodt1     *[ 7:4 ]=0
rdodt2     *[11:8 ]=0
rdodt3     *[15:12]=0
wrodt0     *[19:16]=5
wrodt0     *[23:20]=6
wrodt0     *[27:24]=5
wrodt0     *[31:28]=9

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_odt
```

Выдать полей MC_ODT для требуемого ЦП

```

# get cpu_odt
Usage:
    get cpu_odt <cpu>
                <cpu>: node# 0-3

# get cpu_odt 0
MC_ODT      [31:0 ]=0x95650000
rdodt0     *[ 3:0 ]=0
rdodt1     *[ 7:4 ]=0
rdodt2     *[11:8 ]=0
rdodt3     *[15:12]=0
wrodt0     *[19:16]=5
wrodt0     *[23:20]=6
wrodt0     *[27:24]=5
wrodt0     *[31:28]=9
#

```

```
set cpu_am0
```

ТВГИ.00406-01 34 01

Установить/получить поля MC_AM0 для требуемого ЦП

```
# set cpu_am0
Usage:
    set cpu_am0 <cpu> <msb> <lsb> <val>
        <cpu>: node#    0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff

Result in NVRAM: cpu_am0.[msb:lsb] = val
# set cpu_am0 0 0 0 0
MC_AM0    [31:0 ]=0xFFFFFFFF0
regnum    [ 1:0 ]=0
rt        *[ 3:2 ]=0
acrs      *[ 9:4 ]=0x3F
lbrs0     *[15:10]=0x3F
lbofl     *[19:16]=0xF
bgrs0     *[25:20]=0x3F
bgrs1     *[31:26]=0x3F

Save Tune ? (y or n) y
Tune effect after reset
#
```

```
get cpu_am0
```

Выдать поля MC_AM0 для требуемого ЦП

```
# get cpu_am0
Usage:
    get cpu_am0 <cpu>
        <cpu>: node# 0-3

# get cpu_am0 0
MC_AM0    [31:0 ]=0xFFFFFFFF0
regnum    [ 1:0 ]=0
rt        *[ 3:2 ]=0
acrs      *[ 9:4 ]=0x3F
lbrs0     *[15:10]=0x3F
lbofl     *[19:16]=0xF
bgrs0     *[25:20]=0x3F
bgrs1     *[31:26]=0x3F
#
```

```
set cpu_am1
```

Установить/получить поля MC_AM1 для нужного ЦП

```
# set cpu_am1
Usage:
    set cpu_am1 <cpu> <msb> <lsb> <val>
        <cpu>: node#    0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff

Result in NVRAM: cpu_am1.[msb:lsb] = val
# set cpu_am1 0 0 0 0
MC_AM1    [31:0 ]=0xFFFFFFFFC
```

ТВГИ.00406-01 34 01

```

regnum    [ 1:0 ]=0
pbb0      *[ 7:2 ]=0x3F
pbb1      *[13:8 ]=0x3F
rars0     *[19:14]=0x3F
rars1     *[25:20]=0x3F
cirs      *[31:26]=0x3F

```

```

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_am1
```

Установить/получить поля MC_AM1 для нужного ЦП

```

# get cpu_am1
Usage:
    get cpu_am1 <cpu>
    <cpu>: node# 0-3

# get cpu_am1 0
MC_AM1    [31:0 ]=0xFFFFFFFF
regnum    [ 1:0 ]=0
pbb0      *[ 7:2 ]=0x3F
pbb1      *[13:8 ]=0x3F
rars0     *[19:14]=0x3F
rars1     *[25:20]=0x3F
cirs      *[31:26]=0x3F
#

```

```
set cpu_ax0
```

Установить/получить поля MC_AX0 для требуемого ЦП

```

# set cpu_ax0
Usage:
    set cpu_ax0 <cpu> <msb> <lsb> <val>
    <cpu>: node# 0-3
    <msb>: hex|dec 0-31
    <lsb>: hex|dec 0-31
    <val>: hex|dec 0-0xffffffff

Result in NVRAM: cpu_ax0.[msb:lsb] = val
# set cpu_ax0 0 0 0 0
MC_AX0    [31:0 ]=0x3FFFFFFC
regnum    [ 1:0 ]=0
xlb0      *[ 9:2 ]=0xFF
xlb1      *[17:10]=0xFF
xbg0      *[25:18]=0xFF
xpb21     [31:26]=0

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_ax0
```

Выдать поля MC_AX0 для требуемого ЦП

ТВГИ.00406-01 34 01

```
# get cpu_ax0
Usage:
    get cpu_ax0 <cpu>
        <cpu>: node# 0-3

# get cpu_ax0 0
MC_AX0    [31:0 ]=0x3FFFFFFC
regnum    [ 1:0 ]=0
xlb0      *[ 9:2 ]=0xFF
xlb1      *[17:10]=0xFF
xbg0      *[25:18]=0xFF
xpb2l     [31:26]=0
#
```

```
set cpu_ax1
```

Установить/получить поля MC_AX1 для нужного ЦП

```
# set cpu_ax1
Usage:
    set cpu_ax1 <cpu> <msb> <lsb> <val>
        <cpu>: node# 0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff

Result in NVRAM: cpu_ax1.[msb:lsb] = val
# set cpu_ax1 0 0 0 0
MC_AX1    [31:0 ]=0x3FFFFFFC
regnum    [ 1:0 ]=0
xbg1      *[ 9:2 ]=0xFF
xpb0      *[17:10]=0xFF
xpb1      *[25:18]=0xFF
xpb2h     [31:26]=0

Save Tune ? (y or n) y
Tune effect after reset
#
```

```
get cpu_ax1
```

Выдать поля MC_AX1 для нужного ЦП

```
# get cpu_ax1
Usage:
    get cpu_ax1 <cpu>
        <cpu>: node# 0-3

# get cpu_ax1 0
MC_AX1    [31:0 ]=0x3FFFFFFC
regnum    [ 1:0 ]=0
xbg1      *[ 9:2 ]=0xFF
xpb0      *[17:10]=0xFF
xpb1      *[25:18]=0xFF
xpb2h     [31:26]=0
#
```

```
set cpu_ax2
```

Установить/получить поля MC_AX2 для нужного ЦП

```
# set cpu_ax2
Usage:
    set cpu_ax2 <cpu> <msb> <lsb> <val>
        <cpu>: node# 0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff

Result in NVRAM: cpu_ax2.[msb:lsb] = val
# set cpu_ax2 0 0 0 0
MC_AX2      [31:0 ]=0x3FFFFFFC
regnum      [ 1:0 ]=0
xcid0       *[ 9:20]=0xFF
xcid1       *[17:10]=0xFF
xcid2       *[25:18]=0xFF
NoUse       [31:26]=0xFF

    Save Tune ? (y or n) y
Tune effect after reset
#
```

```
get cpu_ax2
```

Выдать поля MC_AX2 для нужного ЦП

```
# get cpu_ax2
Usage:
    get cpu_ax2 <cpu>
        <cpu>: node# 0-3

# get cpu_ax2 0
MC_AX2      [31:0 ]=0x3FFFFFFC
regnum      [ 1:0 ]=0
xcid0       *[ 9:20]=0xFF
xcid1       *[17:10]=0xFF
xcid2       *[25:18]=0xFF
NoUse       [31:26]=0xFF
#
```

```
set cpu_errrec
```

Установка/получение полей MC_ERRREC для требуемого ЦП

```
# set cpu_errrec
Usage:
    set cpu_errrec <cpu> <msb> <lsb> <val>
        <cpu>: node# 0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff

Result in NVRAM: cpu_errrec.[msb:lsb] = val
# set cpu_errrec 0 0 0 0
MC_ERRREC   [31:0 ]=8
ermod       *[ 2:0 ]=0
adddel      *[ 5:3 ]=1
```

ТВГИ.00406-01 34 01

```

talon      [ 6:6 ]=0
statrst    [ 7:7 ]=0
phyrst     [ 8:8 ]=0
errlim     [13:9 ]=0
limstp     [14:14]=0
mr         [15:15]=0
mr_freq    [19:16]=0
alert_probe      [21:20]=0
rcdparen        [22:22]=0
alertSyncEn     [23:23]=0
errcnt         [31:24]=0

```

```

Save Tune ? (y or n) y
Tune effect after reset
#

```

```
get cpu_errrec
```

Выдать значения полей MC_ERRREC для требуемого ЦП

```

# get cpu_errrec
Usage:
    get cpu_errrec <cpu>
                <cpu>: node# 0-3
# get cpu_errrec 0
MC_ERRREC      [31:0 ]=8
ermod          *[ 2:0 ]=0
adddel         *[ 5:3 ]=1
talon          [ 6:6 ]=0
statrst        [ 7:7 ]=0
phyrst         [ 8:8 ]=0
errlim         [13:9 ]=0
limstp         [14:14]=0
mr             [15:15]=0
mr_freq        [19:16]=0
alert_probe    [21:20]=0
rcdparen       [22:22]=0
alertSyncEn    [23:23]=0
errcnt         [31:24]=0
#

```

```
set hc_ctrl
```

Установить/получить поля HC_CTRL для требуемого ЦП

```

# set hc_ctrl
Usage:
    set hc_ctrl <cpu> <bit> <val>
                <cpu>: node#    0-3
                <bit>: hex|dec 0-8
                <val>: hex|dec 0-1
Result in NVRAM & NBSR: HC_CTRL.[bit] = val
# set hc_ctrl 0 0 0
CPU0 CMOS HC_CTRL.[31:0] =6
[0] IOL_ERE:0
[1] DMAGE:1
[2] RFOE:1
[3] USTCM:0
[4] DSTCM:0

```

ТВГИ.00406-01 34 01

```
[5] ECRD:0
[6] ECWD:0
[7] DCAE:0
[8] P2PE:0
```

```
Save Tune ? (y or n) y
CPU0 HC_CTRL.DCAE:0
CPU0 NBSR HC_CTRL.[31:0] =6
[0] IOL_ERE:0
[1] DMAGE:1
[2] RFOE:1
[3] USTCM:0
[4] DSTCM:0
[5] ECRD:0
[6] ECWD:0
[7] DCAE:0
[8] P2PE:0
#
```

```
get hc_ctrl
```

Выдать поля HC_CTRL для требуемого ЦП

```
# get hc_ctrl
Usage:
    get hc_ctrl <cpu>
        <cpu>: node# 0-3
# get hc_ctrl 0
CPU0 CMOS HC_CTRL.[31:0] =6
[0] IOL_ERE:0
[1] DMAGE:1
[2] RFOE:1
[3] USTCM:0
[4] DSTCM:0
[5] ECRD:0
[6] ECWD:0
[7] DCAE:0
[8] P2PE:0
#
```

```
set hc_int_mask
```

Установить/получить поля HC_INT_MASK для требуемого ЦП

```
# set hc_int_mask
Usage:
    set hc_int_mask <cpu> <msb> <lsb> <val>
        <cpu>: node# 0-3
        <msb>: hex|dec 0-31
        <lsb>: hex|dec 0-31
        <val>: hex|dec 0-0xffffffff
Result in NVRAM: hc_int_mask.[msb:lsb] = val
# set hc_int_mask 0 0 0 0
HC_INT_MASK.[31:0] = 0x7E
[0] UO: 0
[1] UR: 1
[2] CA: 1
[3] UA: 1
[4] WRO:1
```


ТВГИ.00406-01 34 01

```
[5] IS: 1
[6] UCR:1
[31:7] Rsvd:0
```

```
Save Tune ? (y or n) y
Tune effect after reset
#
```

```
get hc_int_mask
```

Выдать поля HC_INT_MASK для требуемого ЦП

```
# get hc_int_mask
Usage:
    get hc_int_mask <cpu>
    <cpu>: node# 0-3
# get hc_int_mask 0
HC_INT_MASK.[31:0] = 0x7E
[0] UO: 0
[1] UR: 1
[2] CA: 1
[3] UA: 1
[4] WRO:1
[5] IS: 1
[6] UCR:1
[31:7] Rsvd:0
#
```

```
set l3_imsk
```

Установить/получить значение регистра L3_IMSK

```
# set l3_imsk
Usage:
    set l3_imsk <cpu> <msb> <lsb> <val>
    <cpu>: node# 0-3
    <msb>: hex|dec 0-31
    <lsb>: hex|dec 0-31
    <val>: hex|dec 0-0xffffffff
Result in NVRAM: l3_imsk.[msb:lsb] = val
# set l3_imsk 0 0 0 0
[31: 0] L3_IMSK=0
*[0 : 0] ecc_dm=0
[3 : 1] Reserved=0
*[4 : 4] ecc_ld=0
[7 : 5] Reserved=0
*[8 : 8] par_irq=0
*[9 : 9] par_srq=0
*[10:10] par_dat_hdr=0
*[11:11] par_dat_bdy=0
*[12:12] par_ack=0
*[13:13] par_hak=0
*[14:14] par_rls=0
[15:15] Reserved=0
*[16:16] pmon=0
[31:17] Reserved=0
```

```
Save Tune ? (y or n) y
Tune effect after reset
```

#

get l3_imsk

Выдать значение регистра L3_IMSK

get l3_imsk

Usage:

```
get l3_imsk <cpu>
      <cpu>: node# 0-3
```

get l3_imsk 0

[31: 0] L3_IMSK=0

*[0 : 0] ecc_dm=0

[3 : 1] Reserved=0

*[4 : 4] ecc_ld=0

[7 : 5] Reserved=0

*[8 : 8] par_irq=0

*[9 : 9] par_srq=0

*[10:10] par_dat_hdr=0

*[11:11] par_dat_bdy=0

*[12:12] par_ack=0

*[13:13] par_hak=0

*[14:14] par_rls=0

[15:15] Reserved=0

*[16:16] pmon=0

[31:17] Reserved=0

#

set kpi_mask

Установить/получить маску EIOH

set kpi_mask

Usage:

```
set kpi_mask <value>
      <value> is hex or dec digit of range: 1-0xf
```

set kpi_mask 1

OK EIOH Mask 0x1

Save Tune ? (y or n) y

Tunes will take Effect only after Reset

#

get kpi_mask

Выдать маску EIOH

get kpi_mask

Usage:

get kpi_mask

get kpi_mask

OK EIOH Mask 0x1

#

7.1.18. Команды конфигурирования интерфейса I2C

```
set i2cb
```

Установить 8-битный обмен по интерфейсу I2C

```
# set i2cb
```

```
Writing a byte 00 to I2C device 0:00, register 00: error: I/O error
```

```
# set i2cb 00
```

```
Usage:
```

```
set i2c{s} <bus> <address> <register> <data>
```

```
{s} - data size: b,h,w,l (byte,half,word,long)
```

```
<bus> - iohub i2c bus (0-3)
```

```
<address> - 8-bit slave address of required device (0x00-
```

```
0xFE)
```

```
<register> - device register (0x00-0xFF)
```

```
<data> - data to write to register
```

```
# set i2cb 00 0 0 0
```

```
Writing a byte 00 to I2C device 0:00, register 00: error: I/O error
```

```
#
```

```
get i2cb
```

Выдать наличие флага 8-битного доступа к I2C.

I2C 8-bit Access; Param: bus, adr, reg, [data]

```
# get i2cb
```

```
Usage:
```

```
get i2c{b} <bus> <address> <register>
```

```
{b} - data size byte
```

```
<bus> - iohub i2c bus (0-3)
```

```
<address> - 8-bit slave address of required device (0x00-
```

```
0xFE)
```

```
<register> - device register (0x00-0xFF)
```

```
# get i2cb 0 0 0
```

```
Rd a byte from I2C device 0:00, register 00: error: No such file or
directory
```

```
#
```

```
set i2cw
```

Установить 32-битный обмен по интерфейсу I2C

```
# set i2cw
```

```
Usage:
```

```
set i2c{s} <bus> <address> <register> <data>
```

ТВГИ.00406-01 34 01

```

    {s} - data size: b,h,w,l (byte, half, word, long)
    <bus> - iohub i2c bus (0-3)
    <address> - 8-bit slave address of required device (0x00-
0xFE)
    <register> - device register (0x00-0xFF)
    <data> - data to write to register

# set i2cw 0 0 0 0
Writing a byte 0000 to I2C device 0:00, register 0000: error: I/O error

#

get i2cw

```

**Выдать наличие флага 32-битного доступа к I2C.
I2C 32-bit Access; Param: bus, adr, reg, [data]**

```

# get i2cw
Usage:
    get i2c{b} <bus> <address> <register>
    {b} - data size byte
    <bus> - iohub i2c bus (0-3)
    <address> - 8-bit slave address of required device (0x00-
0xFE)
    <register> - device register (0x00-0xFF)

# get i2cw 0 0 0
Rd a byte from I2C device 0:00, register 0000: error: No such file or
directory
#

set i2ch

```

Установить 16-битный обмен по интерфейсу I2C

```

# set i2ch
Usage:
    set i2c{s} <bus> <address> <register> <data>
    {s} - data size: b,h,w,l (byte, half, word, long)
    <bus> - iohub i2c bus (0-3)
    <address> - 8-bit slave address of required device (0x00-
0xFE)
    <register> - device register (0x00-0xFF)
    <data> - data to write to register

# set i2ch 0 0 0 0
Writing a word 0000 to I2C device 0:00, register 00: error: I/O error

#

get i2ch

```

Выдать наличие флага 16-битный доступ I2C;

I2C 16-bit Access; Param: bus, adr, reg, [data]
get i2ch

ТВГИ.00406-01 34 01

```
Usage:
    get i2c{b} <bus> <address> <register>
        {b}          - data size byte
        <bus>        - iohub i2c bus (0-3)
        <address>    - 8-bit slave address of required device (0x00-
0xFE)
        <register>   - device register (0x00-0xFF)

# get i2ch 0 0 0
Rd a word from I2C device 0:00, register 00: error: No such file or
directory
#
```

```
set i2cl
```

Установить 64-битный обмен по интерфейсу I2C

```
# set i2cl
Usage:
    set i2c{s} <bus> <address> <register> <data>
        {s} - data size: b,h,w,l (byte, half, word, long)
        <bus> - iohub i2c bus (0-3)
        <address> - 8-bit slave address of required device (0x00-
0xFE)
        <register> - device register (0x00-0xFF)
        <data> - data to write to register

# set i2cl 0 0 0 0
Writing a halfword 0000 to I2C device 0:00, register 00: error: I/O
error
#
```

```
get i2cl
```

Выдать наличие флага 64-битный доступ I2C**I2C 64-bit Access; Param: bus, adr, reg, [data]**

```
# get i2cl
Usage:
    get i2c{b} <bus> <address> <register>
        {b}          - data size byte
        <bus>        - iohub i2c bus (0-3)
        <address>    - 8-bit slave address of required device (0x00-
0xFE)
        <register>   - device register (0x00-0xFF)

# get i2cl 0 0 0
Rd a halfword from I2C device 0:00, register 00: error: No such file or
directory
#
```

```
set drive_init_delay
```

Задать задержку в секундах перед началом инициализации внешних устройств

```
# set drive_init_delay
Usage: set drive_init_delay <delay>
  delay - delay in seconds, before any drive initialization starts
supported values: 0, 1, 2, 3, 6, 10, 20, 30
EXAMPLE: set ide init_delay 3
# set drive_init_delay 3
#
```

```
get drive_init_delay
```

Вывод текущей задержки в секундах перед началом обращения к устройствам

```
# get drive_init_delay
Usage: get drive_init_delay
# get drive_init_delay
3
#
```

```
get auto_install
```

Выдать установленный флаг Вкл/Выкл автоматическая установка ОС через AoE.

```
# get auto_install
USAGE:
  get auto_install

# get auto_install
1
#
```

```
set test_mem
```

Включить/Выключить тестирование ОЗУ при включении ВК

```
# set test_mem
Usage:
  set test_mem 0/1
# set test_mem 1
#
```

```
get test_mem
```

Вывести значение флага

Включить/Выключить тестирование ОЗУ при включении ВК

```
# get test_mem
Usage:
    get test_mem
# get test_mem
1
#
```

get map

Выдать карту NVRAM и установленных параметров HW CMOS

```
#get map
```

7.1.20. Команда Перезапуск

Reset

Команда Перезапуск

```
#reset
```

7.1.21. Команда для перехода в программный модуль специальных тестов

tests

**Команда для перехода в программный модуль
специальных тестов**

```
#tests
```

**Выход из программного модуля специальных тестов
осуществляется нажатием клавиши [Esc].**

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

АПМДЗ	Аппаратно-программный модуль доверенной загрузки
ВК	Вычислительный комплекс
ВУ	Вычислительное устройство
МН	Машинный носитель
ОС	Операционная система
ПНС	Программа начального старта
ТЛ	Терминал

