

CONVEX

■ Exemplar
■ Open Boot

■ First Edition

CONVEX Exemplar OpenBoot Quick Reference

Order No. DSW-854

Document No. 710-029430-000

Released with CONVEX Exemplar OpenBoot V1.1.

Copyright ©1994 CONVEX Computer Corporation
All rights reserved.

CONVEX and the CONVEX logo "C" are registered trademarks
of CONVEX Computer Corporation.

QUICK REFERENCE

Exemplar OpenBoot

Order No. DSW-854

First Edition

October 1994

Convex Computer Corporation
3000 Waterview Parkway
Richardson, Texas 75080 U.S.A.
214 497-4000

Syntax

Commands entered at the OpenBoot ok prompt are executed left-to-right after a carriage return is typed. Multiple commands separated by one or more spaces may be entered on a single line.

Help Commands

help	List main help categories.
†help <i>category</i>	Show help for all commands in the category. Use only the first word of the category description.
†help <i>command</i>	Show help for individual command (where available).

† This command will be implemented in a future release of Exemplar OpenBoot.

System Information Display Commands

banner	Display the power-on banner.
.version	Display the release and build version of the OBP loader.
.memory	Display the memory configuration of the current hypervisor.

Restricted Monitor Commands

b [<i>specifiers</i>]	Boot the operating system (same as boot at ok prompt).
c	Resume the execution of a halted program (same as go at ok prompt).
f	Enter the Forth interpreter.

Examining and Creating Device Aliases

dealias	Display all current device aliases.
dealias <i>alias</i>	Display the device path corresponding to <i>alias</i> .
dealias <i>alias device-path</i>	Define an alias representing the device path. If an alias with the same name already exists, the new value supersedes the old.

Device Tree Browsing Commands

.attributes	Display the names and values of the current node's properties. (.properties is an alias for .attributes.)
cd <i>device-path</i>	Select the indicated device node, making it the current node.
cd <i>node-name</i>	Select the first node with the given name in the subtree below the current node.
cd ..	Select the device node that is the parent of the current node.
cd /	Select the root machine node.
device-end	Deselect the current device node, leaving no node selected.
ls	Display the names of the current node's children.
pwd	Display the device path name that names the current node.
show-devs [<i>device-path</i>]	Display all the devices known to the system directly beneath a given level in the device hierarchy. (Used by itself, it shows the entire device tree.)
words	Display the names of the current node's methods.

Common Options for the boot Command

boot [*device*] [*boot-directory*] [*boot-arguments*]

[<i>device</i>]	The name (full path name or alias) of a device. Examples: sd0a (alias for default SCSI disk root partition) rmt0 (alias for default SCSI DAT install tape drive) /landmarc/sbus/Convex,afws:narrow/sd@2,0:a fddi0 (alias for FDDI network controller)
[<i>boot-directory</i>]	The HFS full pathname of the directory in the boot partition containing the three boot files (mach, server, tunables). Valid only for disk devices
[<i>boot-arguments</i>]	-s - Boot SPP-UJ in single-user mode. (iff boot-file=mach) -mk_debug - Microkernel will wait for <i>kgdb</i> to attach. -obp_panic - Enter OBP after kernel panic. -halt - Halt after loading (e.g., to set breakpoints). -noheader - boot-file is not an a.out. (.fth or CODE files)

Diagnostic Test Commands

probe-landmarc	Identify devices attached to the Landmarc I/O SBus bridge controller.
†probe-scsi-all [<i>device-path</i>]	Perform <i>probe-scsi</i> on all SCSI buses installed in the system below the specified node. (If <i>device-path</i> is absent, the root node is used).
test <i>device-specifier</i>	Execute the specified device's self-test method.
†test-all [<i>device-specifier</i>]	Test all devices (that have a built-in self-test method) below the specified node. (If <i>device-specifier</i> is absent, the root node is used.)
†watch-clock	Test the clock function.
†watch-net	Monitor the network connection.

† This command will be implemented in a future release of Exemplar OpenBoot.

Emergency Keyboard Commands

Hold down keys during power-on sequence.

ESC Interrupt auto-boot sequence when auto-boot? parameter is true

File Loading Commands

boot [<i>specifiers</i>] -halt	(-)	Load file(s) from specified source.
source <i>pathname</i>	(-)	Load and interpret a Forth file from the default boot-device filesystem.
byte-load	(<i>adr span -</i>)	Interpret a loaded FCode binary file. <i>span</i> is usually 1.
dl	(-)	Load a Forth file via the Console.
dlbin	(-)	Load a binary file via the Console.
dload <i>filename</i>	(<i>adr -</i>)	Load specified file over Network at given address.
go	(-)	Begin executing a previously-loaded binary program, or resume executing an interrupted program.
load [<i>specifiers</i>]	(-)	Load file(s) from specified source. Specifiers are as for <i>boot</i> .

setenv boot-device sd1a
setenv boot-args -root sd1a

PA-RISC Register Commands

%r0 through %r31	(- value)	Return the value in the general register.
%cr0 through %cr31	(- value)	Return the value in the control register.
%sr0 through %sr7	(- value)	Return the value in the space register.
%fr0l through %fr31r	(- value)	Return the value in the named 32-bit floating point register.
regname	(- value)	Return the value in a 32-bit register.
.regs .gdbregs	(-)	Display general registers %r0-%r31, space registers %sr0-%sr7, the pc queue and the program status word. .gdbregs uses display similar to kgdb.
.psw	(-)	Display the PSW and PC queue.
.fregs	(-)	Display values in %f0 through %f31 as 64-bit hexadecimal numbers.
.spregs	(-)	Display values in %fr0L through %fr31L and %fr0R through %fr31R as 32-bit hexadecimal numbers.
.cregs	(-)	Display values in %cr0 through %cr31.
.obr	(-)	Like .regs, using Forth machine names.
ctrace	(-)	Display the return stack showing C subroutines.
set-pc	(value -)	Set %pcoq_head to the given value, and set %pcoq_tail to (value+4).
to regname	(value -)	Change the value of a 32-bit register. Use in the form: <i>value to regname</i> .

Breakpoint Commands

+bp	(adr -)	Add a breakpoint at the given address.
-bp	(adr -)	Remove the breakpoint at the given address.
-bp	(-)	Remove the most-recently-set breakpoint.
.bp	(-)	Display all currently set breakpoints.
.breakpoint	(-)	Perform a specified action when a breakpoint occurs (Example, ['] .regs is .breakpoint).
.instruction	(-)	Display the address, opcode for the last-encountered breakpoint.
.step	(-)	Perform a specified action when a single step occurs (see .breakpoint).
bpoff	(-)	Remove all breakpoints.
finish-loop	(-)	Execute until the end of this loop.
go	(-)	Continue from a breakpoint.
gos	(n -)	Execute go n times.
hop	(-)	Treats a subroutine call as a single instruction.
hops	(n -)	Execute hop n times.
skip	(-)	Skip (do not execute) the current instruction.
step	(-)	Single-step one instruction.
steps	(n -)	Execute step n times.
till	(adr -)	Execute until the given address is encountered.

single booter: ok boot -s

writer in multi: init 2

Disassembler Commands

dis	(adr -)	Begin disassembling at the given address.
+dis	(-)	Continue disassembling where the last dis left off.
disasm	(instr -)	Disassemble one instruction on the top of stack.
disflag	(-)	Toggle disassembler continuation at branch instructions.

Miscellaneous Operations

ftrace	(-)	Show calling sequence when exception occurred.
get-msecs	(- ms)	Return the approximate current time in milliseconds.
ms	(n -)	Delay for n milliseconds. Resolution is 1 millisecond.
reset	(-)	Reset the current hypemode or the complex.
date [datestring]	(-)	Display or set the clock/calendar chip date. (Syntax is like HP-UX date(8) command).

NVRAM Configuration Parameters

auto-boot?	true	If true, boot automatically after power-on or reset. (interrupt with ESC key).
boot-args	null string	default boot arguments (e.g. -is).
boot-device	sd0a	Device from which to boot.
boot-directory	/os	Filesystem directory containing boot-file.
boot-file	mach	File to boot (if 'mach', also load server).
cluster-boot?	false	If true, each hypemode is independent.
diag-device	none	Diagnostic boot source device.
diag-file	none	File to boot in diagnostic mode.
diag-switch?	false	If true, run in diagnostic mode.
fcode-debug?	false	If true, include name fields for plug-in device FCodes.
input-device	no default	OBP keyboard input device. /console is the driver for the SPP console Xterm
internet-id	0.0.0.0	32-bit Internet ID # for network interface.
last-hardware-update	no default	System update information.
load-tunables	true	If true, the tunables file is loaded.
local-mac-address?	false	If true, network drivers use their own MAC address, not system's.
log-device	no default	Power-on device for console-log kernel function (not 'event logging').
mfg-switch?	false	Reserved for future functionality.
nvrarc	empty	Contents of NVRAMRC.
output-device	no default	OBP display device. /console is the driver for the SPP console Xterm.
sbus-probe-list	U1SB1 ...	Which controllers to probe. Any or none of U1SB1, U1SB0, U0SB1 U0SB0. Clear it to null with the command: 'null\$ to sbus-probe-list'.
screen-#columns	80	Number of Console Xterm columns.
screen-#rows	24	Number of Console Xterm rows (lines).
security-#badlogins	no default	Number of incorrect security password attempts.

security-mode	none	Firmware security level (<i>none</i> , <i>full</i> , or <i>command</i>).
security-password	no default	Firmware security password (never displayed). <i>Do not set this directly.</i>
system-mac-address	no default	Network interface hardware address. <i>Do not set this directly.</i>
tz	CST6CDT...	Like the HP-UX TZ shell environment variable - Used by OBP date command.
uXsbY	null string	Name of special FCode driver to use for SBus controller in Unit X and SBus Y.
unit-map	no default	Logical unit table. <i>Do not set this directly.</i>
use-nvramrc?	false	If true, execute commands in NVRAMRC during system start-up.

Viewing and Changing Configuration Parameters

mkmap [-n] unit device	Define an SPP-UX logical-unit to physical-unit mapping. The mapping is a label for tape and network devices which cannot be labeled like disks. The -n switch is used to create a logical-unit property with no physical unit information; use it with network adaptors only.
password	Set the security mode password. You will be prompted for an 8 character password and then asked to verify it.
printenv	Display all current parameters and current default values (numbers are usually shown as decimal values). <i>printenv parameter</i> shows the current value of the named parameter.
rmmmap device	Undefine a mapping created with <i>mkmap</i> .
show-map [device unit]	Display the logical unit mapping for the named device or all entries with same logical unit number. Used by itself, displays all logical-unit mappings for the current hypernode.
setenv [+g -g] parameter value	Set the parameter to the given decimal or text value. +g sets the global attribute, -g clears it. The global attribute causes Exemplar OBP to automatically update the same option on every hypernode. (To set a numeric parameter using a hexadecimal number, the OBP word <i>to</i> may be used, for example: 'h# 30 screen-#rows'.)
set-default parameter	Reset the value of the named parameter to the factory default.
set-defaults	Reset parameter values to the factory defaults. (Note: This function takes about 60 seconds to complete).
set-mac-address hw-address	Set the 6-byte parameter, <i>system-mac-address</i> . The number should be input as 6 hexadecimal numbers, separated with ':' characters.

NVRAMRC Editor Commands

<code>nvalias alias device-path</code>	Store the command " <code>devalias alias device-path</code> " in NVRAMRC. (The alias persists until the <code>nvunalias</code> or <code>set-defaults</code> commands are executed.)
<code>nvedit</code>	Enter the NVRAMRC editor. If data remains in the temporary buffer from a previous <code>nvedit</code> session, resume editing those previous contents. If not, read the contents of NVRAMRC into the temporary buffer and begin editing it.
<code>nvquit</code>	Discard the contents of the temporary buffer, without writing it to NVRAMRC. Prompt for confirmation.
<code>nvrecover</code>	Recover the contents of NVRAMRC if they have been lost as a result of the execution of <code>set-defaults</code> ; then enter the editor as with <code>nvedit</code> . <code>nvrecover</code> fails if <code>nvedit</code> is executed between the time that the NVRAMRC contents were lost and the time that <code>nvrecover</code> is executed.
<code>nvrn</code>	Execute the contents of the temporary buffer.
<code>nvstore</code>	Copy the contents of the temporary buffer to NVRAMRC; discard the contents of the temporary buffer.
<code>nvunalias alias</code>	Delete the corresponding alias from NVRAMRC.

Editor Commands (for Command Lines and NVRAMRC)

	Prev. Line	Beg. Line	Prev. Word	Prev. Char	Next Char	Next Word	End Line	Next Line
Move	<code>^P</code>	<code>^A</code>	<code>IB</code>	<code>^B</code>	<code>^F</code>	<code>IF</code>	<code>^E</code>	<code>^N</code>
Delete		<code>^U</code>	<code>^W</code>	<code>Del</code>	<code>^D</code>	<code>ID</code>	<code>^K</code>	
		Retype line		<code>^R</code>				
		Show all lines		<code>^L</code>				
		Paste after <code>^K</code>		<code>^Y</code>				
		Complete command		<code>^space</code>				
		Show all matches		<code>^/</code> or <code>^?</code> or <code>^}</code>				

! = Press and release Escape key first; ^ = Press and hold Control key

Using the NVRAMRC Editor

```
ok nvedit
:
(use editor commands)
:
^~c (get back to ok prompt)
ok nvstore (save changes)
ok setenv use-nvramrc? true (enable NVRAMRC)
```

Numeric Usage and Stack Comments

- Numeric I/O defaults to hexadecimal.
- Switch to decimal with `decimal`, switch to hexadecimal with `hex`.
- Use `10 .d` to see which base is currently active.

A numeric stack is used for all numeric parameters. Typing any integer puts that value on top of the stack. (Previous values are "pushed" down.) The right-hand item in a set always indicates the topmost stack item.

- The command `."` removes and displays the top stack value.
- The command `.s` non-destructively shows the entire stack contents.

A stack comment such as (n1 n2 -- n3) or (adr len --) or (--) listed after each command name shows the effect on the stack of executing that command. Items *before* the -- are used by the command and removed from the stack. These items *must* be present on the stack *before* the command can properly execute. Items *after* the -- are left on the stack after the command completes execution, and are available for use by subsequent commands.

	Alternate stack results. Example: (input -- adr len false result true).
?	Unknown stack items (changed from ???).
???	Unknown stack items.
acf	Code field address.
adr	Memory address (generally a virtual address, space id 0 only).
adr16	Memory address, must be 16-bit aligned.
adr32	Memory address, must be 32-bit aligned.
adr64	Memory address, must be 64-bit aligned.
byte bxxx	8-bit value (least significant byte in a 32-bit word).
char	7-bit value (least significant byte), high bit unspecified.
cnt/len/size	Count or length.
flag xxx?	0 = false; any other value = true (usually -1).
long Lxxx	32-bit value.
n n1 n2 n3	Normal signed values (32-bit).
+n u	Unsigned, positive values (32-bit).
n[64] or (n.low n.hi)	Extended-precision (64-bit) numbers (2 stack items).
phys	Physical address (actual hardware address).
pstr	Packed string (adr len means unpacked string).
virt	Virtual address (address used by software).
word wxxx	16-bit value (smallest two bytes in a 32-bit word).

Changing the Number Base

binary	(-)	Set the number base to 2.
decimal	(--)	Set the number base to 10.
d# number	(- n)	Interpret the next number in decimal.
octal	(-)	Set the number base to 8.
hex	(--)	Set the number base to 16.
h# number	(- n)	Interpret the next number in hex; base is unchanged.
.d	(n -)	Display n in decimal without changing base.
.h	(n -)	Display n in hex without changing base.

Basic Number Display

.	(n --)	Display a number in the current base.
.s	(-)	Display contents of data stack.
showstac k	(--)	Execute .s automatically before each ok prompt.

Stack Manipulation Commands

-rot	(n1 n2 n3 - n3 n1 n2)	Inversely rotate three stack items.
>r	(n -)	Move a stack item to the return stack. (Use with caution.)
?dup	(n - n n 0)	Duplicate the top stack item if nonzero.
2drop	(n1 n2 -)	Remove two items from the stack.
2dup	(n1 n2 - n1 n2 n1 n2)	Duplicate two stack items.
2over	(n1 n2 n3 n4 - n1 n2 n3 n4 n1 n2)	Copy second two stack items.
2swap	(n1 n2 n3 n4 - n3 n4 n1 n2)	Exchange two pairs of stack items.
clear	(??? -)	Empty the stack.
depth	(??? - ??? +n)	Return the number of items on the stack.
drop	(n -)	Remove the top item from the stack.
dup	(n - n n)	Duplicate the top stack item.
nip	(n1 n2 - n2)	Discard the second stack item.
over	(n1 n2 - n1 n2 n1)	Copy the second stack item to the top of the stack.
pick	(??? +n - ??? n2)	Copy +n-th stack item (1 pick = over).
r>	(- n)	Move a return stack item to the stack. (Use with caution.)
r@	(- n)	Copy the top of the return stack to the stack.
roll	(??? +n - ?)	Rotate +n stack items (2 roll = rot).
rot	(n1 n2 n3 - n2 n3 n1)	Rotate three stack items.
swap	(n1 n2 - n2 n1)	Exchange the top two stack items.
tuck	(n1 n2 - n2 n1 n2)	Copy the top stack item below the second item.

Arithmetic Functions

*	(n1 n2 - n3)	Multiply n1 * n2.
+	(n1 n2 - n3)	Add n1 + n2.
-	(n1 n2 - n3)	Subtract n1 - n2
/	(n1 n2 - quot)	Divide n1 / n2; remainder is discarded.
<<	(n1 +n - n2)	Left-shift n1 by +n bits.
>>	(n1 +n - n2)	Right-shift n1 by +n bits.
>>a	(n1 +n - n2)	Arithmetic right-shift n1 by +n bits.
abs	(n - u)	Absolute value.
and	(n1 n2 - n3)	Bitwise logical AND.
bounds	(startadr len - endadr startadr)	Convert startadr len to endadr startadr for do loop.

bjoin	(b.low b2 b3 b.hi – long)	Join four bytes to form a 32-bit longword.
bwjoin	(b.low b.hi – word)	Join two bytes to form a 16-bit word.
lbsplit	(long – b.low b2 b3 b.hi)	Split a 32-bit longword into four bytes.
lwsplit	(long – w.low w.hi)	Split a 32-bit longword into two 16-bit words.
max	(n1 n2 – n3)	n3 is maximum of n1 and n2.
min	(n1 n2 – n3)	n3 is minimum of n1 and n2.
mod	(n1 n2 – rem)	Remainder of n1 / n2.
negate	(n1 – n2)	Change the sign of n1.
not	(n1 – n2)	Bitwise ones complement.
or	(n1 n2 – n3)	Bitwise logical OR.
wbsplit	(word – b.low b.hi)	Split 16-bit word into two bytes.
wljoin	(w.low w.hi – long)	Join two words to form a longword.
xor	(n1 n2 – n3)	Bitwise exclusive OR.

Memory Access Commands

l	(n adr32 –)	Store a 32-bit number at adr32, must be 32-bit aligned.
+l	(n adr32 –)	Add n to the 32-bit number stored at adr32, must be 32-bit aligned.
@	(adr8 – n)	Fetch a 32-bit number from adr8, no alignment restriction.
cl	(n adr –)	Store low byte of n at adr.
c@	(adr – byte)	Fetch a byte from adr.
cpeek k	(adr – false byte true)	Fetch the byte at adr. Return the data and true if the access was successful. Return false if a read access error occurred. (Also lpeek, wpeek.)
cpoke e	(byte adr – okay?)	Store the byte to adr. Return true if the access was successful. Return false if a write access error occurred. (Also lpoke, wpoke.)
comp	(adr1 adr2 len – n)	Compare two byte arrays, n = 0 if arrays are identical, n = 1 if first byte that is different is greater in array#1, n = -1 otherwise.
diff	(adr1 adr2 len – 0 adr1'adr2')	Compare two byte arrays. Return 0 if arrays are identical, or the addresses of the first differing bytes.
dump	(adr len –)	Display len bytes of memory starting at adr. (Some I/O space cannot be dumped because byte access is used, which is invalid for some CSRs).
fill	(adr size byte –)	Set size bytes of memory to byte.
ll	(n adr32 –)	Store a 32-bit number at adr32.
ll@	(adr32 – long)	Fetch a 32-bit number from adr32.

move	(adr1 adr2 u --)	Copy u bytes from adr1 to adr2, handle buffer overlap properly.
w!	(n adr16 --)	Store a 16-bit number at adr16, must be 16-bit aligned.
w@	(adr16 -- word)	Fetch a 16-bit number from adr16, must be 16-bit aligned.

Memory Allocation

The Convex Exemplar version of OBP contains no virtual memory mapping subsystem; however, heap memory may be allocated from a memory pool initially mapped only once at boot time with a single block-tlb. The mapping is equivalent, i.e. virtual=physical. If a memory allocation fails, there is no pool from which to extend the heap. (That is, the internal OBP word 'alloc-more' is not implemented) For diagnostic and/or test client programs, `allocate-physical` is provided. This word allocates memory from physical memory that is not mapped by the block-tlb. It can be accessed by the program only if the the PSW C and D bits are turned off. The words `translation-off` and `translation-on` may be used to accomplish switching between the two states.

alloc-mem	(size -- virt)	Allocate size bytes of available memory; return the virtual address. Free it with <code>free-mem</code> .
free-mem	(virt size --)	Free memory allocated by <code>alloc-mem</code> .
pagesize	(-- size)	Return the size of a page (often 4K).

Defining Words

: <i>name</i>	(--) Usage: (??? -- ?)	Start creating a new colon definition.
;	(--)	Finish creating a new colon definition.
buffer: <i>name</i>	(size --) Usage: (-- adr64)	Create a named array in temporary storage.
c;	(--)	Finish creating a new code definition.
constant <i>name</i>	(n --) Usage: (-- n)	Define a constant (for example, 3 constant bar).
code <i>name</i>	(--) Usage: (--)	Start creating a new Forth assembler code-word definition.
create <i>name</i>	(--) Usage: (-- adr32)	Generic defining word.
defer <i>name</i>	(--) Usage: (??? -- ?)	Define forward reference or execution vector.
does>	(-- adr32)	Start the run-time clause for defining words.
value <i>name</i>	(n --) Usage: (-- n)	Create a changeable, named 32-bit quantity.
variable <i>name</i>	(--) Usage: (-- adr32)	Define a variable. (A variable is a pointer to a 4-byte word).

Dictionary Searching Commands

' <i>name</i>	(-- acf)	Find the named word in the dictionary. (Returns the code field address. Use outside definitions.)
[] <i>name</i>	(-- acf)	Similar to <code>'</code> but is used either inside or outside definitions.
.calls	(acf --)	Display a list of all words that call the word whose compilation address is <code>acf</code> .

\$find	(adr len -- adr len false acf n)	Find a word. n = 0 if not found, n = 1 if immediate, n = -1 otherwise.
see <i>thisword</i>	(-)	Decompile the named command.
(see)	(acf -)	Decompile the word indicated by the code field address.
sifting <i>ccc</i>	(-)	Display names of all dictionary entries containing the sequence of characters. <i>ccc</i> contains no spaces.
words	(-)	Display all visible words in the dictionary.

Dictionary Compilation Commands

.	(n -)	Place a number in the dictionary.
c,	(byte -)	Place a byte in the dictionary.
w,	(word -)	Place a 16-bit number in the dictionary.
l,	(long -)	Place a 32-bit number in the dictionary.
allot	(n -)	Allocate n bytes in the dictionary.
forget <i>name</i>	(-)	Remove word from dictionary and all subsequent words.
here	(- adr)	Address of top of dictionary.
is <i>name</i>	(n -)	Install a new action in a defer word or value.
patch <i>new-word</i> <i>old-word word-to-patch</i>	(-)	Replace <i>old-word</i> with <i>new-word</i> in <i>word-to-patch</i> .
(patch	(new-n old-n acf -)	Replace <i>old-n</i> with <i>new-n</i> in word indicated by <i>acf</i> .

Controlling Text Input

(<i>ccc</i>)	(-)	Begin a comment.
\ <i>rest-of-line</i>	(-)	Skip the rest of the line.
ascii <i>ccc</i>	(- char)	Get numerical value of first ASCII character of next word.
key	(- char)	Read a character from the assigned input device's keyboard.
key?	(- flag)	True if a key has been typed on the input device's keyboard.

Displaying Text Output

cr	(-)	Terminate a line on the display and go to the next line.
diagcr	(-)	Clear the diagnostic LCD.
diag!	(char -)	Display the character on the diagnostic LCD.
emit	(char -)	Display the character on the console.
type	(adr +n -)	Display n characters on the console.

Manipulating Text Strings

"ccc"	(- adr len)	Collect an input stream string, either interpreted or compiled. Within the string, use "(00,ff...)" to include arbitrary byte values.
."ccc"	(-)	Compile a string for later display.
bl	(- char)	ASCII code for the space character; decimal 32.
\$cat	(adr len pstr -)	Append adr,len to the end of pstr.
count	(pstr - adr +n)	Unpack a packed string.
p"ccc"	(- pstr)	Collect a string from the input stream; store as a packed string.
pack	(adr len pbuf - pstr)	Store a adr,len string as a packed string.

Comparison Commands

<	(n1 n2 - flag)	True if n1 < n2.
<=	(n1 n2 - flag)	True if n1 <= n2.
<>	(n1 n2 - flag)	True if n1 <> n2.
=	(n1 n2 - flag)	True if n1 = n2.
>	(n1 n2 - flag)	True if n1 > n2.
>=	(n1 n2 - flag)	True if n1 >= n2.
between	(n min max - flag)	True if min <= n <= max.
u<	(u1 u2 - flag)	True if u1 < u2, unsigned.
u<=	(u1 u2 - flag)	True if u1 <= u2, unsigned.
u>	(u1 u2 - flag)	True if u1 > u2, unsigned.
u>=	(u1 u2 - flag)	True if u1 >= u2, unsigned.
within	(n min max - flag)	True if min <= n < max.

if-then-else Commands

else	(-)	Execute the following code if if failed.
if	(flag -)	Execute the following code if flag is true.
then	(-)	Terminate if...then...else.

begin (Conditional) Loop Commands

again	(-)	End a begin...again infinite loop.
begin	(-)	Begin a begin...while...repeat, begin...until, or begin...again loop.
repeat	(-)	End a begin...while...repeat loop.
until	(flag -)	Continue executing a begin...until loop until flag is true.
while	(flag -)	Continue executing a begin...while...repeat loop while flag is true.

do (Counted) Loop Commands

+loop	(n -)	End a do...+loop construct; add n to loop index and return to do (if n < 0, index goes from start to end inclusive).
?do	(end start -)	Begin ?do...loop to be executed 0 or more times. Index goes from start to end-1 inclusive. If end = start, loop is not executed.
do	(end start -)	Begin a do...loop. Index goes from start to end-1 inclusive. Example: 10 0 do i . loop (prints 0 1 2...d e f).
i	(- n)	Loop index.
j	(- n)	Loop index for next enclosing loop.
leave	(-)	Exit from do...loop.
loop	(-)	End of do...loop.

case Statement

```
( value )
case
2 of ." it was two" endof
0 of ." it was zero" endof
." it was " dup . (optional default clause)
endcase
```

Cache Manipulation Commands

flush-cache	(adr size -)	Flush instruction and data cache.
-------------	----------------	-----------------------------------

Multiprocessor Commands

module-info	(-)	Display type and speed of CPU modules.
cpu	(n -)	Switch current CPU.
node	(n -)	Switch current hypemode.

Program Execution Control Commands

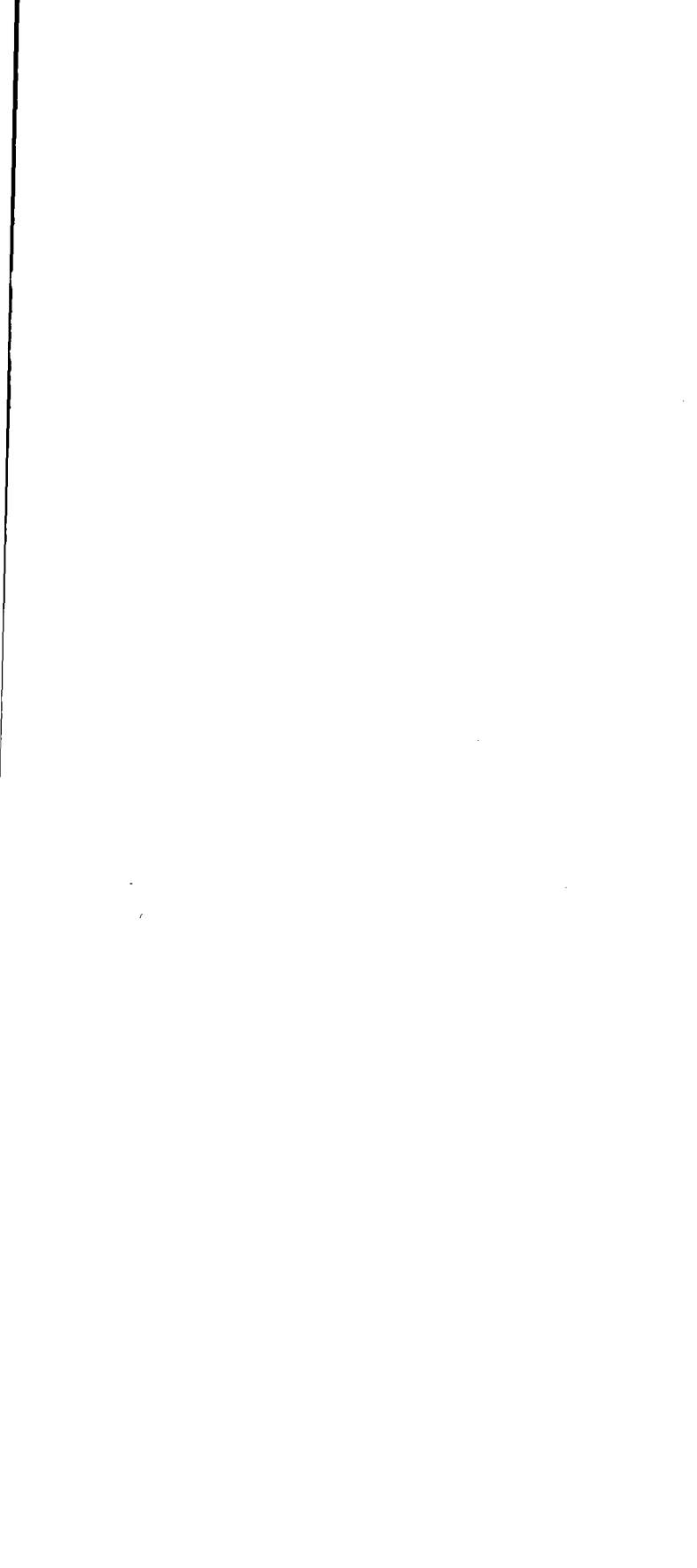
abort	(-)	Abort current execution and interpret keyboard commands.
abort" ccc"	(abort? -)	If flag is true, abort and display message.
eval	(adr len -)	Interpret Forth source from an array.
execute	(acf -)	Execute the word whose code field address is on the stack.
exit	(-)	Return from the current word. (Cannot be used in counted loops.)
quit	(-)	Same as abort, but leave stack intact.

PA-RISC Forth Virtual Machine Register Usage

Forth	PA-RISC	Function in the PA-RISC Forth virtual machine
rip	%r2	Branch return link; used by inner interpreter.
sp	%r3	Parameter stack pointer.
bases	%r4	Base, right shifted 3 bits for 'bv' branches.
ip	%r5	Interpreter pointer.
base	%r6	Base address of dictionary.
rp	%r7	Return stack pointer.
up	%r8	Base address User Area.

Forth	PA-RISC	Function in the PA-RISC Forth virtual machine
tos	%r9	Contents of Top of Stack.
w	%r18	Code field address of word being executed.
t0-t7	%r10-%r17	Temp. registers. Not saved across words.

© 1994, Convex Computer Corporation, Inc.—Printed in the United States of America.
 Convex, Exemplar, SPP1000/XA, SPP1000/CD, are trademarks or registered trademarks of Convex Computer Corporation, Inc. Sun, Sun Microsystems, the Sun logo, and OpenBoot are trademarks or registered trademarks of Sun Microsystems, Inc. THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS.



ORDER NUMBER
DSW-854

DOCUMENT NUMBER
710-029430-000

CONVEX
PRESS

