CHARON-AXP *for Linux®*

The Hewlett-Packard Alpha Virtualization Layer for Linux Host System

User guide for Production release version 1.1 (Build 129-04)

Applies to the following Stromasys products:
CHARON-AXP/4100 for Linux x64
CHARON-AXP/DS10 for Linux x64
CHARON-AXP/DS20 for Linux x64
CHARON-AXP/ES40 for Linux x64
CHARON-AXP/GS80 for Linux x64
CHARON-AXP/GS160 for Linux x64
CHARON-AXP/GS320 for Linux x64

Document number: 60-16-010

Version: 005

Last update: 25 August 2011

1 Preface

1.1 Ownership Notice

Stromasys SA, Geneva, Switzerland, owns all rights, including proprietary rights, copyrights, trademarks, and world-wide distribution rights to a methodology for the execution of HP Alpha applications and system software by means of a software virtualization layer, henceforth referred to as CHARON-AXP. The right to use CHARON-AXP software is governed by a license allowing the execution of the software on a single computer system. The CHARON-AXP license does not transfer ownership of the CHARON-AXP encrypted binary executable, nor does it provide any rights to decrypt, view, analyze, copy or reverse engineer the CHARON-AXP binary or source code. Possession and use of the software described in this publication is authorized only pursuant to a valid license.

Stromasys makes no representations that the use of the CHARON-AXP software as described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

1.2 Trademarks

The CHARON name with logo is a trademark of Stromasys. AXP, XMI, VMS and OpenVMS are trademarks of the Hewlett-Packard Company. All other trademarks and registered trademarks are the property of their respective holders.

1.3 Life support applications

The CHARON products of Stromasys are not designed for use in systems where malfunction of a CHARON product can reasonably be expected to result in a personal injury. Stromasys' customers using or selling our CHARON products for use in such applications do so at their own risk and agree to fully indemnify Stromasys for any damages resulting from such improper use or sale.

2 Content

3 This manual

Stromasys has been building cross platform computer system virtualization products since 1999. The CHARON-AXP product line, which provides Hewlett-Packard Alpha hardware functionality as a virtualization layer on industry standard servers, has followed a development path comparable to the original DEC (now HP) Alpha hardware.

For even higher performance and reliability, CHARON-AXP can be clustered with shared storage or network clusters. With this technology, it is possible to replace large DEC Alpha data centers with a single rack of modern servers.

To get the best performance from the CHARON-AXP virtualization layer, it is essential to use a high performance x64 Linux host system. This manual provides the guidelines for host system selection, CHARON-AXP installation and operation.

In 2011 Stromasys had released first Production CHARON-AXP versions for Linux. Stromasys products and virtual AXP systems covered in this manual are:

Product: CHARON-AXP/4100 for Linux x64 includes the following virtual AXPs:

- **CHARON-AXP/AS400**, a single x64 CPU HP AlphaServer replacement.
- **CHARON-AXP/AS800**, a single x64 CPU HP AlphaServer replacement.
- **CHARON-AXP/AS1000**, a single x64 CPU HP AlphaServer replacement.
- **CHARON-AXP/AS1000A**, a single x64 CPU HP AlphaServer replacement.
- **CHARON-AXP/AS1200**, a 2 x64 CPU HP AlphaServer replacement.
- **CHARON-AXP/AS2000**, a 2 x64 CPU HP AlphaServer replacement.
- **CHARON-AXP/AS2100**, a 4 x64 CPU HP AlphaServer replacement.
- **CHARON-AXP/AS4000**, a 2 x64 CPU HP AlphaServer replacement.
- **CHARON-AXP/AS4100**, a 4 x64 CPU HP AlphaServer replacement.

Product: CHARON-AXP/DS10 for Linux x64 includes the following virtual AXPs:

- **CHARON-AXP/DS10L**, a single x64 CPU HP AlphaServer replacement.

Product: CHARON-AXP/DS20 for Linux x64 includes the following virtual AXPs:

- **CHARON-AXP/DS20**, a 2 x64 CPU HP AlphaServer replacement.

Product: CHARON-AXP/ES40 for Linux x64 includes the following virtual AXPs:

- **CHARON-AXP/ES40**, a 4 x64 CPU HP AlphaServer replacement.

Product: CHARON-AXP/GS80 for Linux x64 includes the following virtual AXPs:

- **CHARON-AXP/GS80**, an 8 x64 CPU HP AlphaServer replacement.

Product: CHARON-AXP/GS160 for Linux x64 includes the following virtual AXPs:

- **CHARON-AXP/GS160**, a 16 x64 CPU HP AlphaServer replacement.

Product: CHARON-AXP/GS320 for Linux x64 includes the following virtual AXPs:

- **CHARON-AXP/GS320**, a 32 x64 CPU HP AlphaServer replacement.

4 Overview

Modern software operating systems contain a hardware abstraction layer (or HAL). The HAL creates a software layer on top of the hardware to "virtualize" the functionality of the hardware components. The CHARON-AXP products are essentially HALs of complete HP Alpha systems, including the HP Alpha I/O devices. They are mathematically precise models of HP Alpha hardware, and contain modules of HP ALPHA CPUs, the console subsystem, the buses and I/O adapters, disks and tapes.

After installation of CHARON-AXP on a general purpose host platform, it provides an exact model of a working HP Alpha system. On this 'virtual' system you install your HP Alpha operating system and HP Alpha applications, just as if you had purchased new HP Alpha hardware. No conversion or sources are needed, and you boot your HP Alpha system as usual. The CHARON-AXP systems execute the same binary HP Alpha code and the same I/O drivers as on the original hardware. We tested with the original HP Alpha hardware diagnostics to verify compatibility.

What you obtain is an HP Alpha, typically running at comparable speed and with a significantly smaller footprint, a reduction in cost of maintenance and energy consumption. An additional advantage of CHARON-AXP over HP Alpha hardware is the scalability with its host system. CHARON-AXP performance is proportional to the host system performance, and every time you move to a faster host system your ‘virtual Alpha’ will also get faster.

Another improvement over the hardware is the amount of memory each model of CHARON-AXP supports; most emulated models supports up to 32 GB of operating memory.

This manual covers:

- The selection of a suitable host system, essentially a Linux multi-core server configured for the specific requirements of a CHARON-AXP product. Each product has its optimal host platform to get the best HP Alpha system performance. Ask Stromasys or one of its Resellers for configuration details for your specific system requirements.
- The installation process of the CHARON-AXP product, which is not significantly different from the installation of typical Linux applications.
- The CHARON-AXP configuration settings that allow you to specify the HP Alpha system configuration of your choice.
- The HP Alpha software installation process is not described in detail, since it is identical to the installation on HP Alpha hardware, and your HP Alpha software documentation applies. Solutions are provided to transfer the contents of the existing HP Alpha system and user disks, avoiding a complete system re-installation in most cases.

Like the original HP Alpha system, CHARON-AXP can run the same supported operating systems, such as Tru64 and OpenVMS. Windows NT for Alpha and Linux for Alpha are not supported.

.

5 Host system selection

This chapter describes the minimum hardware and software requirements the host system must meet for the CHARON-AXP virtualization layer to work properly. Some requirements are checked during installation and/or execution time. If these limits are not met, CHARON-AXP will not install or operate. Other limits are 'soft' and invoke a performance reduction ('safe mode') as described in this manual below.

5.1 Host operating system requirements

The CHARON-AXP virtualization layers are designed for a Linux server platform.

Currently supported Linux x64 versions:

- *Fedora 15*
- *Red Hat Enterprise Linux 6*

STROMASYS provides separate distribution packages for each supported Linux version.

The CHARON-AXP virtualization layers may also work on some other versions of Linux (including most modern ones), but STROMASYS cannot guarantee of CHARON-AXP proper functioning if any other version of Linux is used.

5.2 Host system hardware

5.2.1 CPU selection

The CHARON-AXP products require a multi-processor host system for their operation. The host system must have a physical CPU core available for each virtual Alpha CPU. The number of extra CPU cores required depends on the particular configuration and operation conditions. The optimal configuration is achieved when on top load you have at least one host CPU (core) idle 100% available for the host operation system use. Leaving too little host CPU cores to the I/O will result in performance reduction and malfunction especially in SMP environment.

The recommended host configurations for the specific CHARON-AXP products are as follows:

- For systems with light load, number of available physical CPU cores should be equal or greater than 1.5 times number of emulated AXP CPUs. For example, for lightly loaded GS80 system with 8 AXP cores a hosting server with at least 12 CPU cores are required.
- For systems with medium to heavy load, number of available physical CPU cores should be equal or greater than 2 times number of emulated AXP CPUs. For

example, for heavy loaded GS80 system with 8 AXP cores a hosting server with at least 16 CPU cores is required.

- CPU selection recommendations:
    - Generally Intel CPUs give advantage to CHARON-AXP over AMD CPUs.
    - For configurations with 8 or less virtual AXP CPU cores: Intel Xeon 5600 series or newer, at least 3GHz;
    - For configurations with 7 or more virtual AXP CPU cores: Intel Xeon 7500 series or newer, at least 2.26 GHz.
    - If AMD CPUs are the only available option, Opteron 6100 series, at least 2.2 GHz. (AMD CPUs older than K10 do not support cmpxchg16b instruction required for normal CHARON-AXP SMP operations).

5.2.2 Host system hardware platform recommendations

HP Proliant server products (ML-series towers, DL-series rack mount or BL-series blade servers) with sufficient CPU cores, memory, storage, and network adapter capacity are recommended.

For predictable HP Alpha performance the host system must be dedicated to the CHARON-AXP virtualization layer, with the possible exception of a co-resident HP Alpha console terminal.

5.2.3 Host system memory

The minimum host memory size depends on the amount of HP Alpha memory that is requested from the HP Alpha virtualization layer and on the number of CHARON instances running on one host. As a rule of thumb, the minimum host memory is the amounts of HP Alpha memory multiplied by the number of the instances +2 GB, with a minimum of 2 GB.

The maximum amount of HP Alpha memory that can be created in the CHARON-AXP products and is supported by OpenVMS/Alpha is 32 GB.

5.2.4 Disk storage

The CHARON-AXP virtualization layer requires approximately 30 MB disk space, not counting any (virtual) HP Alpha disks. HP Alpha disks can be in the form of physical disks (locally or on an external storage subsystem) or as HP Alpha disk images, which appear as standard Linux files. It is not recommended to store virtual disk images on non-Linux native file systems (such as MSDOS, NTFS and etc.), since read-write support for such files depends on Linux kernel version thus it may be limited and not sufficient to CHARON-AXP

When HP Alpha disk images are used to represent HP Alpha disk drives, the disk image files have the same size in MBs as the equivalent HP Alpha disk hardware, regardless of their degree of utilization.

When physical disks are used for the virtual Alpha, these disks are connected as SCSI devices to the host platform (locally, via FibreChannel or iSCSI), regardless of the disk architecture configured in the HP Alpha environment. These physical disks are formatted by the HP Alpha operating system and cannot be used by the host system.

5.2.5 Ethernet adapters

In CHARON-AXP, each HP Alpha Ethernet adapter must use a dedicated physical Ethernet adapter, not used by hosting Linux system (IP or any other protocols, including virtualization daemons, etc). The adapter must support dynamic MAC address changes (in essence does not require a reboot of the host system to reload a MAC address). Most modern adapters support necessary functionality. There are two reasons for the requirement of dedicated Ethernet adapters:

- A host system protocol of the same type (e.g. TCP/IP) would interfere with the protocol running on its virtual instance.
- A virtual network adapter uses special code that excludes access from the external network to anything but the Ethernet drivers running on the virtual HP Alpha system. This prevents penetration of malicious code into the host system from the external network.

The recommended way to dedicate an Ethernet adapter to CHARON-AXP is through Ethernet adapter configuration script located in `"/etc/sysconfig/network-scripts/ifcfg-ethN"` (where N is number of the interface to be used for CHARON-AXP). It is absolutely necessary to remove all the IP-setup related parameters.

Example:

```
# Intel Corporation 82540EM Gigabit Ethernet Controller

DEVICE=eth1

HWADDR=00:07:E9:17:DF:71

ONBOOT=no
```

5.3 Specific account to run CHARON-AXP

It is recommended to create user `"`*`charon`*`"` prior installation of CHARON-AXP and use it for running CHARON-AXP. This user must have rights to logon locally and it must have permissions to write to `"/var/lock"` directory. If it is planned to have direct access to host devices (such as physical CD/DVD drives, disks, etc) the user `"`*`charon`*`"` must be a member of the groups `"disk"` and `"`*`cdrom`*`".` If some physical serial lines are used by CHARON-AXP the `"`*`charon`*`"` user must be a member of the `"`*`dialout`*`"` group. Same requirements apply to any other user that is going to run CHARON-AXP.

Privileges for CHARON-AXP executables are assigned automatically during installation by CAPABILITIES kernel feature. If some capabilities are missing (depending on specific

CHARON-AXP configuration), CHARON-AXP reports it to its log file. In this case please use "*setcap*" utility to assign required additional rights.

If CHARON-AXP is started from the "*root*" account it has all the privileges by default.

It is always required to install CHARON-AXP from "*root*" account, whether any other possible accounts having the features described before can be used for running CHARON-AXP.

5.4 Other host system requirements

The host system must provide a USB port for the USB license key. The license key is used constantly by CHARON-AXP during its runtime; it is recommended to connect the key directly to the system USB hub and not via an external USB hub which can cause access problems. Key disconnection causes termination of operation within a few minutes. Note that a quick reconnection of the key might not cancel termination.

The CHARON-AXP virtualization layer interacts directly in several areas with the host system hardware. Where possible without compromising reliability, virtual peripherals are 'mapped' to the local hardware. Some host peripherals that work in Linux will not function correctly with the CHARON-AXP layer. For example, Ethernet adapters which cannot change their MAC address without a power cycle and some classes of peripherals connected to the host system via USB or Firewire connections.

It is recommended to use "*xterm*", Linux text console or "*screen*" multi-terminal as terminal to start CHARON-AXP, since these terminals provide best overall usability in Tru64/OpenVMS.

Note that the layout of the right part of PC keyboard is not mapped to the layout of VT100 keyboard by default. There is a workaround of this problem - usage of special script created by Geoff Kingsmill:

http://www.decuslib.com/decus/freewarev50/decxterm/decxterm

After execution of the script *xterm* will run correctly with CHARON-AXP.

Please also note that *telnet* session to CHARON-AXP console ports does not support ESC sequences of the VT100 by default. To enable it open *xterm*, connect to CHARON-AXP and press "*Ctrl-]*" once connected. Then issue a command "*mode char*" to enable “character” mode. If it is also required to map right part of the keyboard correctly please use the *xterm* started by the Geoff Kingsmill’s script (see above).

To control a number of available to CHARON-AXP host CPUs and their mapping use "*taskset*" Linux command in the following ways:

1. "*taskset -c <list of CPUs> CMD*", where “list of CPUs” is a list of CPUs (ranges or/and CPU numbers separated by comma) available to the process “CMD”.

Example: "`taskset -c 3,6-8 xterm`" defines that only the CPUs number 3,6,7,8 can be used by the process "xterm" (terminal) and by all the applications that start from it (CHARON-AXP in our case).

Running CHARON-AXP with predefined affinity must be performed in 2 stages:

a. "`taskset`" command is performed to get the customized terminal ("xterm"), from which CHARON-AXP emulator is going to be launched.
b. CHARON-AXP instance starts with all required parameters from this terminal ("xterm").

2. "`taskset -c <list of CPUs> -p $$`", where "`list of CPUs`" is a list of CPUs (ranges or/and CPU numbers separated by comma) available to the current shell (its PID is stored in the "$$" variable) . This command assumes that CHARON-AXP is going to be started from this particular shell.

   Example: "`taskset -c 1,2 -p $$`" defines that only the CPUs number 1 and 2 can be used by the current shell and by all the applications that start from it.

   This command must be issued before CHARON-AXP startup from this particular shell.

Either way is correct for setting desired CPU affinity for CHARON-AXP.

5.5 General performance considerations

The configurations referred to earlier in this chapter were target systems used for best performance during product design. The *functionality* of the HP Alpha virtualization layer is the same (in essence an accurate model of the corresponding HP Alpha system) for every host platform that meets the minimum requirements. The effective HP Alpha system *performance* delivered by CHARON-AXP depends on the host system. This allows for instance CHARON-AXP/ES40 to meet or exceed HP AlphaServer ES40 performance when executing on an HP Proliant. CHARON-AXP delivers approximately 380 SPEC2000 INT and 370 SPEC2000 FP per CPU when run on a Xeon 5680 host.

For lower performance requirements, CHARON-AXP can be used on smaller platforms. Since each of the virtual HP Alpha components puts its own requirements on the host system, it is important to look at your specific requirements before deciding what type of host system to use.

Experience shows that the three main areas of performance consideration are: HP Alpha CPU performance, disk I/O speed and network connections.

5.5.1 AXP CPU performance

The component in the virtualization layer that creates a HP Alpha CPU runs several concurrent tasks using a complex proprietary algorithm to optimize performance.

Above 2 GHz host CPU frequency, the memory bandwidth and latency becomes an important virtualization layer performance factor in the current host CPU architecture. Important parameters are host cache memory size (the larger the better) and host memory latency (the lower the better). In addition, the HP Alpha CPU floating point performance is quite dependent on the host CPU design.

The multiple CPU emulation processes that can run in the virtualization layer require a significant amount of host system memory, as specified earlier in this chapter. If less host memory is momentarily available (for example, because another application has started on the same host system), the CPU emulation process becomes less effective and can shut down completely, reducing performance. Therefore, concurrent operation of the CHARON-AXP virtualization layer with other applications on the same system is not recommended.

5.5.2 Disk I/O subsystem

CHARON-AXP Disk I/O throughput scales with the host I/O bandwidth and can exceed that of hardware HP Alpha systems with an order of magnitude. In general disk I/O is rarely a bottleneck.

5.5.3 Network connections

On a high performance host platform the virtual HP ALPHA Ethernet adapter operates approximately the same speed (1 Gbps) as counterpart, but it will not always reach the full 1 Gbps throughput of modern adapters. 1 Gbps host adapters can be used in most cases, and multiple adapters can be configured.

The use of multiple adapters will not necessary increase aggregate throughput beyond that of a single 1 Gbps host adapter. Dropping incoming packets due to temporary overload is acceptable (this happens on hardware HP Alpha systems as well) if the communications protocol can retransmit lost packets in time. For sensitive protocols, (i.e. the communication between instances of the OpenVMS distributed lock manager), configuring a separate Ethernet link reduces the risk of critical packet loss.

5.6 Enhancing virtualization layer reliability

CHARON-AXP executes a number of interrelated processes; each needs sufficient host system performance to provide a stable system. At several levels CHARON-AXP protects itself against a lack of host system capabilities:

- If the frequency of any of the host CPUs is below 1400 MHz, CHARON-AXP execution will terminate. Note that laptop or desktop systems in low power mode often reduce the clock frequency of their CPU(s) below their rated speed. Disable this feature via BIOS.
- If the number of host CPUs is less than requested, execution stops and the virtual layer shuts down completely.

- When insufficient HP Alpha memory can be locked in physical host memory, safe mode is entered to reduce memory requirements. Below a critical size, the virtualization layer shuts down, dependent on the model being virtualized.
- Additional host system load due to other applications running concurrently can prevent timely access to the USB license key, causing CHARON-AXP not to start or to shut down.
- It is possible to run two or more CHARON-AXP virtualization layers on the same host system, once the number of the host CPU (cores) permits the multi-instance operation as well as product license.

As far as possible, a lack of host system resources is reported in the CHARON-AXP log file.

For production use, CHARON-AXP should use a dedicated host system.

6 AXP hardware compatibility

The CHARON-AXP virtualization layers are tested with the UETP set of tests. HP has verified that the CHARON-AXP test results correspond to correctly functioning HP Alpha hardware.

HP provides OpenVMS and layered product licenses for the transfer from a hardware (HP Alpha) to CHARON-AXP; see the following web page:

http://h71000.www7.hp.com/openvms/vax-emulator.html

When CHARON-AXP is running on HP products, the transfer licenses maintain the HP OpenVMS/Alpha and layered software warranties.

The HP Alpha components represented in CHARON-AXP are designed to operate like their hardware equivalents. In addition to AXE like set of the CPU tests (physical Alpha CPU was sampled with billions of the instruction test cases covering various instruction/operand forms and compared with emulation up to 100% binary equivalence which gives conformance with Alpha architecture) we use HP Alpha hardware diagnostics to verify that a virtual HP Alpha component corresponds to its hardware. To avoid adding unnecessary complexity, the virtual components do not include diagnostic logic that was not used in normal hardware operation. Wherever possible without compromising compatibility, the virtual devices were 'redesigned' to avoid hardware limitations. For example, most virtual HP Alphas support a total emulated memory of up to 32GB, despite restrictions of particular hardware models.

The purpose of HP Alpha virtualization is to replace HP Alpha hardware and any HP Alpha operating system or binary application that runs on HP Alpha hardware. Depending on emulated hardware model of virtual HP Alpha system OpenVMS 6.2-1H3, 7.1, 7.1-1H1, 7.2, 7.2-1, 7.2-2, 7.3, 7.3-1, 7.3-2, 8.2, 8.3, and 8.4 and Tru64 4.0, 5.0, and 5.1 with various patch levels were specifically tested.

Since the performance of a virtual HP Alpha depends on the host system hardware, its components operate at a different speed compared to the equivalent HP Alpha hardware. This is similar to moving a HP Alpha operating system and its applications to a HP Alpha with faster hardware components. The HP Alpha operating system will schedule the various application requests as before and applications will simply complete faster. Virtual real-time components, for instance the HP Alpha system clock, receive the correct timing interrupts and will operate as expected, provided the host system meets the specified minimum system requirements.

Every effort has been made to handle unusual HP Alpha coding sequences correctly. Self-modifying HP Alpha binary code, as is used in Oracle RDB, is part of the verification tests and executes correctly. Note that (generally undesirable) coding techniques like using

NOOPs for software delay loops can give unexpected results as the virtual HP Alpha CPU executes NOOPs very quickly.

7 Installing CHARON-AXP

7.1 Before Installation

The CHARON-AXP distribution kit contains a numbered CHARON-AXP USB license key. The latest versions of the CHARON-AXP SPD, user manual, performance benchmark, and release notes are available online on **www.stromasys.com**. We recommend you to review the release notes before starting the installation of CHARON-AXP. The release notes indicate any changes to the documentation, software or installation procedure since the release of this manual.

Your CHARON license key represents the full value of your CHARON-AXP product. It will not be replaced free of charge if lost or destroyed; we recommend to establish an appropriate security procedure for this high value item.

In the very unlikely case that the CHARON key fails, **DO NOT DISCARD THE KEY** and contact Stromasys immediately for replacement and recovery of the key's internal information. For very high availability requirements, a runtime limited backup key can be purchased.

Before installing CHARON-AXP for Linux:

1. *Make sure that you logged in as user `"root"`. Note that all the CHARON-AXP installations/deinstallations must be performed from `"root"` account, whether some custom accounts can be used for running CHARON-AXP (see the chapter 5.3 for more details).*
2. *Issue a command `"yum install glibc.i686"` to install libraries required by CHARON-AXP.*

7.2 Installing the CHARON-AXP products

1. Extract the content of the distribution TAR.GZ file:

   ```
   tar -xvzf charon-axp-1.1-12904.ZZZZ.tar.gz
   ```

   where:

   `'ZZZZ'` – Target operating system identifier. For Fedora Code FC14 'ZZZZ' the value is 'fc14', for Red Hat Linux v6.0 the value is 'el60'.

   Switch to the installation directory:

   ```
   cd charon-axp-1.1-12904.ZZ
   ```

2. Note the RPM file for each specific CHARON-AXP products:

   `charon-axp-ds10l-1.1-12904.ZZ.x86_64.rpm`

   - CHARON-AXP/DS10L v1.1 Build 12904

   `charon-axp-ds20-1.1-12904.ZZ.x86_64.rpm`

   - CHARON-AXP/DS20 v1.1 Build 12904

charon-axp-es40-1.1-12904.ZZ.x86_64.rpm

- CHARON-AXP/ES40 v1.1 Build 12904

charon-axp-gs80-1.1-12904.ZZ.x86_64.rpm

- CHARON-AXP/GS80 v1.1 Build 12904

charon-axp-gs160-1.1-12904.ZZ.x86_64.rpm

- CHARON-AXP/GS160 v1.1 Build 12904

charon-axp-gs320-1.1-12904.ZZ.x86_64.rpm

- CHARON-AXP/GS320 v1.1 Build 12904

charon-axp-as4100-1.1-12904.ZZ.x86_64.rpm

- CHARON-AXP/4100 v1.1 Build 12904

3. Proceed with installation the required products according to the following format:

```
rpm -i Sentinel_HASP_RedHat_and_SuSe_RPM_Run-time_Installer.rpm charon-axp-base-1.1-12904.ZZ.x86_64.rpm charon-axp-utils-1.1-12904.ZZ.x86_64.rpm <the required product RPM> <the required product RPM>
```

Example of CHARON-AXP/ES40 installation:

```
rpm -i Sentinel_HASP_RedHat_and_SuSe_RPM_Run-time_Installer.rpm charon-axp-base-1.1-12904.81120.fc13.x86_64.rpm charon-axp-utils-1.1-12904.fc13.x86_64.rpm charon-axp-es40-1.1-12904.fc13.x86_64.rpm
```

4. Connect your license key containing you license to the host USB port
5. Re-login to apply the PATH settings. As it was mentioned before, it is recommended to use specific account '*charon*' for running CHARON-AXP.

7.3 CHARON-AXP directories structure

By default CHARON-AXP is installed in the `"/opt/charon"` directory. The following subdirectories are created there:

/bin/axp	- contains all executables
/cfg	- contains templates of configuration files
/utils/axp	- contains set of utilities
/lib/axp	- contains product libraries
/doc	- contains documentation
/log	- contains log files
/disks/axp	- contains disk containers

7.4 Configuring the HP Alpha virtualization layer

After installation you should edit a configuration file for the virtual HP Alpha and install a HP Alpha operating system, for instance OpenVMS/Alpha.

Copy the corresponding configuration templates (residing in the '/opt/charon/cfg/' directory by default) to some files having meaningful name, for example:

```
cp /opt/charon/cfg/ds10l.cfg.template /opt/charon/cfg/my_ds10l.cfg
```

The template configuration files contain examples and explanations of many parameters that tune CHARON-AXP to achieve desired configuration. These files can be used as starting point for creating user specific configuration files.

The configuration procedure is described in details in the next chapters.

7.5 Running CHARON-AXP

For running CHARON-AXP AlphaServer models please use the following symbolic link names:

Link name	Emulator to run
ds10l	AlphaServer DS10L
ds20	AlphaServer DS10L
es40	AlphaServer ES40
gs80	AlphaServer GS80
gs160	AlphaServer GS160
gs320	AlphaServer GS320
as400	AlphaServer 400
as800	AlphaServer 800
as1000	AlphaServer 1000
as1000a	AlphaServer 1000a
as1200	AlphaServer 1200
as2000	AlphaServer 2000
as2100	AlphaServer 2100
as4000	AlphaServer 4000
as4100	AlphaServer 4100

Run corresponding CHARON-AXP AlphaServer model with the created configuration file, for example:

```
ds10l /opt/charon/cfg/my_ds10l.cfg
```

To exit from the SRM console of the running emulator use "*power off*" command or press ‘F6’ button.

7.6 Uninstalling CHARON-AXP

To uninstall some CHARON-AXP particular product login as "*root*" user and issue the following command:

```
rpm -e <the product RPM name>
```

For example for uninstallation of the AlphaServer ES40 product use the following command:

```
rpm -e charon-axp-es40-1.1-12904
```

If it is required to uninstall all the products completely (including all the compartment components and drivers) use the following command:

```
rpm -e aksusbd-1.15-1.i368 charon-axp-base-1.1-12904 charon-axp-utils-1.1-12904 <the product RPM name> <the product RPM name>
```

8 Configuring the virtual HP Alpha

8.1 The HP Alpha system architecture

In hardware HP Alpha system, the CPU, memory, peripheral controllers and adapters are connected through the central system buses.

CHARON-AXP implements these central system buses, the HP Alpha CPU(s), memory, disk/tape controllers and the Ethernet components. When CHARON-AXP starts, it follows a configuration script and assembles a virtual HP Alpha system by combining models of the buses, the HP Alpha CPU, memory and controllers into a working unit and loading this into the host system.

The virtual peripheral devices are mapped in the configuration script to a device on the host system. For instance, a virtual DE500BA Ethernet adapter is associated with a dedicated physical Ethernet controller in the host system, thus connecting the virtual HP Alpha to a physical Ethernet network.

When the configuration script is fully executed, CHARON-AXP has created a complete virtual HP Alpha system. It then hands over control to the HP Alpha CPU, which will boot the HP Alpha system software in the same manner as would happen on HP Alpha hardware.

The configuration script consists of one or more text files with a **.cfg** extension. To facilitate structuring of large configurations, a part of the configuration can be stored in a separate file. Such file is incorporated in the main script with the **`include`** command.

Follow the steps below for a quick start with a custom configuration:

1. Choose an emulated Alpha model you would like to run, and copy the relevant default configuration file template to your private configuration file (for example: `my_es40.cfg`);
2. Write a comment at the top of the `my_es40.cfg` to define the purpose for this configuration;
3. Define the amount of RAM you require (for example: `"set ram size=1024"`). By default the memory amount is set to different values depending on the HP Alpha models.
4. Define the virtual operator console with `"load operator_console OPA0"` command.
5. Define the storage units (for example: `"set PKA container[0]="file-name.vdisk"`). This first unit will appear in CHARON-AXP as DKA0 in CHARON-AXP SRM console;

6. Define the Ethernet device connection in the configuration file, for example:

   ```
   load DE500BA/dec21x4x EWA interface=EWA0

   load packet_port/chnetwrk EWA0 interface="eth0"
   ```

   It is necessary to ensure that this interface is dedicated completely to CHARON-AXP. See the following chapters for details on how to do that.

7. Now you can run the configured system.

8.2 The configuration command syntax

This section describes each of the configuration commands and syntax, grouped according to the type of component.

There are two types of configuration commands:

The `load` command instructs CHARON-AXP to add a component to a system bus.

The `set` command defines the characteristics of a loaded component.

To be able to load and manipulate more than one copy of a particular component a logical name is assigned to each loaded component as the following example shows:

```
load "component A" NAME1
load "component B" NAME2
```

NAME1 and NAME2 are freely chosen names *that are only relevant within the configuration file.* These names have no meaning and will never show up in the operating system running on the virtual Alpha.

Using their logical names the two identical components (for instance two DE500BA Ethernet adapters) can now be given individual parameters (for instance the IDs of the host adapters they should use). The example configuration files in the Appendix A show how the logical names are used.

The parameters used with the **set** commands are typically assigned to a value, which can be **true/false**, a **number** or a **text string**. Numbers can be expressed in different formats, as it can be more convenient to use octal or hexadecimal formats:

For octal use a number starting with 0; use the symbols 0 – 7. Example: 07665

For decimal use a number starting with 1 - 9. Example: 12345

For hexadecimal use a number starting with 0x; 0 - 9 and a – f. Example: 0x1234abc

The **set** commands are listed separately in this manual. Alternatively, all **load** commands can contain additional parameters to get a more compact configuration file. For instance:

```
load "component A" NAME1 <parameter>="abcd"
```

Is equivalent to:

```
load "component A" NAME1
set NAME1 <parameter>="abcd"
```

8.3 *The virtual AXP models specifics*

All the emulators included to the CHARON-AXP have certain specific in terms of the PCI bus configuration and the peripherals that can be connected. This specific reflects original HP Alpha system hardware configurations and introduced in virtual HP Alpha system to provide better compatibility with original HP Alpha operating systems (presumably old versions of HP Tru64 UNIX Operating System).

8.3.1 AlphaServer 400 (1 PCI bus, 3 PCI slots)

In addition to 3 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 5 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	6	0	11	NCR 53C810 PCI SCSI Adapter
-	0	7	0	-	Intel i82378 PCI ISA Bridge (SATURN)
0	0	11	0	10	Option
1	0	12	0	15	Option
2	0	13	0	9	Option

So far, the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

8.3.2 AlphaServer 800 (DECchip 21172, 4 PCI slots)

In addition to 4 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 7 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	5	0	0	QLOGIC ISP1020 PCI SCSI Adapter
-	0	6	0	0	S3 Trio32/64 Display Adapter
-	0	7	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
0	0	11	0	10	option
1	0	12	0	15	option

Slot	PCI	Device	Function	IRQ	
2	0	13	0	9	option
3	0	14	0	7	option

The IRQ stands for input line of ASIC interrupt controllers. It has nothing to do with "EISA" style interrupts.

So far, the CHARON-AXP emulators do not emulate S3 Trio32/64 Display Adapter. So position of the device 6, function 0 on the PCI 0 remains empty.

8.3.3 AlphaServer 1000 (DECchip 21072, 3 PCI slots)

In addition to 3 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 5 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	6	0	12	NCR 53C810 PCI SCSI Adapter
-	0	7	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
0	0	11	0	0	option
1	0	12	0	4	option
2	0	13	0	8	option

The IRQ stands for input line of ASIC interrupt controllers. It has nothing to do with "EISA" style interrupts.

So far, the CHARON-AXP emulators do not emulate NCR 53C810 PCI SCSI adapter. Instead, emulation of QLOGIC ISP1040B PCI SCSI adapter is used.

8.3.4 AlphaServer 1000A (DECchip 21072, 7 PCI slots)

In addition to 7 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 10 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					

Slot	PCI	Device	Function	IRQ	
-	0	7	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
-	0	8	0	-	DECchip 21050 PCI-to-PCI Bridge
0	0	11	0	1	option
1	0	12	0	3	option
2	0	13	0	5	option
pci_1					
-	1	0	0	0	NCR 53C810 PCI SCSI Adapter
3	1	1	0	7	option
4	1	2	0	9	option
5	1	3	0	11	option
6	1	4	0	13	option

The IRQ stands for input line of ASIC interrupt controllers. It has nothing to do with "EISA" style interrupts.

So far, the CHARON-AXP emulators do not emulate NCR 53C810 PCI SCSI adapter. Instead, emulation of QLOGIC ISP1040B PCI SCSI adapter is used.

8.3.5 AlphaServer 1200 (1 IOD, 6 PCI slots)

In addition to 6 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 8 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					
-	1	1	0	4	NCR 53C810 PCI SCSI Adapter
0	1	2	0	8	option
1	1	3	0	12	option
2	1	3	0	16	option
pci_0					
-	0	0	0	-	Intel i82375 PCI EISA Bridge (MERCURY)

Slot	PCI	Device	Function	IRQ	
3	0	1	0	8	option
4	0	2	0	12	option
5	0	3	0	16	option

So far, the CHARON-AXP emulators do not emulate NCR 53C810 PCI SCSI adapter. Instead, emulation of QLOGIC ISP1040B PCI SCSI adapter is used.

8.3.6 AlphaServer 2000 (1 PCI bus, 3 PCI slots)

In addition to 3 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 6 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	0	0	2	DEC TULIP PCI Ethernet adapter
-	0	1	0	1	NCR 53C810 PCI SCSI Adapter
-	0	2	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
0	0	6	0	0	option
1	0	7	0	4	option
2	0	8	0	5	option

So far, the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

8.3.7 AlphaServer 2100 (1 PCI bus, 3 PCI slots)

In addition to 3 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 6 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	0	0	2	DEC TULIP PCI Ethernet adapter
-	0	1	0	1	NCR 53C810 PCI SCSI Adapter

Slot	PCI	Device	Function	IRQ	
-	0	2	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
0	0	6	0	0	option
1	0	7	0	4	option
2	0	8	0	5	option

So far, the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

8.3.8 AlphaServer 4000 (4 PCI busses, 16 PCI slots)

In addition to 16 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 18 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					
-	1	1	0	4	NCR 53C810 PCI SCSI Adapter
-	1	2	0	8	option
-	1	3	0	12	option
-	1	4	0	16	option
-	1	5	0	20	option
pci_0					
-	0	1	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
-	0	2	0	8	option
-	0	3	0	12	option
-	0	4	0	16	option
-	0	5	0	20	option
pci_3					
-	3	2	0	8	option
-	3	3	0	12	option

Slot	PCI	Device	Function	IRQ	
-	3	4	0	16	option
-	3	5	0	20	option
pci_2					
-	2	2	0	8	option
-	2	3	0	12	option
-	2	4	0	16	option
-	2	5	0	20	option

So far, the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

8.3.9 AlphaServer 4100 (2 PCI busses, 8 PCI slots)

In addition to 8 PCI vacant slots there are 2 PCI positions occupied by on-board devices. All 10 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					
-	1	1	0	4	NCR 53C810 PCI SCSI Adapter
-	1	2	0	8	option
-	1	3	0	12	option
-	1	4	0	16	option
-	1	5	0	20	option
pci_0					
-	0	1	0	-	Intel i82375 PCI EISA Bridge (MERCURY)
-	0	2	0	8	option
-	0	3	0	12	option
-	0	4	0	16	option
-	0	5	0	20	option

So far, the CHARON-AXP emulators do not support virtual NCR 53C810 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.

8.3.10 AlphaServer DS10L (1 PCI bus, 4 PCI slot)

In addition to 4 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 9 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	7	0	-	ALi M1543C PCI ISA bridge
-	0	9	0	29	DECchip 21143 PCI Ethernet Adapter
-	0	11	0	30	DECchip 21143 PCI Ethernet Adapter
-	0	13	0	-	ALi M1543C PCI IDE/ATAPI controller
1	0	14	0	35	Option
2	0	15	0	39	Option
3	0	16	0	43	Option
4	0	17	0	47	Option
-	0	19	0	-	ALi M1543C PCI USB adapter

So far, the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. The position of the device 19, function 0 on the PCI 0 remains empty.

8.3.11 AlphaServer DS20 (2 PCI busses, 6 PCI slots)

In addition to 6 PCI vacant slots there are 5 PCI positions occupied by on-board devices. All 11 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					
4	1	7	0	47	option
5	1	8	0	43	option
6	1	9	0	39	option

Slot	PCI	Device	Function	IRQ	
pci_0					
-	0	5	0	-	ALi M1543C PCI ISA bridge
-	0	6	0	19	Adaptec AIC-7895 (channel 0)
-	0	6	1	18	Adaptec AIC-7895 (channel 1)
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter
1	0	7	0	31	option
2	0	8	0	27	option
3	0	9	0	23	option

Unless SCSI option is plugged into PCI slot 4, 5, or 6, the onboard SCSI controllers appear as PKA (pka7.0.0.6.0) and PKB (pkb7.0.0.106.0) respectively.
So far, the CHARON-AXP emulators do not support virtual Adaptec AIC-7895 PCI SCSI adapter. Instead, virtual QLOGIC ISP1040B PCI SCSI adapter is used.
So far, the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. The position of the device 19, function 0 on the PCI 0 remains empty.

8.3.12 AlphaServer ES40 (2 PCI busses, 10 PCI slots)

In addition to 10 PCI vacant slots there are 3 PCI positions occupied by on-board devices. All 13 PCI positions are listed in the following table in the order in which Alpha SRM console enumerates them.

Slot	PCI	Device	Function	IRQ	
pci_1					
5	1	1	0	24	Option
6	1	2	0	28	Option
7	1	3	0	32	Option
8	1	4	0	36	Option
9	1	5	0	40	Option
10	1	6	0	44	Option
pci_0					

Slot	PCI	Device	Function	IRQ	
1	0	1	0	8	Option
2	0	2	0	12	Option
3	0	3	0	16	Option
4	0	4	0	20	Option
-	0	5	0	-	ALi M1543C PCI ISA bridge
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter

So, far the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. The position of the device 19, function 0 on the PCI 0 remains empty.

8.3.13 AlphaServer GS80 (2 QBBs, 8 PCI busses, 27 PCI slots)

Slot	PCI	Device	Function	IRQ	
qbb_0_pca_0_pci_0					
0/1	0	1	0	36	QLOGIC ISP1040B PCI SCSI Adapter
2	0	2	0	40	option
3	0	3	0	44	option
-	0	7	0	-	ALi M1543C PCI ISA bridge
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter
qbb_0_pca_0_pci_1					
4	1	4	0	48	option
5	1	5	0	52	option
6	1	6	0	56	option
7	1	7	0	60	option
qbb_0_pca_1_pci_0					
0/1	2	0	0	32	option

Slot	PCI	Device	Function	IRQ	
2	2	2	0	40	option
3	2	3	0	44	option
qbb_0_pca_1_pci_1					
4	3	4	0	48	option
5	3	5	0	52	option
6	3	6	0	56	option
7	3	7	0	60	option
qbb_1_pca_0_pci_0					
0/1	8	0	0	32	option
2	8	2	0	40	option
3	8	3	0	44	option
qbb_1_pca_0_pci_1					
4	9	4	0	48	option
5	9	5	0	52	option
6	9	6	0	56	option
7	9	7	0	60	option
qbb_1_pca_1_pci_0					
0/1	10	0	0	32	option
2	10	2	0	40	option
3	10	3	0	44	option
qbb_1_pca_1_pci_1					
4	11	4	0	48	option
5	11	5	0	52	option
6	11	6	0	56	option
7	11	7	0	60	option

PCA 2 and 3 on each QBB are not populated.

So far, the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter.

The position of the device 19, function 0 on the PCI 0 remains empty.
Total number of PCI devices configured through CFG file may not exceed 27.

8.3.14 AlphaServer GS160 (4 QBBs, 16 PCI busses, 55 PCI slots)

Slot	PCI	Device	Function	IRQ	
qbb_0_pca_0_pci_0					
0/1	0	1	0	36	QLOGIC ISP1040B PCI SCSI Adapter
2	0	2	0	40	option
3	0	3	0	44	option
-	0	7	0	-	ALi M1543C PCI ISA bridge
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter
qbb_0_pca_0_pci_1					
4	1	4	0	48	option
5	1	5	0	52	option
6	1	6	0	56	option
7	1	7	0	60	option
qbb_0_pca_1_pci_0					
0/1	2	0	0	32	option
2	2	2	0	40	option
3	2	3	0	44	option
qbb_0_pca_1_pci_1					
4	3	4	0	48	option
5	3	5	0	52	option
6	3	6	0	56	option
7	3	7	0	60	option
qbb_1_pca_0_pci_0					
0/1	8	0	0	32	option

Slot	PCI	Device	Function	IRQ	
2	8	2	0	40	option
3	8	3	0	44	option
qbb_1_pca_0_pci_1					
4	9	4	0	48	option
5	9	5	0	52	option
6	9	6	0	56	option
7	9	7	0	60	option
qbb_1_pca_1_pci_0					
0/1	10	0	0	32	option
2	10	2	0	40	option
3	10	3	0	44	option
qbb_1_pca_1_pci_1					
4	11	4	0	48	option
5	11	5	0	52	option
6	11	6	0	56	option
7	11	7	0	60	option
qbb_2_pca_0_pci_0					
0/1	16	0	0	32	option
2	16	2	0	40	option
3	16	3	0	44	option
qbb_2_pca_0_pci_1					
4	17	4	0	48	option
5	17	5	0	52	option
6	17	6	0	56	option
7	17	7	0	60	option
qbb_2_pca_1_pci_0					

Slot	PCI	Device	Function	IRQ	
0/1	18	0	0	32	option
2	18	2	0	40	option
3	18	3	0	44	option
qbb_2_pca_1_pci_1					
4	19	4	0	48	option
5	19	5	0	52	option
6	19	6	0	56	option
7	19	7	0	60	option
qbb_3_pca_0_pci_0					
0/1	24	0	0	32	option
2	24	2	0	40	option
3	24	3	0	44	option
qbb_3_pca_0_pci_1					
4	25	4	0	48	option
5	25	5	0	52	option
6	25	6	0	56	option
7	25	7	0	60	option
qbb_3_pca_1_pci_0					
0/1	26	0	0	32	option
2	26	2	0	40	option
3	26	3	0	44	option
qbb_3_pca_1_pci_1					
4	27	4	0	48	option
5	27	5	0	52	option
6	27	6	0	56	option
7	27	7	0	60	option

PCA 2 and 3 on each QBB are not populated.

So far, the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter. The position of the device 19, function 0 on the PCI 0 remains empty.

Total number of PCI devices configured through CFG file may not exceed 27.

8.3.15 AlphaServer GS320 (8 QBBs, 32 PCI busses, 111 PCI slots)

Slot	PCI	Device	Function	IRQ	
qbb_0_pca_0_pci_0					
0/1	0	1	0	36	QLOGIC ISP1040B PCI SCSI Adapter
2	0	2	0	40	option
3	0	3	0	44	option
-	0	7	0	-	ALi M1543C PCI ISA bridge
-	0	15	0	-	ALi M1543C PCI IDE/ATAPI controller
-	0	19	0	-	ALi M1543C PCI USB adapter
qbb_0_pca_0_pci_1					
4	1	4	0	48	option
5	1	5	0	52	option
6	1	6	0	56	option
7	1	7	0	60	option
qbb_0_pca_1_pci_0					
0/1	2	0	0	32	option
2	2	2	0	40	option
3	2	3	0	44	option
qbb_0_pca_1_pci_1					
4	3	4	0	48	option
5	3	5	0	52	option
6	3	6	0	56	option
7	3	7	0	60	option
qbb_1_pca_0_pci_0					

Slot	PCI	Device	Function	IRQ	
0/1	8	0	0	32	option
2	8	2	0	40	option
3	8	3	0	44	option
qbb_1_pca_0_pci_1					
4	9	4	0	48	option
5	9	5	0	52	option
6	9	6	0	56	option
7	9	7	0	60	option
qbb_1_pca_1_pci_0					
0/1	10	0	0	32	option
2	10	2	0	40	option
3	10	3	0	44	option
qbb_1_pca_1_pci_1					
4	11	4	0	48	option
5	11	5	0	52	option
6	11	6	0	56	option
7	11	7	0	60	option
qbb_2_pca_0_pci_0					
0/1	16	0	0	32	option
2	16	2	0	40	option
3	16	3	0	44	option
qbb_2_pca_0_pci_1					
4	17	4	0	48	option
5	17	5	0	52	option
6	17	6	0	56	option
7	17	7	0	60	option

Slot	PCI	Device	Function	IRQ	
qbb_2_pca_1_pci_0					
0/1	18	0	0	32	option
2	18	2	0	40	option
3	18	3	0	44	option
qbb_2_pca_1_pci_1					
4	19	4	0	48	option
5	19	5	0	52	option
6	19	6	0	56	option
7	19	7	0	60	option
qbb_3_pca_0_pci_0					
0/1	24	0	0	32	option
2	24	2	0	40	option
3	24	3	0	44	option
qbb_3_pca_0_pci_1					
4	25	4	0	48	option
5	25	5	0	52	option
6	25	6	0	56	option
7	25	7	0	60	option
qbb_3_pca_1_pci_0					
0/1	26	0	0	32	option
2	26	2	0	40	option
3	26	3	0	44	option
qbb_3_pca_1_pci_1					
4	27	4	0	48	option
5	27	5	0	52	option
6	27	6	0	56	option

Slot	PCI	Device	Function	IRQ	
7	27	7	0	60	option
qbb_4_pca_0_pci_0					
0/1	24	0	0	32	option
2	24	2	0	40	option
3	24	3	0	44	option
qbb_4_pca_0_pci_1					
4	25	4	0	48	option
5	25	5	0	52	option
6	25	6	0	56	option
7	25	7	0	60	option
qbb_4_pca_1_pci_0					
0/1	26	0	0	32	option
2	26	2	0	40	option
3	26	3	0	44	option
qbb_4_pca_1_pci_1					
4	27	4	0	48	option
5	27	5	0	52	option
6	27	6	0	56	option
7	27	7	0	60	option
qbb_5_pca_0_pci_0					
0/1	24	0	0	32	option
2	24	2	0	40	option
3	24	3	0	44	option
qbb_5_pca_0_pci_1					
4	25	4	0	48	option
5	25	5	0	52	option

Slot	PCI	Device	Function	IRQ	
6	25	6	0	56	option
7	25	7	0	60	option
qbb_5_pca_1_pci_0					
0/1	26	0	0	32	option
2	26	2	0	40	option
3	26	3	0	44	option
qbb_5_pca_1_pci_1					
4	27	4	0	48	option
5	27	5	0	52	option
6	27	6	0	56	option
7	27	7	0	60	option
qbb_6_pca_0_pci_0					
0/1	24	0	0	32	option
2	24	2	0	40	option
3	24	3	0	44	option
qbb_6_pca_0_pci_1					
4	25	4	0	48	option
5	25	5	0	52	option
6	25	6	0	56	option
7	25	7	0	60	option
qbb_6_pca_1_pci_0					
0/1	26	0	0	32	option
2	26	2	0	40	option
3	26	3	0	44	option
qbb_6_pca_1_pci_1					
4	27	4	0	48	option

Slot	PCI	Device	Function	IRQ	
5	27	5	0	52	option
6	27	6	0	56	option
7	27	7	0	60	option
qbb_7_pca_0_pci_0					
0/1	24	0	0	32	option
2	24	2	0	40	option
3	24	3	0	44	option
qbb_7_pca_0_pci_1					
4	25	4	0	48	option
5	25	5	0	52	option
6	25	6	0	56	option
7	25	7	0	60	option
qbb_7_pca_1_pci_0					
0/1	26	0	0	32	option
2	26	2	0	40	option
3	26	3	0	44	option
qbb_7_pca_1_pci_1					
4	27	4	0	48	option
5	27	5	0	52	option
6	27	6	0	56	option
7	27	7	0	60	option

PCA 2 and 3 on each QBB are not populated in emulator.
So far, the CHARON-AXP emulators do not support virtual ALi M1543C PCI USB adapter.
The position of the device 19, function 0 on the PCI 0 remains empty.
Total number of PCI devices configured through CFG file may not exceed 27.

8.4 Multi instance support

Several instances of CHARON-AXP could run simultaneously on the same host. Emulator types and number of instances allowed to run simultaneously is encoded into CHARON-AXP license key.

In order to run several instances simultaneously, please notice:

1. The host system should have enough CPU cores and memory to cover the requirements of all the instances at the same time.

   Each virtual HP Alpha CPU occupies one host CPU, so the total number of CPUs should be greater than a sum of all the emulated CPUs. Note that some CPUs needs to be used for I/O processing and at least one CPU for the operating system housekeeping. Thus the total amount of the host CPUs depends on the number of the CPUs needed for I/O. The general recommendation is to leave at least 1/3 of the CPUs available to an instance for the instance I/O (for example, 4 CPUs out of 12), but depending on data flow this number might need a change for each instance separately.

   The minimal host memory is calculated as a sum of emulated memory of each CHARON-AXP instance plus at least 2 GB of additional memory.

2. Each instance should have its own configuration and log files, rom, nvram and toy containers. Configuration file of each CHARON-AXP instance should exactly specify the following:

   a) The number of CPUs to emulate (`"n_of_cpus"`). By default this parameter is equal to the number of the CPUs the particular HP Alpha model supports. But this number can be reduced by changing the parameter or by the license restrictions

   b) The number of CPUs chosen for I/O operations (`"n_of_io_cpus"`). By default this parameter is equal to 1/3 of the CPUs available for certain emulator (round by 1) and cannot be less than one. However it is possible to dedicate a chosen number of CPUs for I/O processing in case of intensive or, in opposite case, very low data flow.

   c) Number of the CPUs the instance allocates. By default CHARON-AXP instance grabs as many CPUs as possible. To balance the number of host CPUs between different instances a special parameter `"set session affinity"` is provided. This parameter specifies what CPUs in particular each instance can allocate.

3. Once the configuration files are updated for each particular instance CHARON-AXP, it is recommended to test those configurations separately.

   <u>Example:</u>

   ```
   ➢ gs160 gs160_first.cfg
   ```

- `gs160 gs160_second.cfg`

See the next chapter for detailed description of the `set session n_of_cpus`, `set session n_of_io_cpus`, and `set session affinity` parameters.

8.5 General configuration parameters

A few `set` parameters have no `load` counterpart; they set general parameters influencing startup and logging:

Set parameters for session	Type	Value
hw_model	Text string	The virtual HP Alpha system hardware model for which the configuration file is created. Using a default configuration template for a particular model as a starting point for a custom configuration would ensure that the parameter is set correctly.
log	Text string	A string specifying the file name to store the log of the session.
log_method	Text string	"overwrite" (default) or "append". Determines if previous log information is maintained. Note that this parameter must be specified only in addition to the "log" parameter on the same line with it. For example: **set session log="log.txt" log_method="append"**
log_mode	Text string	"shared" (default) or "private". Determines if the LOG file can be shared with external viewer. Private LOG creates less overhead for writing, while shared LOG file is more convenient for monitoring. Note that this parameter must be specified only in addition to the "log" parameter on the same line with it. For example: **set session log="log.txt" log_mode="private"**
log_show_messages	Text string	Defines the message types that should be shown. The parameter is a string of comma delimited words: "all", "info", "warning" and "error" which defines which message types should be logged. The default value is "all" message types.
log_repeat_filter	Text string	Specifies if repeated messages should be filtered or not. Possible values are "on" (default) and "off". If the value is "on", immediately following messages with the same identifier and system error code are not listed in the log, but they are counted. When a different log message is generated, the repeat count of the earlier log message is reported with "The previous message has been repeated N times.", and the counter is cleared.
log_locale	Text string	Sets the language of message database. So far the following values are supported:

Set parameters for session	Type	Value
		• "Dutch", • "English", • "Swedish", • "Spanish", • "Chinese-Simplified". By default it is set to "English". If specified an unsupported value, "English" is used. For example: **set session log_locale="Dutch"**
affinity	Text string	Overrides initial process's affinity mask provided by host operating system. Once specified it allows binding the running instance of emulator to particular host CPUs. Might be used for soft partitioning host CPU resources and for isolating host CPUs for other applications. By default the emulator instance allocates as many host CPUs as possible. The "affinity" overrides that and allows explicit specification on which host CPU the instance shall run. Host CPUs are enumerated as comma separated list of host system assigned CPU numbers, for example: **set session affinity="0, 2, 4, 6"**
n_of_io_cpus	Numeric	Tells how many host CPUs (of those specified by "affinity" parameter, if any) the emulator shall use for I/O handling. By default the emulator instance reserves one third of available host CPUs for I/O processing (round down, at least one). The "n_of_io_cpus" overrides that by specifying number of I/O host CPUs explicitly, for example: **set session n_of_io_cpus=2**
n_of_cpus	Numeric	Limits number of emulated CPUs. For example: **set session n_of_cpus=3** Maximum number of CPUs enabled by CHARON-AXP is specified by the license key, but cannot exceed the original hardware restrictions: • AlphaServer_AS400 – 1 CPU • AlphaServer_AS800 – 1 CPU • AlphaServer_AS1000 – 1 CPU • AlphaServer_AS1000A – 1 CPU • AlphaServer_AS1200 – 2 CPU • AlphaServer_AS2000 – 2 CPUs • AlphaServer_AS2100 – 4 CPUs • AlphaServer_AS4000 – 2 CPUs • AlphaServer_AS4100 – 4 CPUs • AlphaServer_DS10L – 1 CPUs • AlphaServer_DS20 – 2 CPUs

Set parameters for session	Type	Value
		- AlphaServer_ES40 – 4 CPUs - AlphaServer_GS80 – 8 CPUs - AlphaServer_GS160 – 16 CPUs - AlphaServer_GS320 – 32 CPUs At startup emulator adjusts the number of emulated CPUs accordingly to the number of available host CPU cores (enabled by "affinity" if any). This option overrides automatic adjustment. Note that in any case emulator reserves at least one host CPU core for I/O management, so that given N host CPU cores emulator supports up-to N-1 emulated CPUs. Please notice that maximum number of the emulated CPUs could be also restricted through the CHARON license.

Example:

```
set session hw_model="AlphaServer_ES40"
```

This command specifies HP Alpha Server model the configuration file is designed for. It must be the first command in a configuration file. Various CHARON-AXP products create specific virtual HP Alpha CPU models and have different configuration commands. This command helps to detect errors and prevents execution in case an incorrect virtual HP Alpha model is started. If the `set session hw_model="...."` statement is not found, the configuration file is ignored, and the virtual HP Alpha will not be activated.

Example:

```
set session log="clipper.log" log_method="append"
```

Creates a log file in the directory where CHARON-AXP starts. Specify the full path to locate the log file elsewhere. The specified log file is created or overwritten at each start depending on the `log_method` parameter. The `log_method` parameter must be specified on the same line with the `"log"` parameter.

8.6 The HP ALPHA and its console interface

The virtual HP Alpha system supports one serial console port, which in CHARON-AXP is identified with the logical name **OPA0**. To use the **OPA0** a physical or virtual serial line connection must be loaded in the configuration file.

Emulated Alpha models AS400 and DS10L also have a second serial line, **TTA0**.

Terminals can also be connected to CHARON-AXP via TCP/IP or LAT terminal servers.

Load parameter	Function
physical_serial_line	This command associates a TTY port in the Linux host system with the OPA0 console port. The TTY port can be a physical port part of the host

Load parameter	Function
	system hardware or a logical TTY port as created by, for example, an Ethernet serial port device.
virtual_serial_line	This command associates a network connection in the Linux host system with the OPA0 console port.
operator_console	This command associates the current TTY console CHARON-AXP runs in with the OPA0 console port.

set physical_serial_line parameter	Type	Value
Line	Text string	"/dev/ttyY " A defined TTY port on the Linux host system. See explanation below for more details.
break_on	Numeric	Specifies which byte sequences received over physical serial line shall trigger HALT command with switching to CHARON-AXP SRM console. Specify th following values: "**Crtl-P**", or "**none**" to disable triggering HALT condition. For example: **break_on=" Crtl-P "** The default value is " **Crtl-P** ".
baud	Numeric	Forces the baud rate of the corresponding TTY port to the specified value. Variety of supported values depends on underlying physical communication resource (TTY port that is). The most widely used values are: 300, 1200, 9600, 19200, 38400. For example: **load physical_serial_line OPA0 baud=38400**

Example:

```
load physical_serial_line OPA0
set OPA0 line="/dev/ttyS1"
```

or in a more compact form:

```
load physical_serial_line OPA0 line="/dev/ttyS1"
```

set virtual_serial_line parameter	Type	Value
host	Text string	The remote host’s IP address or host name and optionally remote TCP/IP port number for the virtual serial line to connect to. If omitted, the virtual serial line does not initiate connection to remote host while still listening for incoming connection requests. Specify the value in the following form: **host="<host-name>[:<port-no>]"** If the <port-no> is not specifed the virtual serial line uses TCP/IP port number specified by the "**port**" parameter (see below).
port	Numeric	TCP/IP port number for the virtual serial line. The virtual serial line always listens on this port for incoming connection requests.
stop_on	Text string	Specifies which byte sequences received over virtual serial line shall trigger STOP condition. The STOP condition causes CHARON-AXP to terminate. Specify value as a comma separated combination of the following: "**F6**", or as "**none**" to disable triggering STOP condition. For example: **stop_on="F6"** The default value is "**none**". Set to "**F6**" to trigger the STOP condition upon reception of the sequence "<ESC>[17~". Terminal may send these sequences when pressing the F6 button.
break_on	Text string	Specifies which byte sequences received over physical serial line shall trigger HALT command with switching to CHARON-AXP SRM console. Specify value as a comma separated combination of the following: "**Crtl-P**", "**F5**", "**Break**" or "**none**" to disable triggering HALT condition. For example: **break_on=" Crtl-P "** The default value is " **F5** " and " **Break** ".

Example defining a local terminal session as the serial console terminal:

```
load virtual_serial_line OPA0
set OPA0 port=10003 stop_on="F6"
```

The following example defines the HP Alpha console as an application on node 192.168.1.1, which should be listening on port 10000 when CHARON-AXP starts:

```
load virtual_serial_line OPA0 host="192.168.1.1" port=10000
```

set ***operator_console*** **parameter**	**Type**	**Value**
stop_on, break_on	Text string	Those parameters are hardcoded to the following values that cannot be changed: **stop_on=" F6"** **break_on="Ctrl-P,F5"**

Example defining a local session as the serial console terminal:

```
load operator_console OPA0 stop_on="Ctrl-P,F5"
```

Note that the `"ttyY"` notation can have different form depending on the nature of the device used:

1. Linux virtual `tty` (switchable by `alt+F1-atl+F12` on a text console) – are represented as `"/dev/ttyN"` where N is from 0 to 11. Those `tty` devices must be free from the Linux `getty/mgetty` and similar programs (specified in `"/etc/inittab"`)
2. Onboard serial lines are represented as `"/dev/ttySN"` where N is a number. For example `"/dev/ttyS1"`
3. Proprietary (depending on a driver) devices are represented as `"/dev/ttyXXX"` where XXX is a complex letter/number notation. For example `"/dev/ttyR01"` is a first port of the MOXA card and the `"/dev/ttyaa"` stands for the first port of the DIGI card.

8.7 Specifying HP Alpha memory

The memory subsystem is permanently loaded and has the logical name **`ram`**. The effective amount of memory is determined in steps, starting with the `"set ram size"` statement in the configuration file:

1. If no `set ram` statement is found, the memory size is set to 512MB, except for the AlphaServer_GS320 for which it is set to 1024MB.
2. Where applicable, the memory is capped to the maximum as defined in the CHARON-AXP license key.

The following table summarizes parameters of emulated RAM for various hardware models of virtual HP Alpha system:

Hardware model	**RAM size (in MB)**			
	Min	**Max**	**Default**	**Increment**
AlphaServer 400	256	1024	512	256
AlphaServer 800	256	8192	512	256

AlphaServer 1000	256	1024	512	256
AlphaServer 1000A	256	1024	512	256
AlphaServer 1200	256	32768	512	256
AlphaServer 2000	256	2048	512	256
AlphaServer 2100	256	2048	512	256
AlphaServer 4000	256	32768	512	256
AlphaServer 4100	256	32768	512	256
AlphaServer DS10L	256	32768	512	256
AlphaServer DS20	256	32768	512	256
AlphaServer ES40	256	32768	512	256
AlphaServer GS80	256	65526	512	256
AlphaServer GS160	512	131072	512	512
AlphaServer GS320	1024	262144	1024	1024

Examples (the size must be specified in MB):

```
set ram size = 512
```

Creates 512 MB

```
set ram size = 4096
```

Creates 4 GB HP Alpha memory

In addition, CHARON-AXP will generate an error message in the log file and reduce its effective memory size further if the Linux host system cannot allocate enough memory to map the calculated HP Alpha memory size.

Please notice that the maximum RAM size is also depends on the purchased product configuration and could be restricted by CHARON-AXP license.

8.8 Virtual HP Alpha system time and date

The virtual HP Alpha system maintains its time and date via TOY (time-of-year) component. In order to preserve time and date while virtual HP Alpha system is not running the TOY component uses small binary file on the Linux host system. Name of the file is specified by "container" option of the TOY component.

set parameter	Type	Value
container	Text string	Specifies the name of file in which the virtual HP Alpha system preserves its time and date during "offline" period. By default it is left unspecified.

For example:

```
set TOY container="my_virtual_alpha.dat"
```

The virtual HP Alpha system may have its time and date different from system time and date of the Linux host system, but relies on correctness of the host's system time and date to calculate duration of "offline" period (i.e. while virtual HP Alpha system is not running).

8.9 Virtual HP Alpha SRM console environment

The virtual HP Alpha system implements minimal Alpha SRM console environment according to Alpha Architecture Reference Manual. The virtual HP Alpha SRM console environment is part of virtual HP Alpha ROM (which also carries virtual HP Alpha firmware). In order to preserve console environment settings (such as, default boot device, boot OS flags, boot file name, etc …) while virtual HP Alpha system is not running the ROM component uses nearly 2MB binary file on the Linux host. Name of the file is specified by "container" option of the ROM component.

set parameter	Type	Value
Container	Text string	Specifies the name of file in which the virtual HP Alpha system preserves its firmware image and console environment during "offline" period. By default it is left unspecified.

For example:

```
set ROM container="my_virtual_alpha.bin"
```

The same file also carries copy of virtual HP Alpha firmware. Each new version of the CHARON-AXP software updates the firmware preserved in the file thus clearing console environment variables.

8.10 Virtual HP Alpha interval timer

The CHARON-AXP virtualization layer provides interval timer interrupts to virtual Alpha CPU(s) at frequency 100Hz (100 interrupts a second). This is default behavior which may be changed through "clock_period" configuration parameter of virtual ISA or EISA bus, depending on emulated hardware model of virtual HP Alpha system. Value of the parameter is interval timer period in microseconds. By default it is set to 10000. By changing it to 1000 frequency of virtual interval timer interrupts may be increased to 1000Hz (1000 interrupts a second).

set parameter	Type	Value
clock_period	Numeric	Specifies period of interval timer, in microseconds. Only two values are supported: • 10000 (which corresponds to 100Hz interval timer), and • 1000 (which corresponds to 1000Hz interval timer). By default it is set to 10000.

For example (AlphaServer 400, DS, ES, GS):

```
set ISA clock_period=1000
```

or (AlphaServer 800, 1000, 1000A, 2000, 2100, 4000, 4100):

```
set EISA clock_period=1000
```

Note that higher interval timer frequency creates higher load for virtual Alpha CPU which may cause degradation of overall virtual system performance.

8.11 Data storage in the virtualization layer

8.11.1 Physical disk drives and disk images

The following options are supported for the disk storage for the virtual HP Alpha:

1. Disk images, which are essentially binary files in the Linux file system. They could be located on a local or remote storage. They are easy to maintain and deliver good performance. Backup could be performed with standard Linux tools, making lengthy OpenVMS backups unnecessary. By copying an HP Alpha system or user disk back in place, the disk is fully restored.

   The disk images can easily be compressed and sent to a remote site, facilitating remote maintenance and upgrade of CHARON-AXP systems.

   It is NOT recommended to define disk images in network shared directories. A disconnect of the network storage will permanently disable access from CHARON-AXP to the remote disk image.

2. Physical disk drives connected to the Linux host system by host bus adapter or iSCSI Initiator. These disk drives must not be mounted in the host operating system; otherwise the drive is not available for use in CHARON-AXP.

   Using a host SCSI or iSCSI connection permits the use of FC, (S)ATA or SCSI drives on a storage backend and the possibility to configure these physical disks in a high reliability RAID of OpenVMS disk cluster configuration.

3. SAN attached storage volumes. These volumes must not be mounted in the host operating system; otherwise the drive is not available for use in CHARON-AXP.

4. CD and DVD drives on the host server can be used by the virtualization layer by specifying the usual Linux device name in the configuration script. For example: "/dev/sr0"

Note that disk images and physical disk drives offer similar I/O throughput. Disk images can be generated with the MKDISK utility.

8.11.2 Physical SCSI tape drives and tape images

Tape handling is implemented in CHARON-AXP in the following ways:

A SCSI tape drive can be connected to a SCSI controller in the Linux host system. The tape device is referenced in the configuration file with its usual Linux device name or file name. For instance "/dev/sgN" is a tape drive connected to the host system, and "/tape_images/mkc500.vtape" represents a virtual tape connected to a container file. Tape operation speed is essentially limited by the capabilities of the physical tape drive and the throughput of the SCSI connection.

8.11.3 Physical CD/DVD drives and CD/DVD images

The following options are supported for CD/DVD storage for virtual HP Alpha system:

- CD/DVD images, which are essentially binary files on the host system. They could be located on a local or remote storage. They are easy to maintain and deliver good performance.
- Physical CD/DVD drives attached to the host. Media in CD/DVD drives is shared with the host and may be mounted in both host and guest operating systems simultaneously. Nevertheless it is recommended to keep it mounted only in one system at a time and keep the automount daemon disabled on the host operating system.

8.11.4 Virtual Acer Labs 1543C IDE/ATAPI controller

The IDE[1] is an instance name for an integrated virtual Acer Labs 1543C IDE/ATAPI controller. Thus no "load" command is required to use it.

set parameter	Type	Value
container	Text string	Specifies the name of SCSI Generic interface to physical ATAPI or SATA CD/DVD-ROM drive attached to the host system. The supported values are of the form "/dev/sg*N*". For example: **set ide container="/dev/sg0"** By default it is left unspecified.

[1] Applicable to AlphaServer DS, ES, GS series only.

set parameter	Type	Value
		Note that when running HP OpenVMS/Alpha Operating System on top of CHARON-AXP virtualization layer the specified CD/DVD-ROM drive is available as **DQA0:** device.

The virtual Acer Labs 1543C IDE/ATAPI controller does NOT support CD/DVD images and physical CD/DVD drives other than ATAPI or SATA.

Please disable all the CD-ROM `automount` demons/software to avoid any problems accessing CD-ROM by CHARON-AXP.

8.11.5 Virtual KZPBA PCI SCSI adapter

The KZPBA is a PCI SCSI adapter (DEC-KZPBA, based on the QLogic ISP1040 Fast Wide SCSI adapter chip) for the HP Alpha. In CHARON-AXP it supports up to 120 disks and tapes.

Virtual KZPBA has the following features:

- *Up to 120 connected units (disks or tapes) supported in parallel.*
- *For systems with more than 16 heavily used units configure several virtual KZPBA PCI SCSI adapters and distribute the heavily loaded units evenly.*

8.11.5.1 Attaching virtual KZPBA PCI SCSI Adapter to virtual HP Alpha system

To create an instance of virtual KZPBA PCI SCSI Adapter use "load" command in configuration file as follows:

```
load KZPBA <instance-name>
```

Note that *<instance-name>* is not visible outside configuration file. Operating systems running on top of virtual HP Alpha system use different naming policy and name assigned to virtual KZPBA PCI SCSI Adapter by those operating systems has nothing to do with *<instance-name>* assigned in configuration files.

For example:

```
load KZPBA SCSI_A
```

In the above example, SCSI_A is instance name of virtual KZPBA PCI SCSI Adapter. But HP OpenVMS Alpha operating system uses names PKA,PKB,PKC,… to identify instances of virtual KZPBA PCI SCSI Adapters.

8.11.5.2 Configuring virtual KZPBA PCI SCSI Adapter

Virtual KZPBA PCI SCSI Adapter offers several configuration parameters controlling its behavior in virtual HP Alpha system and its appearance to software running on virtual HP Alpha system (HP OpenVMS Alpha and HP Tru64 UNIX operating systems).

Configuration parameters of virtual KZPBA PCI SCSI Adapter can be separated into several functional groups:

- Parameters controlling configuration of storage elements attached to virtual KZPBA PCI SCSI Adapter and their appearance to software running on virtual HP Alpha system. This group includes the following parameters:
    - *container*
    - *media_type*
    - *removable*
    - *use_io_file_buffering*
- Parameters controlling location of the virtual KZPBA PCI SCSI Adapter on virtual SCSI bus. This group includes the following parameters:
    - *scsi_id*
- Parameters controlling connection of the virtual KZPBA PCI SCSI Adapter to virtual PCI bus of virtual HP Alpha system and Adapter's appearance to software running on virtual HP Alpha system. This group includes the following parameters:
    - *bus*
    - *device*
    - *function*
    - *irq_bus*
    - *irq*

Detailed description of each configuration parameter of virtual KZPBA PCI SCSI Adapter is given below.

8.11.5.3 Parameters of virtual KZPBA PCI SCSI Adapter

8.11.5.3.1 CONTAINER

SYNTAX

```
container[unit-number]="{file-path\}file-name.vdisk"

container[unit-number]="{file-path\}file-name.vtape"

container[unit-number]="{file-path\}file-name.iso"

container[unit-number]="/dev/sdL"  (L is letter here)

container[unit-number]="/dev/sgN"

container[unit-number]="/dev/srN" | "/dev/cdrom"
```

Where unit-number is (scsi-id * 100 + lun). In most cases unit-number is 0, 100, 200, …, 1500 corresponding to device names such as DKA0, DKA100, DKA200, …, DKA1500 in OpenVMS.

DESCRIPTION

When specified, this configuration parameter instructs the CHARON-AXP software to create virtual SCSI device and connect to the virtual HP Alpha system through the virtual KZPBA SCSI Adapter. Type of the virtual SCSI device depends on value of the configuration parameter.

- The **.vdisk** file represents container of virtual disk. When path to **.vdisk** file is assigned to **container** configuration parameter the CHARON-AXP software creates virtual SCSI disk device. The CHARON-AXP software supports also **.dsk** files for backward compatibility, although use of **.dsk** extension is not recommended.
- The **.vtape** file represents container of virtual tape. When path to **.vtape** file is assigned to **container** configuration parameter the CHARON-AXP software creates virtual SCSI tape device. The CHARON-AXP software supports also **.mtd** files for backward compatibility, although use of **.mtd** extension is not recommended.
- The **.iso** file represents container of virtual cdrom. When path to **.iso** file is assigned to **container** configuration parameter the CHARON-AXP software creates virtual SCSI cdrom device.
- The `/dev/sdL` (L is letter here) object represents logical or physical disk attached to the host. It is also possible to use not a whole disk, but previously created partitions on it. In this case the syntax is the following: `/dev/sdLN` where N is the number of partition to be used. When certain `/dev/sdN` is assigned to **container** configuration parameter the CHARON-AXP software creates virtual SCSI disk device.
- The `/dev/sgN` object represents physical SCSI device attached to the host. Typically this parameter is used for connecting physical tape drives. In this case when certain `/dev/sgN` is assigned to **container** configuration parameter the CHARON-AXP software creates virtual SCSI tape device. In the same way one may connect physical SCSI devices of type other than tape.

  The following procedure is recommended for finding needed values for the `/dev/sgN` devices:

  In the console please issue:

```
cat /proc/scsi/sg/device_hdr; cat /proc/scsi/sg/devices
```

  The output will look something like:

```
host    chan    id     lun    type    opens    qdepth  busy    online
4       0       0      0      5       1        1       0       1
5       0       0      0      0       1        1       0       1
```

  The fifth field ("**type**") is the device type.
  5 means CD-ROM, 1 means tape, 0 means disk

  The "**N**" in the `/dev/sgN` is the line number in this table (starting from 0) corresponded to the devices CHARON-AXP will use.
  Thus `/dev/sg0` will be CD-ROM mapping in this example.

Another possibility is the following: on a freshly booted system please issue the following command:

```
dmesg | grep sg
```

The output will look like that:

```
[ 1.503622] sr 4:0:0:0: Attached scsi generic sg0 type 5
[ 1.780897] sd 5:0:0:0: Attached scsi generic sg1 type 0
```

Note that this table lists all the devices, not only the real SCSI ones (SATA/IDE for example). CHARON-AXP supports only real SCSI devices.

- The `/dev/srN` (`/dev/cdrom` syntax is also possible) object represents logical or physical optical drive attached to the host. When certain `/dev/srN` is assigned to **container** configuration parameter the CHARON-AXP software creates virtual SCSI cdrom device.

If the **container** configuration parameter is *not* specified, the CHARON-AXP software does *not* create virtual SCSI device for the corresponding unit number.

By default the **container** configuration parameter is *not* specified.

Please note that you should use the `/dev/sgN` and the `/dev/sdL` devices very carefully since in the case of specifying some partitions or disks incorrectly (providing that the user has all the required rights) the Linux system may be damaged or even destroyed completely.

EXAMPLES

1. Virtual HP AlphaServer ES40 system with two virtual disks DKA0 and DKA100 and virtual tape MKA600:

```
set session hw_model=AlphaServer_ES40
...
load KZPBA SCSI_A bus=pci_1 device=1 function=0
...
set SCSI_A container[0]="/opt/charon/disks/dka0.vdisk"
set SCSI_A container[100]="/opt/Charon/disks/dka100.vdisk"
...
set SCSI_A container[600]="/opt/Charon/disks/mka600.vtape"
...
```

2. Virtual HP AlphaServer 400 system with physical CD-ROM drive attached as DKB600:

```
set session hw_model=AlphaServer_400
...
load KZPBA SCSI_B
...
set SCSI_B container[600]="/dev/sr0"
...
```

In this example note that AlphaServer 400 has on-board primary SCSI controller which appears as PKA to HP OpenVMS/Alpha operating system.

Please disable all the CD-ROM `automount` demons/software to avoid any problems accessing CD-ROM by CHARON-AXP.

3. Virtual HP AlphaServer 4100 system with tape changer attached as MKB500:

```
set session hw_model=AlphaServer_4100
...
load KZPBA SCSI_B
...
set SCSI_B container[500]="/dev/sg2"
...
```

In this example note that AlphaServer 4100 has on-board primary SCSI controller which appears as PKA to HP OpenVMS/Alpha operating system.

8.11.5.3.2 MEDIA TYPE

SYNTAX

media_type[*unit-number***]="***string***"**

Where:

- *unit-number* is the same as for **container** configuration parameter.

DESCRIPTION

When specified, the **media_type** configuration parameter instructs the CHARON-AXP software to use the supplied value as PRODUCT field in SCSI INQUIRY data returned to software running on virtual HP Alpha system in response to SCSI INQUIRY command.

Bye default the **media_type** configuration parameter is *not* specified.

If the **media_type** configuration parameter is *not* specified, the CHARON-AXP software attempts to guess SCSI INQUIRY data based on virtual SCSI device type and underlying container (which is specified in the corresponding **container** configuration parameter).

EXAMPLES

1. Make sure that virtual SCSI disk appears as DEC HSZ70, and virtual SCSI cdrom device appears as DEC RRD43:

```
set session hw_model=AlphaServer_400
...
load KZPBA SCSI_B
...
set SCSI_B container[0]="dkb0.vdisk"
set SCSI_B media_type[0]="HSZ70"
...
set SCSI_B container[600]="/dev/cdrom"
set SCSI_B media_type[600]="RRD43"
...
```

8.11.5.3.3 REMOVABLE

SYNTAX

```
removable[unit-number]=true|false
```

Where:

- *unit-number* is the same as for **container** configuration parameter.

DESCRIPTION

When set to TRUE, the **removable** configuration parameter instructs the CHARON-AXP software to report the corresponding virtual SCSI device as removable.

By default the **removable** configuration parameter is set to FALSE.

Note that virtual SCSI tape and cdrom devices are always reported as removable regardless of the **removable** configuration parameter.

Note that HP Tru64 UNIX operating system does not install on removable virtual SCSI disk device of the virtual HP Alpha system.

EXAMPLES

1. Make virtual SCSI disk device of virtual HP AlphaServer ES40 removable:

```
set session hw_model=AlphaServer_ES40
...
load KZPBA SCSI_A
...
set SCSI_A container[400]="/opt/charon/disks/d4.vdisk"
set SCSI_A removable[400]=true
...
```

8.11.5.3.4 USE_IO_FILE_BUFFERING

SYNTAX

```
use_io_file_buffering[unit-number]=true|false
```

Where:

- *unit-number* is the same as for **container** configuration parameter.

DESCRIPTION

When set to TRUE, the **use_io_file_buffering** configuration parameter instructs the CHARON-AXP software to enable host operating system I/O cache when reading/writing the corresponding container (specified by the corresponding **container** configuration parameter).

When enabled, the host operating system I/O cache may significantly improve I/O performance of the virtual HP Alpha system. At the same time maintaining I/O cache requires additional host resources (CPU and memory) which may negatively affect overall performance of the virtual HP Alpha system. Use this option with care, especially with virtual HP SMP Alpha systems (such as virtual AlphaServer ES40).

By default the **use_io_file_buffering** configuration parameter is set to FALSE.

Note that host operating system I/O cache is always enabled for **.vtape** file containers regardless of the **use_io_file_buffering** configuration parameter. The CHARON-AXP software takes specific actions to avoid possible negative impact of the corresponding virtual SCSI tape device on overall performance of the virtual HP Alpha system.

EXAMPLES

2. Enable use of operating system I/O cache for virtual SCSI disk device of virtual HP AlphaServer ES40:

```
set session hw_model=AlphaServer_ES40
...
load KZPBA SCSI_A
...
set SCSI_A container[0]="/opt/charon/disks/d0.vdisk"
set SCSI_A use_io_file_buffering[0]=true
...
```

8.11.5.3.5 SCSI ID

SYNTAX

```
scsi_id=0|1|2|...|14|15
```

DESCRIPTION

The **scsi_id** configuration parameter specifies self SCSI ID (Initiator SCSI ID) of the virtual KZPBA PCI SCSI Adapter. The same SCSI ID is also used by virtual KZPBA PCI SCSI Adapter when it is configured as virtual SCSI target in virtual SCSI cluster configuration.

By default the **scsi_id** configuration parameter is set to 7.

EXAMPLES

1. Loading two virtual KZPBA PCI SCSI Adapters with different self SCSI IDs:

```
set session hw_model=AlphaServer_ES40
...
load KZPBA SCSI_A bus=pci_1 device=1 function=0
set SCSI_A scsi_id=7
...
load KZPBA SCSI_B bus=pci_1 device=2 function=0
set SCSI_B scsi_id=6
...
```

8.11.5.3.6 BUS

SYNTAX

```
bus=virtual-pci-bus-instance-name
```

DESCRIPTION

When specified, the **bus** configuration parameter tells the CHARON-AXP software the virtual PCI bus to which the virtual HP Alpha system shall connect the virtual KZPBA PCI SCSI Adapter.

By default the **bus** configuration parameter is not specified.

If the **bus** configuration parameter is not specified, the CHARON-AXP software connects the virtual KZPBA PCI SCSI Adapter to the first available virtual PCI bus.

Name of virtual PCI bus is selected depending on particular hardware model of virtual HP Alpha system.

Note that the four configuration parameters **bus**, **device**, **function**, and **irq_bus** (for virtual HP AlphaServer 400 system only) must be specified together or all left not specified to correctly identify location of the virtual KZPBA PCI SCSI Adapter in the virtual HP Alpha system.

Examples

1. Loading virtual KZPBA PCI SCSI Adapter in virtual HP AlphaServer ES40 system into specific virtual PCI slot:

```
set session hw_model=AlphaServer_ES40
...
load KZPBA SCSI_A bus=pci_1 device=1 function=0
...
```

2. Loading virtual KZPBA PCI SCSI Adapter in virtual HP AlphaServer GS320 system into specific virtual PCI slot:

```
set session hw_model=AlphaServer_GS320
...
load KZPBA SCSI_A bus=qbb_7_pca_0_pci_1 device=4 function=0
...
```

8.11.5.3.7 DEVICE

SYNTAX

```
device=0|1|...|19
```

DESCRIPTION

When specified, the **device** configuration parameter specifies position of the virtual KZPBA PCI SCSI Adapter on virtual PCI bus.

By default the **device** configuration parameter is not specified.

If the **device** configuration parameter is not specified, the CHARON-AXP software connects the virtual KZPBA PCI SCSI Adapter at the first available position of the virtual PCI bus.

Note that the four configuration parameters **bus**, **device**, **function**, and **irq_bus** (for virtual HP AlphaServer 400 system only) must be specified together or all left not specified to correctly identify location of the virtual KZPBA PCI SCSI Adapter in the virtual HP Alpha system.

EXAMPLES

See examples for **bus** configuration parameter.

8.11.5.3.8 FUNCTION

SYNTAX

```
function=0|1|2|3
```

DESCRIPTION

When specified, the **function** configuration parameter specifies position of the virtual KZPBA PCI SCSI Adapter on virtual PCI bus.

By default the **function** configuration parameter is not specified.

If the **function** configuration parameter is not specified, the CHARON-AXP software connects the virtual KZPBA PCI SCSI Adapter at the first available position of the virtual PCI bus.

Note that the four configuration parameters **bus**, **device**, **function**, and **irq_bus** (for virtual HP AlphaServer 400 system only) must be specified together or all left not specified to correctly identify location of the virtual KZPBA PCI SCSI Adapter in the virtual HP Alpha system.

EXAMPLES

See examples for **bus** configuration parameter.

8.11.5.3.9 IRQ_BUS

SYNTAX

```
irq_bus=virtual-bus-instance-name
```

DESCRIPTION

When specified, the **irq_bus** configuration parameter specifies virtual bus routing interrupt requests from virtual KZPBA PCI SCSI Adapter to virtual Alpha CPUs in the virtual HP Alpha system.

By default the **irq_bus** configuration parameter is not specified.

The **irq_bus** configuration parameter *must* be set to "**ISA**" for virtual KZPBA SCSI Adapter in virtual AlphaServer 400. For virtual HP Alpha systems other than AlphaServer 400 the **irq_bus** configuration parameter must be left as is (i.e. not specified).

Note that the four configuration parameters **bus**, **device**, **function**, and **irq_bus** (for virtual HP AlphaServer 400 system only) must be specified together or all left not specified to correctly identify location of the virtual KZPBA PCI SCSI Adapter in the virtual HP Alpha system.

EXAMPLES

1. Loading virtual KZPBA PCI SCSI Adapter as secondary SCSI adapter on a virtual HP AlphaServer 400 system:

   ```
   set session hw_model=AlphaServer_400
   ...
   load KZPBA SCSI_B irq_bus=isa
   ...
   ```

2. Loading virtual KZPBA PCI SCSI Adapter as secondary SCSI adapter on a virtual HP AlphaServer ES40 system (note missing **irq_bus** configuration parameter):

   ```
   set session hw_model=AlphaServer_ES40
   ...
   load KZPBA SCSI_B
   ...
   ```

8.11.5.3.10 IRQ

SYNTAX

```
irq=interrupt-request-number
```

DESCRIPTION

When specified, the **irq** configuration parameter assigns interrupt request to the virtual KZPBA PCI SCSI Adapter in the virtual HP Alpha system.

By default the **irq** configuration parameter is not specified.

If the **irq** configuration parameter is not specified, the CHARON-AXP software uses the correct value depending on the selected PCI position of virtual KZPBA PCI SCSI Adapter in the virtual HP Alpha system.

EXAMPLES

1. Loading virtual KZPBA PCI SCSI Adapter in virtual HP AlphaServer ES40 system into specific virtual PCI slot (same as above but with explicit assignment for **irq** configuration parameter):

   ```
   set session hw_model=AlphaServer_ES40
   ...
   load KZPBA SCSI_A bus=pci_1 device=1 function=0 irq=24
   ...
   ```

8.12 Virtual PCI Ethernet controllers

CHARON-AXP implements the following virtual PCI Network controllers:

- *DE435*
- *DE450*

- *DE500AA*
- *DE500BA*

Each of them is a PCI Ethernet adapter (based on the DEC21040 PCI Ethernet adapter chip) for the HP Alpha. CHARON-AXP maps the virtual adapter to a dedicated Ethernet adapter in the Linux host system.

The Ethernet adapter in the Linux host system must support dynamic changes of its MAC address (i.e. no reboot of the host system is required to change the MAC address), which is the case with nearly all modern Ethernet adapters.

The proper sequences is to first load an instance of virtual Ethernet controller, then load an instance of virtual network interface connected to the `"ethN"` network interface, and then finally link the two virtual entities. For example:

```
load DE500BA/dec21x4x IFC
load packet_port /chnetwrk IFC0 interface="eth0"
set IFC interface=IFC0
```

8.12.1 Virtual DEXXX configuration parameters

Load parameter	Function
DEXXX/dec21x4x	This command creates an instance of the DEXXX Ethernet controller and associates it with a logical name.

Examples:

```
load DE435/dec21x4x EWA
load DE450/dec21x4x EWB
load DE500AA/dec21x4x EWC
load DE500BA/dec21x4x EWD
```

Set parameter for DEXXX	Type	Value
Parameters controlling connection to virtual network interface		
Interface	Text string	This connects the logical name representing a DEXXX instance with the logical name of a host network port, after the host network port is loaded.
Parameters controlling connection of virtual PCI adapter to virtual PCI bus		

Set parameter for DEXXX	Type	Value
Bus	Text string	Specifies to which virtual PCI bus the virtual PCI adapter shall be connected. In combination with "device", "function", "irq", and "irq_bus" parameters allows replicating exact configuration of hardware HP Alpha machine. These five parameters are advanced configuration options which should be handled with care. Possible combinations of values depend on particular CHARON-AXP virtualization layer and are given in "The virtual AXP models specifics". By default the parameter is left unspecified.
Device	Numeric	Together with "function" parameter specifies "position" of virtual PCI adapter on virtual PCI bus. In combination with "bus", "function", "irq", and "irq_bus" parameters allows replicating exact configuration of hardware HP Alpha machine. These five parameters are advanced configuration options which should be handled with care. Possible combinations of values depend on particular CHARON-AXP virtualization layer and are given in "The virtual AXP models specifics". By default the parameter is left unspecified.
Function	Numeric	Together with "device" parameter specifies "position" of virtual PCI adapter on virtual PCI bus. In combination with "bus", "device", "irq", and "irq_bus" parameters allows replicating exact configuration of hardware HP Alpha machine. These five parameters are advanced configuration options which should be handled with care. Possible combinations of values depend on particular CHARON-AXP virtualization layer and are given in "The virtual AXP models specifics". By default the parameter is left unspecified.

Set parameter for DEXXX	Type	Value
Irq	Number	Specifies the virtual IRQ number assigned to the virtual PCI adapter. In combination with "bus", "device", "function", and "irq_bus" parameters allows replicating exact configuration of hardware HP Alpha machine. These five parameters are advanced configuration options which should be handled with care. Possible combinations of values depend on particular CHARON-AXP virtualization layer and are given in "The virtual AXP models specifics". By default the parameter is left unspecified.
irq_bus	Text string	Specifies the virtual bus routing virtual interrupts from virtual PCI adapter to virtual CPU. In combination with "bus", "device", "function", and "irq" parameters allows replicating exact configuration of hardware HP Alpha machine. These five parameters are advanced configuration options which should be handled with care. Possible combinations of values depend on particular CHARON-AXP virtualization layer and are given in "The virtual AXP models specifics". By default the parameter is left unspecified. It must remain unspecified for any virtual HP Alpha system except AlphaServer 400. It must be set to "isa" for any virtual PCI adapter on virtual AlphaServer 400. For example: **load DE435/dec21x4x ETH1 irq_bus=isa**

Example (Assuming that the network packet port (see the corresponding paragraph below) is defined as `EWx0`):

```
load DE435/dec21x4x EWA interface=EWA0
load DE450/dec21x4x EWB interface=EWB0
load DE500AA/dec21x4x EWC interface=EWC0
load DE500BA/dec21x4x EWD interface=EWD0
```

It is recommended to review the sample configuration files to see the correct structure of the Ethernet configuration commands.

If your OpenVMS/Alpha system disk is configured for automatic TCP/IP startup and you use UCX, not loading an Ethernet adapter in the CHARON-AXP configuration can cause

OpenVMS to crash. The problem appears only if UCX is enabled while the networking device is missing. DECnet works correctly.

For the extended set of the adapter configuration parameters, tuning and troubleshooting please refer to the 'Charon networking Guide'.

8.12.2 CHARON Packet Port configuration parameters

The CHARON specific `"packet_port"` interface establishes a connection between an Ethernet adapter in the Linux host system and Ethernet adapter in the virtual HP Alpha system. For every DE435/DE500BA instance loaded, one dedicated host Ethernet adapter is required.

Load parameter	***Function***
packet_port	This command associates an Ethernet adapter in the Linux host system with an Ethernet adapter in the virtual HP Alpha system.

Example:

```
load packet_port/chnetwrk EXA0
```

It is convenient to extend the load command with the interface ID assignment. See the examples below.

set parameter for packet_port	***Type***	***Value***
interface	Text string	Identifies the dedicated Ethernet adapter in the Linux host system. Example of usage: :**set <name> interface="<adapter>"**

Example:

```
load DE500BA/dec21x4x EWA
load packet_port/chnetwrk EXA_0 interface="eth0"
set EWA interface=EXA_0
```

For the extended set of the port configuration parameters, tuning and troubleshooting please read the *"CHARON-VAX and CHARON-AXP/4100/DS/ES/GS networking Guide".*

9 Operating CHARON-AXP

When CHARON-AXP starts, license checking takes a few seconds. If you remove the license key while CHARON-AXP is running, a warning message is given after a few minutes, and you have a maximum of 1 hour to save your files and shut down your virtual HP Alpha system.

However it is still possible to replace CHARON-AXP license with new one without stopping CHARON-AXP. The emulator will recheck the new license and continue to work normally. Note that new license key must be plugged in as soon as possible once the old one is removed.

It is also possible to update content of the license w/o stopping CHARON-AXP. Please use specific utilities provided by Sentinel for doing that (for example HASPRUS and its analogs).

If CHARON-AXP does not start, you might have a license key with a product revision code different than the installed version. Since CHARON-AXP has not yet read the configuration file with the log file definition, its log file cannot be updated.

10 Utilities

The CHARON-AXP installation includes several utilities (located in the "/utils/axp" and "/disks/axp" folders) that help to create empty disk containers, verify content of the CHARON-AXP license and manage the way CHARON-AXP utilizes the host system resources.

10.1 The "mkdskcmd" utility

The "mkdskcmd" utility creates empty disk images of given standard disk drive types or of custom disk images.

The first step is obtaining the name of the disk that should be created:

```
mkdskcmd --list
```

This command results in getting a list of all supported disk types. Choose a desired disk (for example RZ22) and command the "mkdskcmd" to create a virtual disk image:

```
mkdskcmd --disk rz22 --output rz22.vdisk
```

The disk container "rz22.vdisk" will be created in the current directory

It is also possible to create custom disk image using switches "--blcount" and "--blsize".

To get all the available parameters please use the switch "--help"

10.2 The "hasp_srm_view" utility

The *"hasp_srm_view"* utility allows seeing content of the license that is embedded in your CHARON-AXP HASP-SRM license key. Just run this utility w/o any parameters to see the license details.

10.3 The "idle" utility (applicable only for models with single CPU emulation)

The "Idle" utility significantly reduces the CHARON-AXP host CPU usage whenever a VMS/Alpha system running on CHARON-AXP is idle. "Idle" utility stalls the emulated CPU (note that at the moment it supports the models emulating just 1 CPU only, namely: AlphaStation 400, 800, 1000, 1000A and DS10L) when it detects an OpenVMS idle condition. While the "Idle" utility is running the emulated CPU consumes, on average, less host system CPU time. However it is not recommended to employ "Idle" utility in real-time process control environments.

The supported OpenVMS versions are from V6.2-1H3 up to V8.4. Common PCSI distributive is used for all the versions of OpenVMS.

The "Idle" utility is provided in form of a virtual disk image named "idle_vms_pkg.vdisk". Mount this disk with the "over=id" qualifier under the emulated VMS/Alpha operating system and go to the "[000000.AXP]" directory.

The following files are resided there:

```
README.TXT
SRI-AXPVMS-IDLE-V0102--1.PCSI
VMS62TO71U2_PCSI-V0200.PCSI-DCX_AXPEXE
VMS62TO71U2_PCSI-V0200.TXT
```

At the first step it is needed to apply a specific PCSI patch "VMS62TO71U2_PCSI" if the target VMS/Alpha operating system version is below V7.2. Copy the "*VMS62TO71U2_PCSI-V0200.PCSI-DCX_AXPEXE*" file to some directory on any spare disk and run this file from there:

```
$ RUN VMS62TO71U2_PCSI-V0200.PCSI-DCX_AXPEXE
```

then proceed with the patch installation:

```
$ PRODUCT INSTALL VMS62TO71U2_PCSI /SOURCE=<directory containing the VMS62TO71U2_PCSI kit>
```

Once the installation is over please return to the "[000000.AXP]" directory of the "idle_vms_pkg.vdisk" and proceed with installation of the "Idle" utility itself:

```
$ PRODUCT INSTALL IDLE /SOURCE=<directory containing the IDLE kit>
```

Once the "Idle" utility is installed it starts to take effect immediately, reducing the host system CPU usage if VMS/Alpha system running on CHARON-AXP is idle. No reboot is required. The utility is loaded automatically on reboot, no additional configuring or startup sequence is needed.

Deinstallation of the "Idle" utility:

```
$ PRODUCT REMOVE IDLE
```

The utility stops working after the system reboot.

Please also refer to the supplied documents "*README.TXT*" and "*VMS62TO71U2_PCSI-V0200.TXT*" for more details.

11 Installing and transferring HP Alpha software

There are several ways to transfer data from a HP Alpha system to CHARON-AXP:

11.1 Using the Local Area Network

First, perform a standard installation of your HP Alpha Operating System from the manufacturer's original media using the Linux host CD-ROM drive. Then configure a network (DECnet and/or TCP/IP) to your CHARON-AXP for your existing Network with a unique address, and use DECnet or TCP/IP to copy your applications and data to your CHARON-AXP system. If for any reason installing a HP Alpha Operating System from scratch is a problem, call your CHARON-AXP sales contact for help. Once you have CHARON-AXP connected to your network, you may use standard utilities to transfer the required data. Before copying the data you will have to configure CHARON-AXP with adequate free space on disks, or on disk images which can be created with the `"mkdskcmd"` utility.

11.2 Using a physical disk drive

You can remove a SCSI disk from your HP Alpha system and reconnect it to a SCSI adapter on your Linux system. Assign the SCSI disk within the CHARON-AXP configuration file to a disk controller, and it becomes a disk drive in the CHARON-AXP. If the SCSI disk is a bootable OpenVMS/Alpha disk from an HP ALPHA, you can boot CHARON-AXP from it.

11.3 Using a tape

CHARON-AXP supports the connection of a SCSI tape drive to a SCSI adapter in your Linux system. Assign the tape drive in the CHARON-AXP configuration file to access the tape drive by the HP Alpha operating system. This way you can boot from standalone tape to restore your system backup.

Note that the reliability of a physical tape connection depends on factors like SCSI controller type, tape drive model and host CPU speed. Tape connections are not guaranteed to work in all cases.

Appendix A: Configuration files examples

This section provides the following configuration file example: "The virtual HP AlphaServer ES40 configuration template. (e.g. es40.cfg.template)"

This file contains basic information on how to set configuration parameters for the emulated devices provided by CHARON-AXP AlphaServer ES40. Make a copy and edit it to set up the connections to your disks, disks images, tape drives, network adapters, etc.

Note: In the CHARON-AXP installation directory you can find the *as400.cfg.template, as800.cfg.template, as1000.cfg.template, as1000a.cfg.template, as1200.cfg.template, as2000.cfg.template, as2100.cfg.template, as4000.cfg.template, as4100.cfg.template, ds10l.cfg.template, ds20.cfg.template, es40.cfg.template, gs80.cfg.template, gs160.cfg.template, and gs320.cfg.template* files for the particular model installed.

```
#
# Copyright (C) 1999-2011 STROMASYS
# All rights reserved.
#
# The software contained on this media is proprietary to and embodies
# the confidential technology of STROMASYS. Posession, use, duplication,
# or dissemination of the software and media is authorized only pursuant
# to a valid written license from STROMASYS.
#
#============================================================================
#
# Sample configuration file for AlphaServer ES40 (code name Clipper)
# machines.
#
#----------------------------------------------------------------------------

set session hw_model="AlphaServer_ES40"

#============================================================================
#
# Comment the following line if you do not want the log to be saved into
# file (change name of the file as well if you'd like). Each new session of
# the emulator appends its log to this file, therefore it grows bigger with
# time.
#
#----------------------------------------------------------------------------

set session log="AlphaServer_ES40.log" log_method="append"
#set session log="AlphaServer_ES40.log" log_method="overwrite"

#============================================================================
#
# Overrides system assigned process's CPU affinity. The session changes
# the process's CPU affinity to the one specified.
#
#----------------------------------------------------------------------------

#set session affinity="0, 1, 2, 3"

#============================================================================
#
# The 'n_of_io_cpus' option overrides number of host CPU cores reserved for
# I/O processing. If ommited the session reserves 33% of available host CPU
# cores for I/O processing. Note that total amount of available host CPU
# cores is determined based on process's CPU affinity.
```

```
#
#-----------------------------------------------------------------------------

#set session n_of_io_cpus=1
#set session n_of_io_cpus=2
#set session n_of_io_cpus=...

#=============================================================================
#
# The 'n_of_cpus' option reduces number of emulated Alpha CPUs in the
# configuration.
#
#-----------------------------------------------------------------------------

#set session n_of_cpus=1
#set session n_of_cpus=2
#set session n_of_cpus=3

#=============================================================================
#
# Specify size of RAM from 256MB up to 32768MB (32GB) in 256MB extents.
#
#-----------------------------------------------------------------------------

#set ram size=256
#set ram size=512
#set ram size=1024
#set ram size=4096
#set ram size=32768

#=============================================================================
#
# Uncomment to allow the SRM console environment be preserved across
# emulator restarts.
#
#-----------------------------------------------------------------------------

#set rom container="clipper.bin"

#=============================================================================
#
# Uncomment to allow saving CMOS NVRAM content, so that to preserve
# Time & Date information.
#
#-----------------------------------------------------------------------------

#set toy container="clipper.dat"

#=============================================================================
#
# Select connection for the console serial line OPA0.
#
#-----------------------------------------------------------------------------

#load physical_serial_line OPA0 line="/dev/ttyN"
load operator_console OPA0

#=============================================================================
#
# Improve granularity of emulated AXP timer.
#
#-----------------------------------------------------------------------------

#set isa clock_period=1000

#=============================================================================
#
# Uncomment to connect the emulator's DQA0 to host's ATAPI CD/DVD-ROM drive.
#
#-----------------------------------------------------------------------------
```

```
#set ide container="/dev/sgN"

#=============================================================================
#
# Load optional DE500BA PCI Ethernet Adapter (EWA).
#
#-----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWA interface=EWA0
#load packet_port/chnetwrk EWA0 interface="eth0"

#=============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWB).
#
#-----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWB interface=EWB0
#load packet_port/chnetwrk EWB0 interface="eth1"

#=============================================================================
#
# Load another optional DE500BA PCI Ethernet Adapter (EWC).
#
#-----------------------------------------------------------------------------

#load DE500BA/dec21x4x EWC interface=EWC0
#load packet_port/chnetwrk EWC0 interface="eth2"
#=============================================================================
#
# Uncomment to enable emulation of DEC-KZPBA SCSI controller.
#
#-----------------------------------------------------------------------------

#load KZPBA PKA scsi_id=7

#=============================================================================
#
# Uncomment to connect the emulator's DKA0 to the disk image.
#
#-----------------------------------------------------------------------------

#set PKA container[0]="<file-name>.vdisk"

#=============================================================================
#
# Uncomment to connect the emulator's DKA100 to host's disk drive.
#
#-----------------------------------------------------------------------------

#set PKA container[100]="/dev/sdL"

#=============================================================================
#
# Uncomment to connect the emulator's GKA200 to an unknown SCSI device.
#
#-----------------------------------------------------------------------------

#set PKA container[200]="/dev/sgN"

#=============================================================================
#
# Uncomment to connect the emulator's DKA300 to host's CD/DVD-ROM drive.
#
#-----------------------------------------------------------------------------

#set PKA container[300]="/dev/cdrom"
#set PKA container[300]="/dev/srN"

#=============================================================================
#
```

```
# Uncomment to connect the emulator's DKA400 to .ISO file (CD/DVD-ROM image).
#
#----------------------------------------------------------------------------

#set PKA container[400]="<file-name>.iso"

#============================================================================
#
# Uncomment to connect the emulator's MKA500 to host's SCSI tape drive.
#
#----------------------------------------------------------------------------

#set PKA container[500]="/dev/sgN"

#============================================================================
#
# Uncomment to connect the emulator's MKA600 to .VTAPE file (tape image).
#
#----------------------------------------------------------------------------

#set PKA container[600]="<file-name>.vtape"

# this is the end of the configuration file #################################
```

Appendix B: Reader's Comment Form

We appreciate your comments, suggestions, criticism and updates for this manual. Please contact Stromasys at info@stromasys.com for comments and suggestions, and at support@stromasys.com for product support issues.

Please refer to the document **60-16-010-005**

If you found any errors, please list them with their page number