# Getting Started with Seamless CVE

Software Version 4.3

changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Mentor Graphics to determine whether any changes have been made.

The terms and conditions governing the sale and licensing of Mentor Graphics products are set forth in written agreements between Mentor Graphics and its customers. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Mentor Graphics whatsoever.

See the separate "End User License Agreement" section.

MENTOR GRAPHICS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

MENTOR GRAPHICS SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF MENTOR GRAPHICS CORPORATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

**RESTRICTED RIGHTS LEGEND 03/97**

U.S. Government Restricted Rights. The SOFTWARE and documentation have been developed entirely at private expense and are commercial computer software provided with restricted rights. Use, duplication or disclosure by the U.S. Government or a U.S. Government subcontractor is subject to the restrictions set forth in the license agreement provided with the software pursuant to DFARS 227.7202-3(a) or as set forth in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19, as applicable.

Contractor/manufacturer is:
Mentor Graphics Corporation
8005 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777.



A complete list of trademark names appears in the "Trademark Information" section.

# Trademark Information

## Mentor Graphics Trademarks

The following names are trademarks, registered trademarks, and service marks of Mentor Graphics Corporation:

3D Design™, A World of Learning℠, ABIST™, Arithmetic BIST™, AccuPARTner™, AccuParts™, AccuSim®, ADEPT™, ADVance™ MS, ADVance™ RFIC, AMPLE™, Analog Analyst™, Analog Station™, AppNotes℠, ARTgrid™, ArtRouter™, ARTshape™, ASICPlan™, ASICVector Interfaces™, Aspire™ Assess2000℠, AutoActive®, AutoCells™, AutoDissolve™, AutoFilter™, AutoFlow™, AutoLib™, AutoLinear™, AutoLink™, AutoLogic™, AutoLogic BLOCKS™, AutoLogic FPGA™, AutoLogic VHDL®, AutomotiveLib™, AutoPAR®, AutoTherm®, AutoTherm Duo™, AutoThermMCM™, AutoView™, Autowire Station™, AXEL™, AXEL Symbol Genie™, BISTArchitect™, BIST Compiler℠, BIST-In-Place℠, BIST-Ready℠, Board Architect™, Board Designer™, Board Layout™, Board Link™, Board Process Library™, Board Station®, Board Station Consumer™, BOLD Administrator™, BOLD Browser™, BOLD Composer™, BSDArchitect™, BSPBuilder™, Buy on Demand™, Cable Analyzer™, Cable Station™, CAECO Designer™, CAEFORM™, Calibre®, Calibre CB™, Calibre DESIGNrev™, Calibre DRC™, Calibre DRC-H™, Calibre FRACTUREh™, Calibre FRACTUREj™, Calibre FRACTUREk™, Calibre FRACTUREm™, Calibre FRACTUREt™, Calibre Interactive™, Calibre LITHOview™, Calibre LVS™, Calibre LVS-H™, Calibre MDPview™, Calibre MGC™, Calibre OPCpro™, Calibre OPCsbar™, Calibre ORC™, Calibre PRINTimage™, Calibre PSMgate™, Calibre PSMcheck™, Calibre RVE™, Calibre TDopc™, Calibre WORKbench™, Calibre xRC™, CAM Station™, Capture Station®, CAPITAL™, CAPITAL Analysis™, CAPITAL Bridges™, CAPITAL Documents™, CAPITAL H™, CAPITAL Harness™, CAPITAL Harness Systems™, CAPITAL H the complete desktop engineer®, CAPITAL Insight™, CAPITAL Integration™, CAPITAL Manager™, CAPITAL Manufacturer™, CAPITAL Support™, CAPITAL Systems™, Cell Builder™, Cell Station®, CellFloor™, CellGraph™, CellPlace™, CellPower™, CellRoute™, Centricity™, CEOC™, ChaseX™, CheckMate™, CHEOS™, Chip Station®, ChipGraph™, CommLib™, CommLib BMC™, Concurrent Board Process℠, Concurrent Design Environment™, Connectivity Dataport™, Continuum™, Continuum Power Analyst™, CoreAlliance™, CoreBIST™, Core Builder™, Core Factory™, Co-Verification Environment™, CTIntegrator™, DataCentric Model™, DataFusion™, Datapath™, Data Solvent™, dBUG™, Debug Detective™, DC Analyzer™, Design Architect®, Design Architect Elite™, DesignBook®, Design Capture™, Design Manager™, Design Station®, DesignView™, DesktopASIC™, Destination PCB®, DFTAdvisor™, DFTArchitect™, DFTInsight™, DirectConnect℠, DSV™, Direct System Verification™, Documentation Station™, DSS (Decision Support System)™, DSV™, E3LCable™, ECO Immunity℠, EDGE (Engineering Design Guide for Excellence)℠, EDT™, Eldo™, EldoNet™, ePartners™, EParts®, Empowering Solutions™, Engineer's Desktop™, EngineerView™, ENRead™, ENWrite™, ESim™, Exemplar™, Exemplar Logic™, Expedition™, Expert2000℠, Explorer CAECO Layout™, Explorer CheckMate™, Explorer Datapath™, Explorer Lsim™, Explorer Lsim-C™, Explorer Lsim-S™, Explorer Ltime™, Explorer Schematic™, Explorer VHDLsim™, ExpressI/O™, FabLink™, Falcon®, Falcon Framework®, FastScan™, FastStart™, FastTrack Consulting℠, First-Pass Design Success™, First-Pass success℠, FlexSim™, FlexTest™, FDL (Flow Definition Language)™, FlowTabs™, FlowXpert™, FORMA™, FormalPro™, FPGA Advantage®, FPGAdvisor™, FPGA BoardLink™, FPGA Builder™, FPGASim™, FPGA Station®, FrameConnect™, Galileo®, Gate Station®, GateGraph™, GatePlace™, GateRoute™, GDT®, GDT Core®, GDT Designer™, GDT Developer™, GENIE™, GenWare™, Geom Genie™, HDL2Graphics™, HDL Architect™, HDL Architect Station™, HDL Author™, HDL Designer™, HDL Designer Series™, HDL Detective™, HDL Inventor™, HDL Pilot™, HDL Processor™, HDL Sim™, HDLWrite™,Hardware Modeling Library™, HIC rules™, Hierarchical Injection™, Hierarchy Injection™, HotPlot®, Hybrid Designer™, Hybrid Station®, IBD™, IC Design Station™, IC Designer™, IC Layout Station™, IC Station®, ICbasic™, ICblocks™, ICcheck™, ICcompact™, ICdevice™, ICextract™, ICGen™, ICgraph™, ICLink™, IClister™, ICplan™, ICRT Controller Lcompiler™, ICrules™, ICtrace™, ICverify™, ICview™, ICX™, ICX Active™, ICX Custom Model™, ICX Custom Modeling™, ICX Plan™, ICX Pro™, ICX Project Modeling™, ICX Sentry™, ICX Standard Library™, ICX Verify™, ICX Vision™, IDEA Series™, Idea Station®, INFORM®, IFX™, Inexia™, Integrated Product Development®, Integra Station™, Integration Tool Kit™, INTELLITEST®, Interactive Layout™, Interconnect Table™, Interface-Based Design™, IntraStep℠, Inventra™, InventraIPX™, Inventra Soft Cores™, IP Engine ™, IP Evaluation Kit™, IP Factory™, IP -PCB™, IP QuickUse™, IPSim™, IS_Analyzer™, IS_Floorplanner™, IS_MultiBoard™, IS_Optimizer™, IS_Synthesizer™, ISD Creation℠, ITK™, It's More than Just Tools℠, Knowledge Center℠, Knowledge-Sourcing℠, LAYOUT™, LNL™, LBIST™, LBISTArchitect™, Language Neutral Licensing™, Lc™, Lcore™, Leaf Cell Toolkit™, Led™, LED LAYOUT™, Leonardo®, LeonardoInsight™, LeonardoSpectrum™, LIBRARIAN™, Library Builder™, Logic Analyzer on a Chip℠, Logic Builder™, Logical Cable™, LogicLib™, *logio*™, Lsim™, Lsim DSM™, Lsim-Gate™, Lsim Net™, Lsim Power Analyst™, Lsim-Review™, Lsim-Switch™, Lsim-XL™, Mach PA™, Mach TA™, Manufacture View™, Manufacturing Advisor™, Manufacturing Cable™, MaskCompose™, MaskPE®, MBIST™, MBISTArchitect™, MBIST Full-Speed™, MBIST Flex™, MBIST Manager™, MCM Designer™, MCM Station®, MDV™, MegaFunction™, Memory Builder™, Memory Builder Conductor™, Memory Builder Mozart™, Memory Designer™, Memory Model Builder™, Mentor™, Mentor Graphics®, Mentor Graphics Support CD℠, Mentor Graphics SupportBulletin℠, Mentor Graphics SupportCenter℠, Mentor Graphics SupportFax℠, Mentor Graphics SupportNet-Email℠, Mentor Graphics SupportNet-FTP℠, Mentor Graphics SupportNet-Telnet℠, Mentor Graphics We Mean Business™, MicroPlan™, MicroRoute™, Microtec®, Mixed-Signal Pro™, ModelEditor™, Model*Sim*®, Model*Sim* LNL™, Model*Sim* VHDL™, Model*Sim* VLOG™, Model*Sim* SE™, ModelStation®, Model Technology™, ModelViewer™, ModelViewer*Plus*™, *MODGEN*™, Monet®, Mslab™, Msview™, MS Analyzer™, MS Architect™, MS-Express™, MSIMON™, MTPI℠, Nanokernel®, NetCheck™, NETED™, Nucleus™, Online Knowledge Center℠, OpenDoor℠, Opsim™, OutNet™, P&RIntegrator™, PACKAGE™, PARADE™, ParallelRoute-Autocells™, ParallelRoute-MicroRoute™, PathLink™, Parts SpeciaList™, PCB-Gen™, PCB-Generator™, PCB IGES™, PCB Mechanical Interface™, PDLSim™, Personal Learning Program™, Physical Cable™, Physical Test Manager:SITE™, PLA Lcompiler™, Platform Express™, PLDSynthesis™, PLDSynthesis II™, Power Analyst™, PowerAnalyst Station™, Power To Create®, Precision™, Precision Synthesis™, Precision HLS™, Precision PNR™, Precision PTC™, Pre-Silicon™, ProjectXpert™, ProtoBoard™, ProtoView™, QNet™, QualityIBIS™, QuickCheck™, QuickConnect™, QuickFault™, QuickGrade™, QuickHDL™, QuickHDL Express™, QuickHDL Pro™, QuickPart Builder™, QuickPart Tables™, QuickParts™, QuickPath™, QuickSim™, QuickSimII™, QuickStart™, QuickUse™, QuickVHDL®, RAM Lcompiler™, RC-Delay™, RC-Reduction™, RapidExpert™, REAL Time Solutions!™, Registrar™, Reinstatement 2000℠, Reliability Advisor™, Reliability Manager™, REMEDI™, Renoir™, RF Architect™, RF Gateway™, RISE™, ROM Lcompiler™, RTL X-Press™, Satellite PCB Station™, ScalableModels™, Scaleable Verification™, SCAP™, Scan-Sequential™, Scepter™, Scepter DFF™, Schematic View Compiler, SVC™, Schemgen™, SDF™ (Software Data Formatter), SDL2000 Lcompiler™, Seamless®, Seamless C-Bridge™, Seamless Co-Designer™, Seamless CVE™, Seamless Express™, Selective Promotion™, SignaMask OPC™, Signal Spy™, Signal Vision™, Signature Synthesis™, Simulation Manager™, SimExpress™, SimPilot™, SimView™, SiteLine2000℠, SmartMask™, SmartParts™, SmartRouter™, SmartScripts™, Smartshape™, SNX™, SneakPath Analyzer™, SOS Initiative™, Source Explorer™, SpeedGate™, SpeedGate DSV™, SpiceNet™, SST Velocity®, Standard Power Model Format (SPMF)™, Structure Recovery™, Super C™, Super IC Station™, Support Services BaseLine℠, Support Services ClassLine℠, Support Services Latitudes℠, Support Services OpenLine℠, Support Services PrivateLine℠, Support Services SiteLine℠, Support Services TechLine℠, Support Services RemoteLine℠, Symbol Genie™, Symbolscript™, SYMED™, SynthesisWizard™, System Architect™, System Design Station™, System Modeling Blocks™, Systems on Board Initiative™, System Vision™, Target Manager™, Tau®, TeraCell™, TeraPlace™, TeraPlace-GF™, TechNotes™, The Ultimate Tool for HDL Simulation™, TestKompress™, Test Station®, Test Structure Builder™, The Ultimate Site For HDL Simulation™, TimeCloser™, Timing Builder™, TNX™, ToolBuilder™, TrueTiming™, Vlog™, V-Express™, V-Net™, VHDLnet™, VHDLwrite™, Verinex™, ViewCreator™, ViewWare®, Virtual Library™, Virtual Target™, Virtual Test Manager:TOP™, VR-Process℠, VRTX®, VRTXmc™, VRTXoc™, VRTXsa™, VRTX32®, Waveform DataPort™, We Make TMN Easy™, Wiz-o-matic™, WorkXpert™, xCalibre™, xCalibrate™, Xconfig™, XlibCreator™, Xpert™, Xpert API™, XpertBuilder™, Xpert Dialogs™, Xpert Profiler™, XRAY®, XRAY MasterWorks®, XSH®, Xtrace®, Xtrace Daemon™, Xtrace Protocol™, Zeelan®, Zero Tolerance Verification™, Zlibs™

## Third-Party Trademarks

The following names are trademarks, registered trademarks, and service marks of other companies that appear in Mentor Graphics product publications:

Adobe, the Adobe logo, Acrobat, the Acrobat logo, Exchange, FrameMaker, FrameViewer, PostScript,and Reader are registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Altera, ByteBlaster, Excalibur, and Quartus are trademarks or registered trademarks of Altera Corporation in the United States and other countries.

AM188, AMD, AMD-K6, and AMD Athlon Processor are trademarks of Advanced Micro Devices, Inc.

Apple and Laserwriter are registered trademarks of Apple Computer, Inc.

ARIES is a registered trademark of Aries Technology.

AMBA, ARM, ARMulator, ARM7TDMI, ARM7TDMI-S, ARM9TDMI, ARM9E-S, ARM946E-S, ARM966E-S, EmbeddedICE, StrongARM, TDMI, and Thumb are trademarks or registered trademarks of ARM Limited.

ASAP, Aspire, C-FAS, CMPI, Eldo-FAS, EldoHDL, Eldo-Opt, Eldo-UDM, EldoVHDL, Eldo-XL, Elga, Elib, Elib-Plus, ESim, Fidel, Fideldo, GENIE, GENLIB, HDL-A, MDT, MGS-MEMT, MixVHDL, Model Generator Series (MGS), Opsim, SimLink, SimPilot, SpecEditor, Success, SystemEldo, VHDeLDO and Xelga are registered trademarks of ANACAD Electrical Engineering Software, a unit of Mentor Graphics Corporation.

Avant! and Star-Hspice are trademarks of Avant! Corporation.

AVR is a registered trademark of Atmel Corporation.

Cadence, Affirma signalscan, Allegro, Analog Artist, Composer, Concept, Design Planner, Dracula, GDSII, GED, HLD Systems, Leapfrog, Logic DP, NC-Verilog, OCEAN, Physical DP, Pillar, Silicon Ensemble, Spectre, Verilog, Verilog XL, Veritime, and Virtuoso are trademarks or registered trademarks of Cadence Design Systems, Inc.

CAE+Plus and ArchGen are registered trademarks of Cynergy System Design.

CalComp is a registered trademark of CalComp, Inc.

Canon is a registered trademark of Canon, Inc. BJ-130, BJ-130e, BJ-330, and Bubble Jet are trademarks of Canon, Inc.

Centronics is a registered trademark of Centronics Data Computer Corporation.

ColdFire and M-Core are registered trademarks of Motorola, Inc.

Ethernet is a registered trademark of Xerox Corporation.

Foresight and Foresight Co-Designer are trademarks of Nu Thena Systems, Inc.

FLEXlm is a trademark of Globetrotter Software, Inc.

GenCAD is a trademark of Teradyne Inc.

Hewlett-Packard (HP), LaserJet, MDS, HP-UX, PA-RISC, APOLLO, DOMAIN and HPare registered trademarks of Hewlett-Packard Company.

HCL-eXceed and HCL-eXceed/W are registered trademark of Hummingbird Communications. Ltd.

HyperHelp is a trademark of Bristol Technology Inc.

Installshield is a registered trademark and service mark of InstallShield Corporation.

IBM, PowerPC, and RISC Systems/6000 are trademarks of International Business Machines Corporation.

I-DEAS and UG/Wiring are registered trademarks of Electronic Data Systems Corporation.

IKON is a trademark of Tahoma Technology.

IKOS and Voyager are registered trademarks of IKOS Systems, Inc.

Imagen, QMS, QMS-PS 820, Innovator, and Real Time Rasterization are registered trademarks of MINOLTA-QMS Inc. imPRESS and UltraScript are trademarks of MINOLTA-QMS Inc.

ImageGear is a registered trademark of AccuSoft Corporation.

Infineon, TriCore, and C165 are trademarks of Infineon Technologies AG.

Intel, i960, i386, and i486 are registered trademarks of Intel Corporation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc.

Linux is a registered trademark of Linus Torvalds.

MemoryModeler MemMaker are trademarks of Denali Software, Inc.

MIPS is a trademark of MIPS Technologies, Inc.

MS-DOS, Windows 95, Windows 98, Windows 2000, and Windows NT are registered trademarks of Microsoft Corporation.

MULTI is a registered trademark of Green Hills Software, Inc.

NEC and NEC EWS4800 are trademarks of NEC Corp.

Netscape is a trademark of Netscape Communications Corporation.

Novas, Debussy, and nWave are trademarks or registered trademarks of Novas Software, Inc.

OakDSPCore is a registered trademark for DSP Group, Inc.

Oracle, Oracle8i, and SQL*Plus are trademarks or registered trademarks of Oracle Corporation.

OSE is a registered trademark of OSE Systems.

PKZIP is a registered trademark of PKWARE, Inc.

Pro/CABLING and HARNESSDESIGN are trademarks or registered trademarks of Parametric Technology Corporation.

Quantic is a registered trademark of Quantic EMC Inc.

QUASAR is a trademark of ASM Lithography Holding N.V.

Red Hat is a registered trademark of Red Hat Software, Inc.

SCO and the SCO logo are trademarks or registered trademarks of Caldera International, Inc.

Sneak Circuit Analysis Tool (SCAT) is a registered trademark of SoHaR Incorporated.

Updated 5/14/02

# End User License Agreement

**IMPORTANT - USE OF THIS SOFTWARE IS SUBJECT TO LICENSE RESTRICTIONS CAREFULLY READ THIS LICENSE AGREEMENT BEFORE USING THE SOFTWARE**

This license is a legal "Agreement" concerning the use of Software between you, the end-user, either individually or as an authorized representative of the company purchasing the license, and Mentor Graphics Corporation, Mentor Graphics (Ireland) Limited, Mentor Graphics (Singapore) Private Limited, and their majority-owned subsidiaries ("Mentor Graphics"). USE OF SOFTWARE INDICATES YOUR COMPLETE AND UNCONDITIONAL ACCEPTANCE OF THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT. If you do not agree to these terms and conditions, promptly return or, if received electronically, certify destruction of Software and all accompanying items within 10 days after receipt of Software and receive a full refund of any license fee paid

## END-USER LICENSE AGREEMENT

1. **GRANT OF LICENSE**. The software programs you are installing, downloading, or have acquired with this Agreement, including any updates, modifications, revisions, copies, and documentation ("Software") are copyrighted, trade secret and confidential information of Mentor Graphics or its licensors who maintain exclusive title to all Software and retain all rights not expressly granted by this Agreement. Mentor Graphics or its authorized distributor grants to you, subject to payment of appropriate license fees, a nontransferable, nonexclusive license to use Software solely: (a) (in machine-readable, object-code form; (b) for your internal business purposes; and (c) on the computer hardware or at the site for which an applicable license fee is paid, or as authorized by Mentor Graphics. A site is restricted to a one-half mile (800 meter) radius. Mentor Graphics' then-current standard policies, which vary depending on Software, license fees paid or service plan purchased, apply to the following and are subject to change: (a) relocation of Software; (b) use of Software, which may be limited, for example, to execution of a single session by a single user on the authorized hardware or for a restricted period of time (such limitations may be communicated and technically implemented through the use of authorization codes or similar devices); (c) eligibility to receive updates, modifications, and revisions; and (d) support services provided. Current standard policies are available upon request.

2. **ESD SOFTWARE**. If you purchased a license to use embedded software development (ESD) Software, Mentor Graphics or its authorized distributor grants to you a nontransferable, nonexclusive license to reproduce and distribute executable files created using ESD compilers, including the ESD run-time libraries distributed with ESD C and C++ compiler Software that are linked into a composite program as an integral part of your compiled computer program, provided that you distribute these files only in conjunction with your compiled computer program. Mentor Graphics does NOT grant you any right to duplicate or incorporate copies of Mentor Graphics' real-time operating systems or other ESD Software, except those explicitly granted in this section, into your products without first signing a separate agreement with Mentor Graphics for such purpose.

3. **BETA CODE**

   3.1. Portions or all of certain Software may contain code for experimental testing and evaluation ("Beta Code"), which may not be used without Mentor Graphics' explicit authorization. Upon Mentor Graphics' authorization, Mentor Graphics grants to you a temporary, nontransferable, nonexclusive license for experimental use to test and evaluate the Beta Code without charge for a limited period of time specified by Mentor Graphics. This grant and your use of the Beta Code shall not be construed as marketing or offering to sell a license to the Beta Code, which Mentor Graphics may choose not to release commercially in any form.

   3.2. If Mentor Graphics authorizes you to use the Beta Code, you agree to evaluate and test the Beta Code under normal conditions as directed by Mentor Graphics. You will contact Mentor Graphics periodically during your use of the Beta Code to discuss any malfunctions or suggested improvements. Upon completion of your evaluation and testing, you will send to Mentor Graphics a written evaluation of the Beta Code, including its strengths, weaknesses and recommended improvements.

3.3. You agree that any written evaluations and all inventions, product improvements, modifications or developments that Mentor Graphics conceives or makes during or subsequent to this Agreement, including those based partly or wholly on your feedback, will be the exclusive property of Mentor Graphics. Mentor Graphics will have exclusive rights, title and interest in all such property. The provisions of this subsection shall survive termination or expiration of this Agreement.

4. **RESTRICTIONS ON USE**. You may copy Software only as reasonably necessary to support the authorized use. Each copy must include all notices and legends embedded in Software and affixed to its medium and container as received from Mentor Graphics. All copies shall remain the property of Mentor Graphics or its licensors. You shall maintain a record of the number and primary location of all copies of Software, including copies merged with other software, and shall make those records available to Mentor Graphics upon request. You shall not make Software available in any form to any person other than your employer's employees and contractors, excluding Mentor Graphics' competitors, whose job performance requires access. You shall take appropriate action to protect the confidentiality of Software and ensure that any person permitted access to Software does not disclose it or use it except as permitted by this Agreement. Except as otherwise permitted for purposes of interoperability as specified by the European Union Software Directive or local law, you shall not reverse-assemble, reverse-compile, reverse-engineer or in any way derive from Software any source code. You may not sublicense, assign or otherwise transfer Software, this Agreement or the rights under it without Mentor Graphics' prior written consent. The provisions of this section shall survive the termination or expiration of this Agreement.

**5. LIMITED WARRANTY**

5.1. Mentor Graphics warrants that during the warranty period Software, when properly installed, will substantially conform to the functional specifications set forth in the applicable user manual. Mentor Graphics does not warrant that Software will meet your requirements or that operation of Software will be uninterrupted or error free. The warranty period is 90 days starting on the 15th day after delivery or upon installation, whichever first occurs. You must notify Mentor Graphics in writing of any nonconformity within the warranty period. This warranty shall not be valid if Software has been subject to misuse, unauthorized modification or installation. MENTOR GRAPHICS' ENTIRE LIABILITY AND YOUR EXCLUSIVE REMEDY SHALL BE, AT MENTOR GRAPHICS' OPTION, EITHER (A) REFUND OF THE PRICE PAID UPON RETURN OF SOFTWARE TO MENTOR GRAPHICS OR (B) MODIFICATION OR REPLACEMENT OF SOFTWARE THAT DOES NOT MEET THIS LIMITED WARRANTY, PROVIDED YOU HAVE OTHERWISE COMPLIED WITH THIS AGREEMENT. MENTOR GRAPHICS MAKES NO WARRANTIES WITH RESPECT TO: (A) SERVICES; (B) SOFTWARE WHICH IS LOANED TO YOU FOR A LIMITED TERM OR AT NO COST; OR (C) EXPERIMENTAL BETA CODE; ALL OF WHICH ARE PROVIDED "AS IS."

5.2. THE WARRANTIES SET FORTH IN THIS SECTION 5 ARE EXCLUSIVE. NEITHER MENTOR GRAPHICS NOR ITS LICENSORS MAKE ANY OTHER WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO SOFTWARE OR OTHER MATERIAL PROVIDED UNDER THIS AGREEMENT. MENTOR GRAPHICS AND ITS LICENSORS SPECIFICALLY DISCLAIM ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

6. **LIMITATION OF LIABILITY**. EXCEPT WHERE THIS EXCLUSION OR RESTRICTION OF LIABILITY WOULD BE VOID OR INEFFECTIVE UNDER APPLICABLE STATUTE OR REGULATION, IN NO EVENT SHALL MENTOR GRAPHICS OR ITS LICENSORS BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS) WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY, EVEN IF MENTOR GRAPHICS OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL MENTOR GRAPHICS' OR ITS LICENSORS' LIABILITY UNDER THIS AGREEMENT EXCEED THE AMOUNT PAID BY YOU FOR THE SOFTWARE OR SERVICE GIVING RISE TO THE CLAIM. IN THE CASE WHERE NO AMOUNT WAS PAID, MENTOR GRAPHICS AND ITS LICENSORS SHALL HAVE NO LIABILITY FOR ANY DAMAGES WHATSOEVER.

7. **LIFE ENDANGERING ACTIVITIES**. NEITHER MENTOR GRAPHICS NOR ITS LICENSORS SHALL BE LIABLE FOR ANY DAMAGES RESULTING FROM OR IN CONNECTION WITH THE USE OF SOFTWARE IN ANY APPLICATION WHERE THE FAILURE OR INACCURACY OF THE SOFTWARE MIGHT RESULT IN DEATH OR PERSONAL INJURY. YOU AGREE TO INDEMNIFY AND HOLD HARMLESS MENTOR GRAPHICS AND ITS LICENSORS FROM ANY CLAIMS, LOSS, COST, DAMAGE, EXPENSE, OR LIABILITY, INCLUDING ATTORNEYS' FEES, ARISING OUT OF OR IN CONNECTION WITH SUCH USE.

**8. INFRINGEMENT**

8.1. Mentor Graphics will defend or settle, at its option and expense, any action brought against you alleging that Software infringes a patent or copyright in the United States, Canada, Japan, Switzerland, Norway, Israel, Egypt, or the European Union. Mentor Graphics will pay any costs and damages finally awarded against you that are attributable to the claim, provided that you: (a) notify Mentor Graphics promptly in writing of the action; (b) provide Mentor Graphics all reasonable information and assistance to settle or defend the claim; and (c) grant Mentor Graphics sole authority and control of the defense or settlement of the claim.

8.2. If an infringement claim is made, Mentor Graphics may, at its option and expense, either (a) replace or modify Software so that it becomes noninfringing, or (b) procure for you the right to continue using Software. If Mentor Graphics determines that neither of those alternatives is financially practical or otherwise reasonably available, Mentor Graphics may require the return of Software and refund to you any license fee paid, less a reasonable allowance for use.

8.3. Mentor Graphics has no liability to you if the alleged infringement is based upon: (a) the combination of Software with any product not furnished by Mentor Graphics; (b) the modification of Software other than by Mentor Graphics; (c) the use of other than a current unaltered release of Software; (d) the use of Software as part of an infringing process; (e) a product that you design or market; (f) any Beta Code contained in Software; or (g) any Software provided by Mentor Graphics' licensors which do not provide such indemnification to Mentor Graphics' customers.

8.4. THIS SECTION 8 STATES THE ENTIRE LIABILITY OF MENTOR GRAPHICS AND ITS LICENSORS AND YOUR SOLE AND EXCLUSIVE REMEDY WITH RESPECT TO ANY ALLEGED PATENT OR COPYRIGHT INFRINGEMENT BY ANY SOFTWARE LICENSED UNDER THIS AGREEMENT.

9. **TERM**. This Agreement remains effective until expiration or termination. This Agreement will automatically terminate if you fail to comply with any term or condition of this Agreement or if you fail to pay for the license when due and such failure to pay continues for a period of 30 days after written notice from Mentor Graphics. If Software was provided for limited term use, this Agreement will automatically expire at the end of the authorized term. Upon any termination or expiration, you agree to cease all use of Software and return it to Mentor Graphics or certify deletion and destruction of Software, including all copies, to Mentor Graphics' reasonable satisfaction.

10. **EXPORT**. Software is subject to regulation by local laws and United States government agencies, which prohibit export or diversion of certain products, information about the products, and direct products of the products to certain countries and certain persons. You agree that you will not export in any manner any Software or direct product of Software, without first obtaining all necessary approval from appropriate local and United States government agencies.

11. **RESTRICTED RIGHTS NOTICE**. Software has been developed entirely at private expense and is commercial computer software provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the U.S. Government or a U.S. Government subcontractor is subject to the restrictions set forth in the license agreement under which Software was obtained pursuant to DFARS 227.7202-3(a) or as set forth in subparagraphs (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19, as applicable. Contractor/manufacturer is Mentor Graphics Corporation, 8005 Boeckman Road, Wilsonville, Oregon 97070-7777 USA.

12. **THIRD PARTY BENEFICIARY**. For any Software under this Agreement licensed by Mentor Graphics from Microsoft or other licensors, Microsoft or the applicable licensor is a third party beneficiary of this Agreement with the right to enforce the obligations set forth in this Agreement.

13. **CONTROLLING LAW**. This Agreement shall be governed by and construed under the laws of Ireland if the Software is licensed for use in Israel, Egypt, Switzerland, Norway, South Africa, or the European Union, the laws of Japan if the Software is licensed for use in Japan, the laws of Singapore if the Software is licensed for use in Singapore, People's Republic of China, Republic of China, India, or Korea, and the laws of the state of Oregon if the Software is licensed for use in the United States of America, Canada, Mexico, South America or anywhere else worldwide not provided for in this section

14. **SEVERABILITY**. If any provision of this Agreement is held by a court of competent jurisdiction to be void, invalid, unenforceable or illegal, such provision shall be severed from this Agreement and the remaining provisions will remain in full force and effect.

15. **MISCELLANEOUS**. This Agreement contains the entire understanding between the parties relating to its subject matter and supersedes all prior or contemporaneous agreements, including but not limited to any purchase order terms and conditions, except valid license agreements related to the subject matter of this Agreement which are physically signed by you and an authorized agent of Mentor Graphics. This Agreement may only be modified by a physically signed writing between you and an authorized agent of Mentor Graphics. Waiver of terms or excuse of breach must be in writing and shall not constitute subsequent consent, waiver or excuse. The prevailing party in any legal action regarding the subject matter of this Agreement shall be entitled to recover, in addition to other relief, reasonable attorneys' fees and expenses.

# Table of Contents

# Table of Contents [continued]

# List of Figures

# List of Figures [continued]

# About This Manual

The *Getting Started With Seamless Co-Verification Environment* manual contains short tutorials on how to use the Seamless Co-Verification Environment (Seamless CVE) in cosimulation with several supported simulators. To use this manual, you need to have a basic knowledge of electronic design and design-automation processes.

This manual contains hypertext links that are visible when the document is viewed online with compatible online document browsers. You can use hypertext links to access information at another location in this document.

The hypertext links appear in a contrasting color and may be underlined also. When you position the graphic pointer over the hypertext link, you can click the Select (left) mouse button on the hypertext link to go to its destination.

## Manual Organization

The first chapter, "Introduction," is a an overview of the design used in the tutorials in this manual. Subsequent chapters each contain a tutorial that guides you through a Seamless CVE session with a particular hardware simulator.

## Related Publications

Seamless CVE documentation is located in $CVE_HOME/doc. The following Mentor Graphics manuals contain important information on related topics:

- The *Seamless Co-Verification Environment Installation Instructions* explain how to install the Seamless CVE product.
- The *Seamless Co-Verification Environment User's and Reference Manual* contains an overview of the product, explains how to prepare designs for

cosimulation, and provides procedures for workstation and design configuration and cosimulation.

- The *Seamless CVE PSP Data Manuals* contain the relatively unchanging information on configuration, setup and usage of the Processor Support Packages.

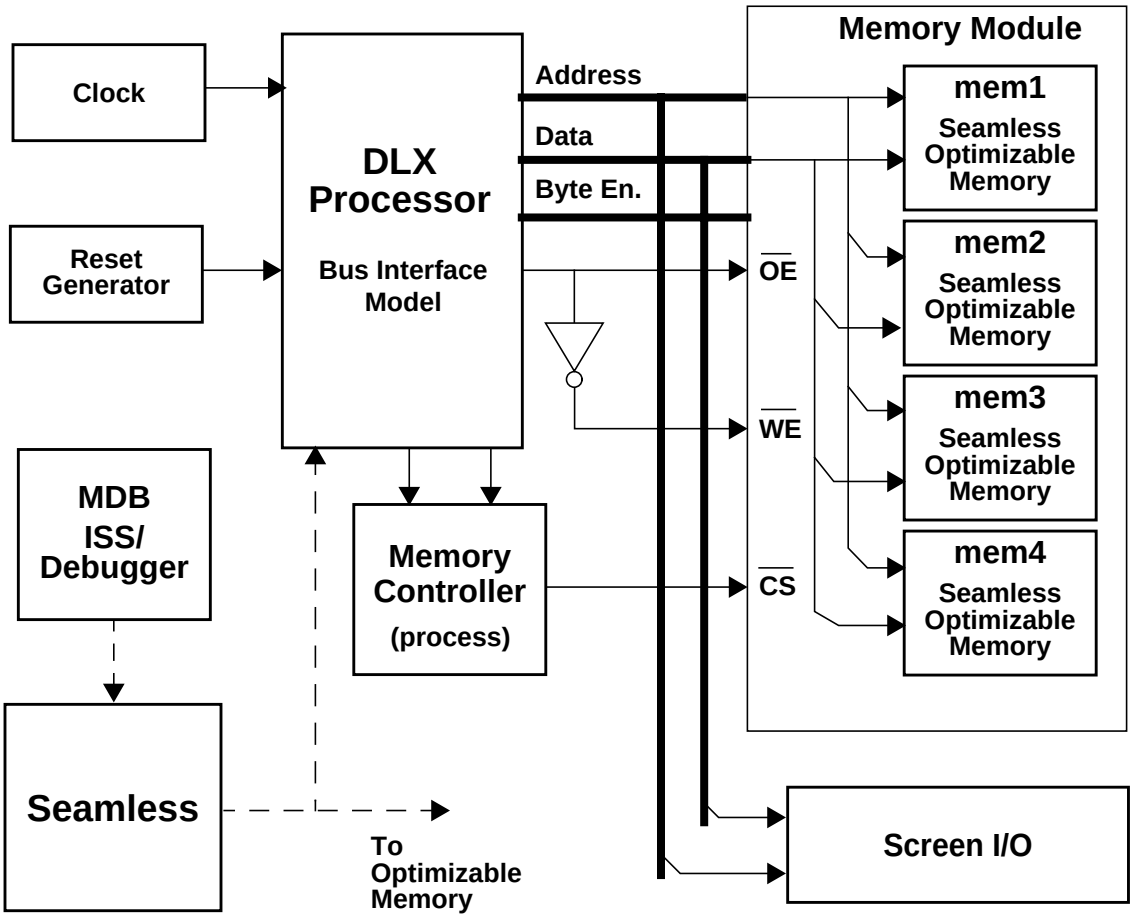
- The *Seamless CVE PSP Release Notes* contain changes and enhancements, known problems and workarounds, and corrected problems for Seamless CVE Processor Support Packages.

- The *Seamless CVE General Release Notes* contains changes and enhancements, known problems and workarounds, and corrected problems for the Seamless CVE product as a whole.

# Chapter 1
# Introduction

The tutorials in the following chapters use the demonstration design shown in Figure 1-1 to explain how to configure and cosimulate a design with Seamless CVE.

**Figure 1-1. Demonstration Design**



The demonstration design consists of the following major components:

- DLX Processor bus-interface model, through which the DLX target software running in the MDB instruction-set simulator (ISS) interacts with the hardware design.
- Memory Module, which consists of four Seamless optimizable memory instances.
- Clock Generator, which provides the system clock.
- Reset Generator, which initializes the processor.
- Screen I/O, which acts as a terminal, allowing direct user input and output through a display window.

Seamless CVE controls communications between the ISS, bus-interface model, and Seamless optimizable memory models. In Seamless optimizable memory models, memory access can take place either through hardware bus cycles, which are simulated in the hardware simulator, or through direct access from the ISS, without hardware bus cycles.

# Cosimulation with Hardware Simulators

The following chapters contain tutorials demonstrating cosimulation with these supported hardware simulators.

- Seamless CVE and ModelSim VHDL Tutorial
- Seamless CVE and NC-Verilog Tutorial

# Chapter 2
# Seamless CVE and ModelSim VHDL Tutorial

This tutorial consists of several sections designed to be performed in sequence, as follows:

- "Setting Up Your Environment" explains how to set up your workstation to run the tutorial.
- "Setting Up the Design Files" explains how to prepare the design components for cosimulation.
- "Invoking Seamless CVE" explains how to invoke Seamless CVE on the design.
- "Invoking the Hardware Simulator" explains how to set up the hardware simulator invocation. You invoke the simulator and allow the Seamless CVE models to register themselves with Seamless CVE.
- "Configuring the Processor" explains how to set up the software simulator invocation for the processor and how to configure the processor's address space and memory instances.
- "Saving the Configuration" explains how to save a Seamless CVE configuration in a file.
- "Starting the Cosimulation" explains how to start the cosimulation process.
- "Enabling Address-Range Optimization" explains how to perform Seamless CVE memory-access optimizations.

- "Exiting from the Cosimulation" explains how to exit from a cosimulation session.
- "Restarting Seamless CVE using the Configuration File" explains how to start up Seamless CVE using a previously saved configuration. It also shows how to use Seamless CVE time optimization to eliminate unnecessary hardware bus cycles.
- "Enabling Time Optimization" explains how to use Seamless CVE time optimization to eliminate unnecessary hardware bus cycles.
- "Enabling Instruction Fetch Optimization" describes a convenient way to enable data-access optimizations for all instruction fetches.

# Setting Up Your Environment

This tutorial assumes that Seamless CVE and ModelSim VHDL are installed and that proper license files and servers are set up as described in the product installation instructions. (See "Related Publications.")

Before starting the tutorial, set the following environment variables:

- CVE_HOME points to the top of the Seamless CVE installation tree for your platform (ss5 or hpu).
- MGLS_HOME points to *$CVE_HOME/mgls* (or to another appropriate location for the license server).
- MGLS_LICENSE_FILE points to the location of your license file.
- MODELTECH points to the installation directory for the ModelSim simulator.

# Setting Up the Design Files

Copy and set up the design files for the tutorial as follows:

1. Create a directory in which you can place the design files and run the tutorial.

2. Change directories to your tutorial directory.

3. Compile the VHDL design files into your tutorial directory by running the *build_design* script:

```
$CVE_HOME/example/tutorial/vsim_vhdl/build_design
```

   This compiles the DLX processor, SRAM, screen, reset generator, memory module, clock generator, and top-level tutorial design into a *work* directory.

4. Set the COMPILER_PATH environment variable:

```
setenv COMPILER_PATH $CVE_HOME/isms/dlxtools/bin
```

5. Compile and link the assembly-language program:

```
$CVE_HOME/isms/dlxtools/bin/dlx-gcc -g \
       $CVE_HOME/example/tutorial/common/sw/crt_tutorial.S \
          -c -I. -o crt_tutorial.o

$CVE_HOME/isms/dlxtools/bin/dlx-ld crt_tutorial.o -Ttext 0x0000 \
        -o crt_tutorial.elf
```

6. Copy the debugger “include” file to your tutorial directory:

```
cp $CVE_HOME/example/tutorial/common/sw/tutorial_int.inc .
```

# Invoking Seamless CVE

Invoke Seamless CVE using the following command:

```
$CVE_HOME/bin/cve
```

Figure 2-1 shows the Seamless CVE session window that appears after invocation. The main features of this window are as follows:

- File Menu—allows you to open configurations for editing, save configurations, and exit from Seamless CVE.

- Setup Menu—allows you to perform hardware-simulator setup as well as the CPU Setup operations described below.
- Optimize Menu—allows you to set up optimizations and optimization sets.
- View Menu—allows you to perform configuration and version checks and to view simulator output.
- Preference Menu—allows you to setup online help format, and enable help balloons or Auto Check as well as to save these preferences.
- Help Menu— allows you to gain access to online help.
- Tool bar—contains clickable icons that provide short cuts to all the major setup operations that you need when configuring a Seamless CVE cosimulation. The tool bar is divided into four major sections:

  **General Setup**
  contains icons for opening a configuration file, saving a configuration file, and starting the hardware simulator.

  **CPU Setup**
  contains icons for setting up software-simulator execution, defining memory access ranges, and mapping memory instances.

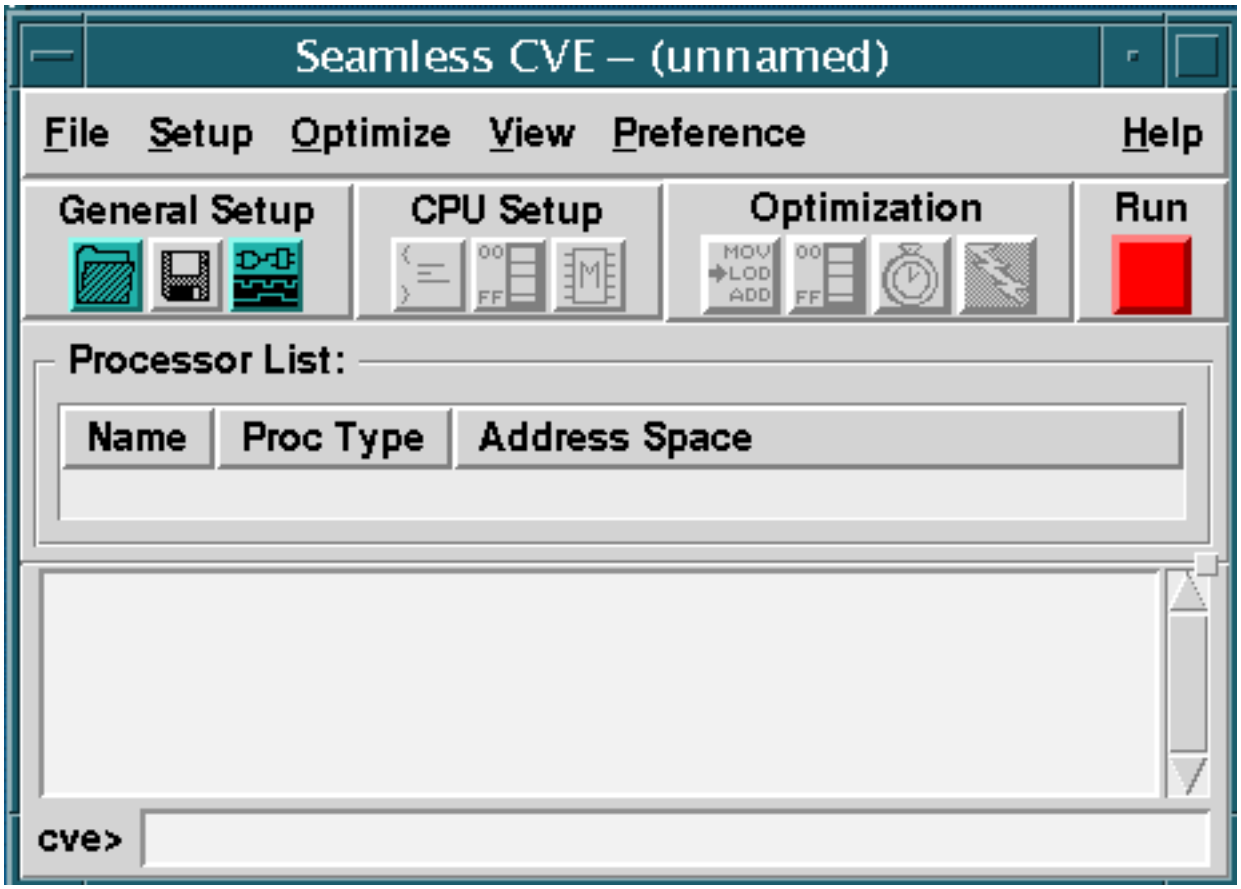
  **Optimization**
  contains icons for optimizing instruction fetches, optimizing accesses to address ranges, and optimizing time.

  **Run**
  contains the Run button, which starts a cosimulation session.

- Processor List window—lists all the registered processors in the design. (There are none registered now.)
- Transcript window— displays Seamless CVE command activity.

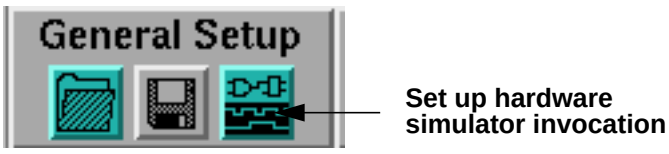
- Command line— allows you enter Seamless CVE commands.

**Figure 2-1. Seamless CVE Session Window**




# Invoking the Hardware Simulator

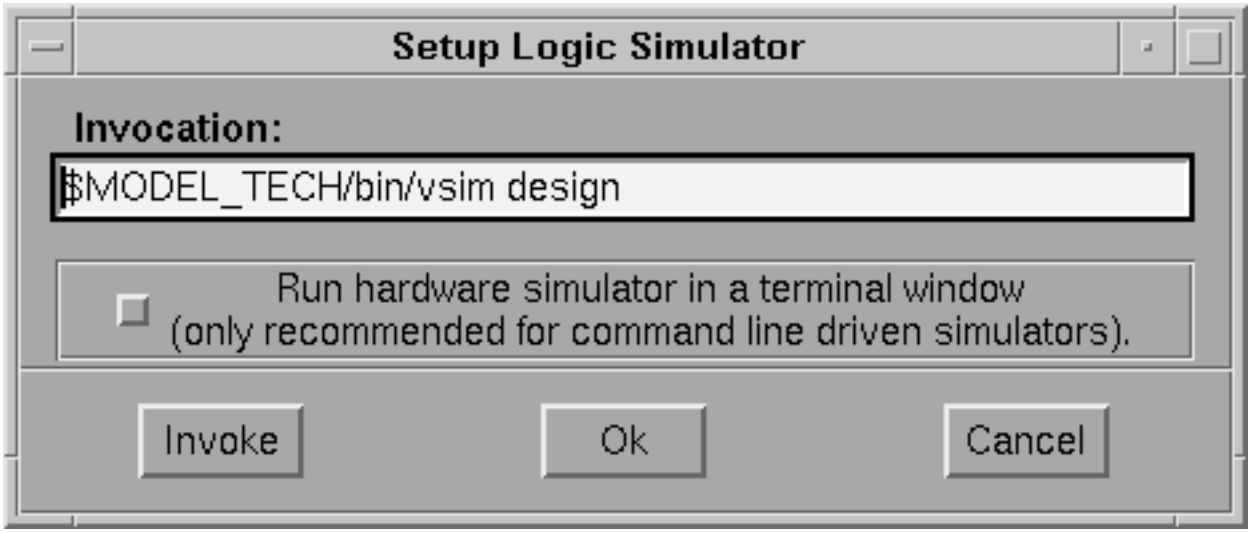
1. Click the Logic Simulation icon in the Session window (as shown below). This activates the Setup Logic Simulator dialog box.




2. Enter the following simulator invocation string in the text-entry box in the Setup Logic Simulator dialog box:

**$MODELTECH/bin/vsim design**

The variable MODELTECH points to the installation directory for the ModelSim VHDL simulator, as set up at the beginning of this tutorial. Do not enable the “Run hardware simulator in a terminal window” button because ModelSim VHDL runs from a GUI.




3. Click the Invoke button in the Setup Logic Simulator dialog box. This invokes the simulator and allows Seamless CVE memory models and processor models within the design to register themselves with the Seamless CVE. (The Ok button allows setup without invocation.)

   The ModelSim VHDL session window comes up. As the simulator loads the design, a listing for the processor model appears in the Processor List window, showing that the model has registered itself.

   Notice that once you set up the invocation, the Logic Simulation icon in the Session window is no longer highlighted, indicating that this step is done. If you wish to view the contents of the hardware simulator terminal window, select the **View > Hardware Output** menu item in the Seamless CVE window.

4. Select the **View > Run Status** menu item in the Seamless CVE window to display the Run Status message box. The indicator lists the hardware simulator as Vsim Uninitialized.

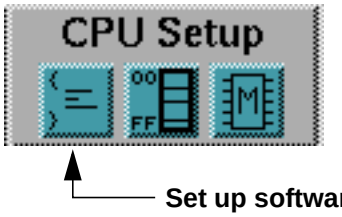
# Configuring the Processor

Setting up a Seamless CVE processor instance for cosimulation is a three-part process consisting of the following steps:

- Setting up the software simulator invocation.
- Mapping memory instances.
- Mapping memory-access ranges.

Set up the software simulator invocation as follows:

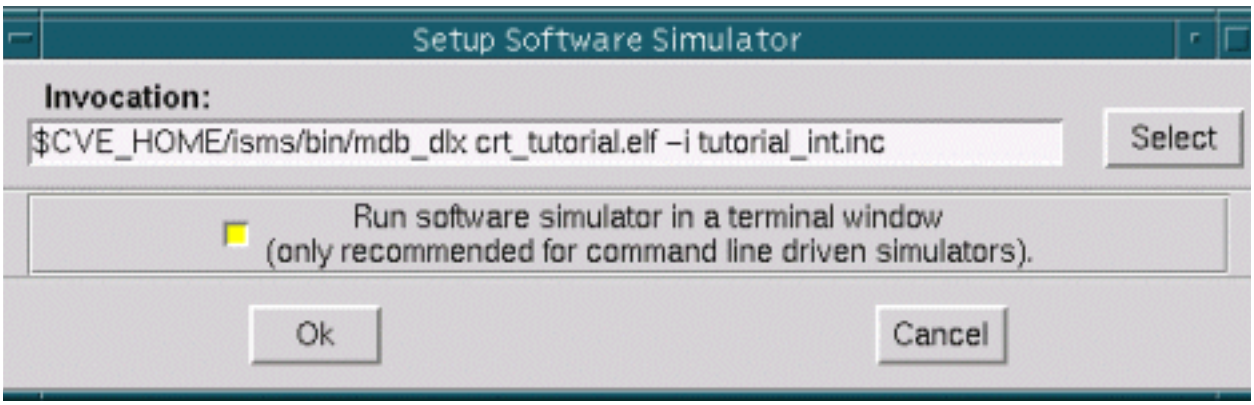
1. Click the Software Simulation icon in the Seamless CVE Session window (shown below). This activates the Setup Software Simulator dialog box.




2. Enter the following simulator invocation string under “Invocation:” in the Setup Software Simulator dialog box:

   **$CVE_HOME/isms/bin/mdb_dlx crt_tutorial.elf -i tutorial_int.inc**

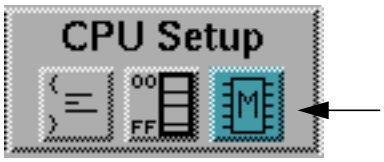
3. Click the button next to “Run software simulator in a terminal window” in the Setup Software Simulator dialog box, as shown below. You do this because the software simulator is not GUI driven.



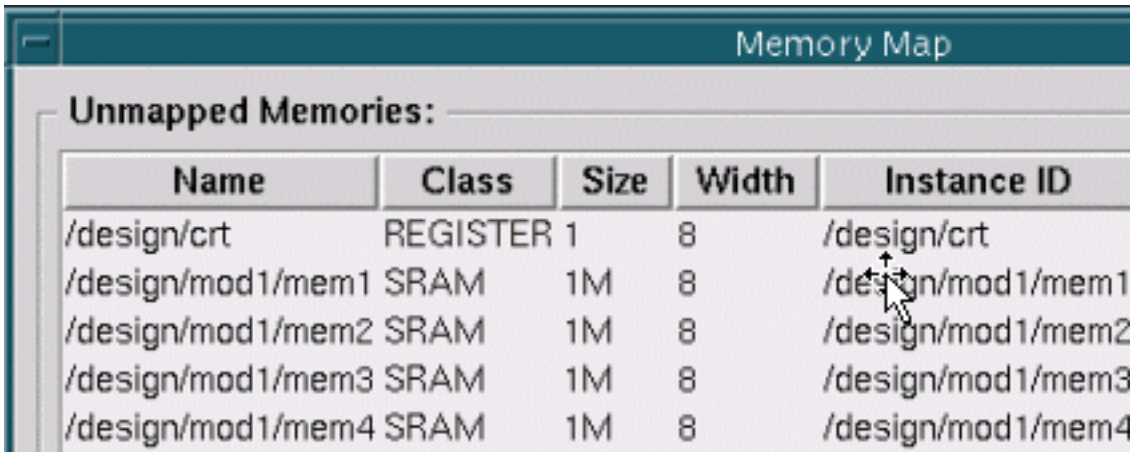
4. Click the Ok button. Notice that the Software Simulation icon in the Session window is no longer highlighted, indicating that setup is done.

You need to map the Seamless optimizable memory instances into the processor's address space. Do this by performing the following steps:

1. Click the Map Memory Instances icon in the CVE Session window (shown below). This activates the Memory Map dialog box.

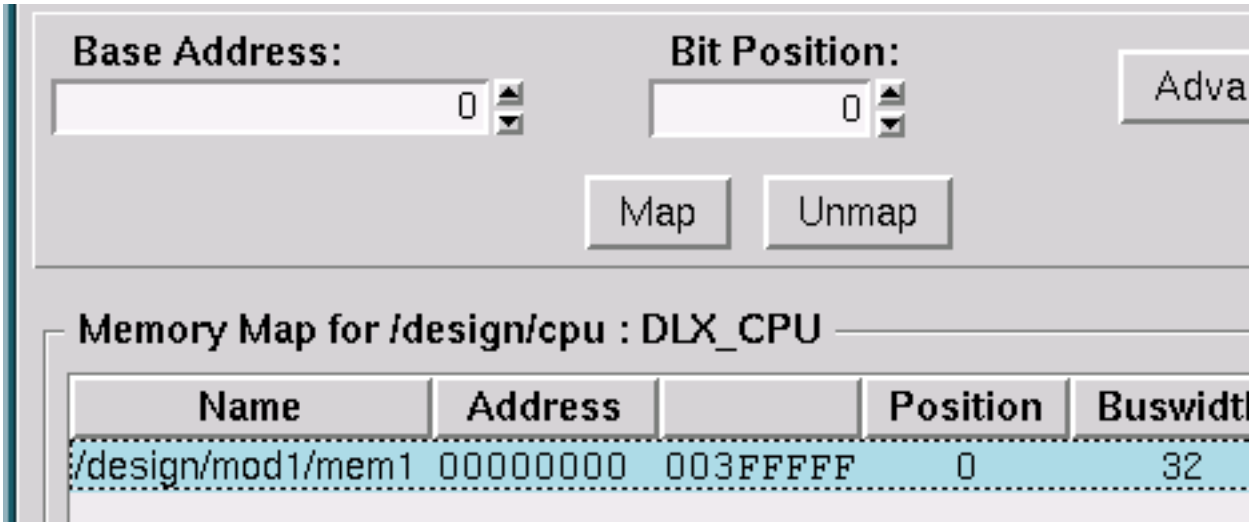



Each of the four memory instances in the memory module is 1Mbyte long and 8 bits wide. These instances will be mapped across the processor's 32-bit data bus such that each instance occupies 8 bits of the bus. As shown below, the logical names of the instances are listed on the far left side and the design instance identifiers are on the right side (and they are the same).




2. Select the *mem1* memory instance by clicking on that instance's entry in the Unmapped Memories list. The instance entry is highlighted when selected. Notice that the Base Address and Bit Position entries are both 0, as shown below.

3. Click the Map button to map *mem1* into the processor address space. The entry for *mem1* appears in the Memory Map list, as shown below.




4. Select the *mem2* memory instance by clicking on that instance's entry in the Unmapped Memories list.

5. Specify the bit position for the *mem2* instance by clicking on the up arrow button to increment the value in the box to 8. This maps the *mem2* instance into bits 8 through 15 of the processor's 32-bit word.

6. Click the Map button to map *mem2* into the processor address space. The entry for *mem2* appears in the Memory Map list with a bit position of 8.

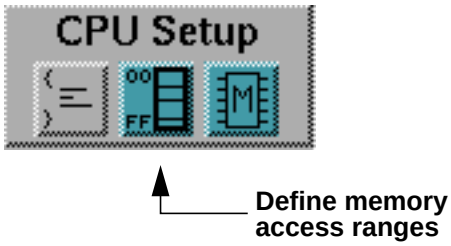
7. Map the *mem3* and *mem4* instances into bit positions 16 and 24, respectively.

8. Select the *crt* entry in the Unmapped Memories list. Enter an address of 3FFFFFF0 in the Base Address box. Set the bit position to 0 and click the Map button to map the instance into the processor address space.

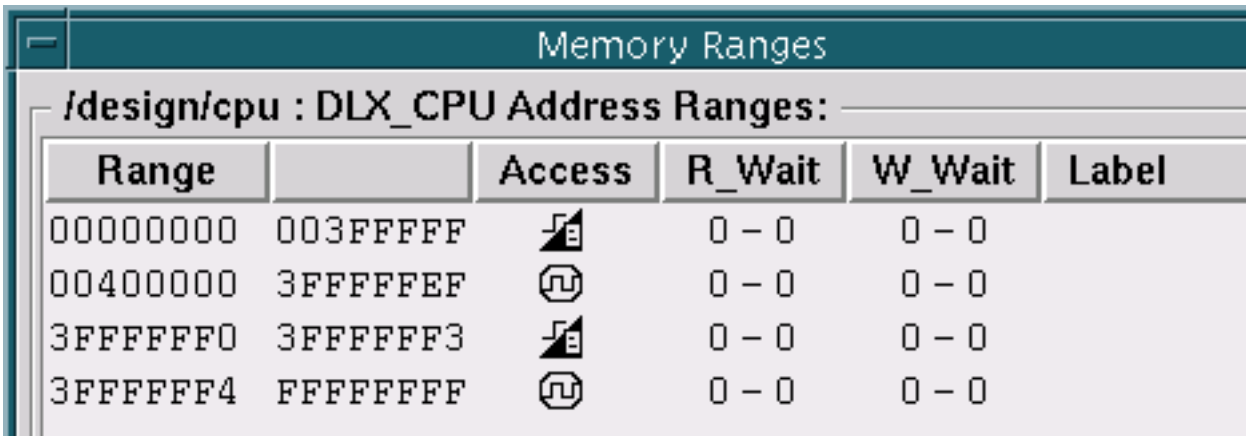
9. Click the Close button to dismiss the Memory Map dialog box.

Set up the memory-access ranges for the processor as follows:

1. Click the Memory Access Ranges icon in the Seamless CVE Session window (shown below).




   The Memory Ranges dialog box appears. The Address Ranges list box contains four entries, as shown below.



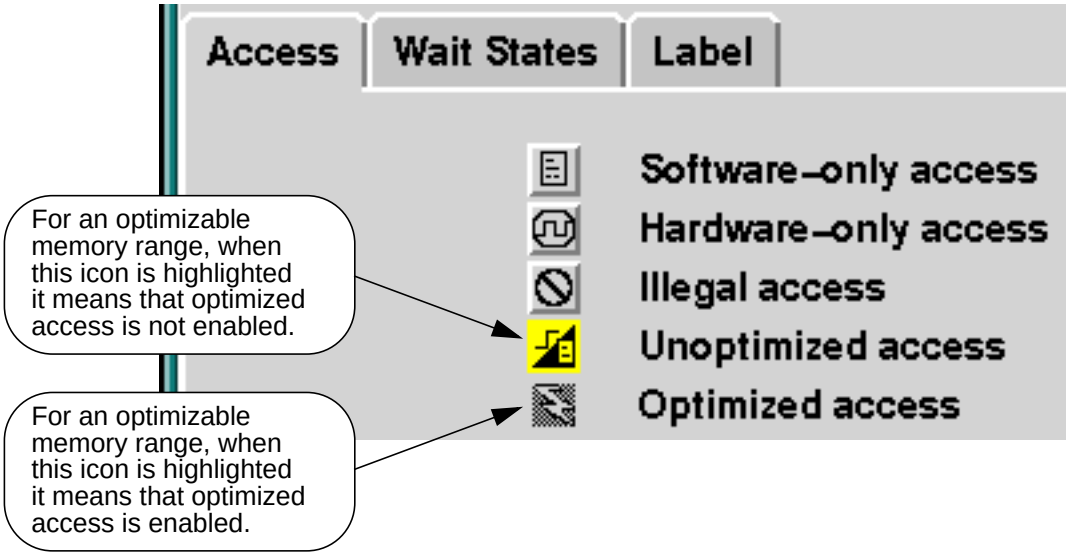

| Range | | Access | R_Wait | W_Wait | Label |
|---|---|---|---|---|---|
| 00000000 | 003FFFFF | | 0 – 0 | 0 – 0 | |
| 00400000 | 3FFFFFEF | | 0 – 0 | 0 – 0 | |
| 3FFFFFF0 | 3FFFFFF3 | | 0 – 0 | 0 – 0 | |
| 3FFFFFF4 | FFFFFFFF | | 0 – 0 | 0 – 0 | |

   Notice that the first entry, for addresses 0 through 3FFFFF, is the address range of the memory module in the hardware design. Since these instances are Seamless memory models, this address range is optimizable.

2. Select the first entry (0 to 3FFFFF) in the Address Ranges list box. The start and end addresses appear in the Start Address and End Address entry boxes.

   Notice the icon under the Access heading in the list box and the corresponding icon on the Access tab in the lower portion of the Memory Ranges dialog box (shown below). Optimized access to this range has not

been enabled yet, meaning that all accesses to this memory range generate hardware bus cycles.




3. Create a label for the optimizable address range as follows:

   a. Click the Label tab.

   b. Enter a label of “optimizable” in the Range Label box, then press Return or click the Apply button. The label appears in the optimizable entry in the Address Ranges list box.

4. Set up wait-state values for the optimizable memory:

   a. Click the Wait States tab.

   b. Enter values of 2 in the Read and Write boxes under Initial, then press Return or click the Apply button. (Leave the Burst boxes blank.) The wait-state values appear in the Address Ranges list box.

5. Select the next unlabeled range (400000 - 3FFFFFFF) listed in the Address Ranges box. Notice that the listed range values appear in the Start and End address boxes.

6. Replace the End Address value by double-clicking on the entry box and entering 4000FF.


**Note**

There are no memory instances, either Seamless CVE or otherwise, in this address range. This and the remaining steps in this section simply demonstrate how to set up memory-access ranges.

7. Make sure the Label tab is on top and enter “hardware” in the Range Label entry box.

8. Click the Apply button. A new entry, at addresses 400000 - 40000FF, with the label “hardware” appears in in the Address Ranges list box. The former hardware-only range is now split into two ranges.

9. Select the next range on the list and label it “hardware2.”

10. Select the *screen* entry, which is next in the list. Although the *screen* is actually based on an optimizable register memory (hence the “optimizable” icon under the Access column), we will pretend it is hardware-only for the present. Enter a label of “screen” for this range then click Apply.

11. Select the unlabeled range at the bottom of the list in the Address Ranges list box (3FFFFFF4 to FFFFFFFF).

12. Enter “illegal” in the Range Label text-entry box, then click the Apply button.

13. Select the Access tab and then click the Illegal access button. This makes addresses 400100 to FFFFFFFF an “illegal” range, as indicated by the icon in the Address Ranges list.

14. Click the Close button to close the window. Notice that the Memory Access Ranges icon in the Session window is no longer highlighted, indicating that this step has been done.

# Saving the Configuration

Save the current configuration by performing the following steps:

1. Click the Save Configuration icon in the Seamless CVE Session window (shown below) or select the **File > Save As** menu item.


**Save Configuration**

2. In the Save As dialog box, enter a configuration filename (and path, if necessary).

3. Click Ok to save the configuration.

# Starting the Cosimulation

1. Click the **Run** button in the Seamless CVE main window. The button changes to yellow and remains pressed, indicating that cosimulation can start. In addition, a terminal window appears showing a software simulator startup message:

   **CVE NOTE: Waiting to connect to hardware process ...**

2. Move the cursor to the ModelSim VHDL simulator's main window and enter the following command at the VSIM prompt:

   **view signals**

3. In the Signals window select **View > Wave > Signals in Region** to bring up the ModelSim VHDL waveform window. If necessary, minimize the Waveform and Signals windows to free some display space.

4. Enter the following at the ModelSim VHDL session window prompt:

   **run -all**

The **run** command starts the hardware simulation (but the simulation cannot run until its software begins to execute). Notice that the Run button turns green, indicating that cosimulation has begun. The Seamless CVE models in the design cause Seamless CVE to start up the software simulator. The software simulator terminal window displays a series of messages followed by this prompt:

**(mdb)**

The Run Status window indicates that the hardware simulator is waiting (for the software simulation to start) and the software simulator is interactive (that is, waiting for you to enter a command to start the software simulation). Neither hardware nor software simulation can proceed until you respond to the software simulator prompt.


5. Enter ***run*** at the **(mdb)** prompt. The Screen window appears and displays the following message.

```
Welcome to the Seamless Tutorial!!
Using the DLX Processor and the MDB Debugger.
```

Notice that the Run Status for both of the simulators changes to “Running” then back to “Interactive” for both simulators. The software simulator runs to a breakpoint and displays a message similar (not necessarily identical) to the following one:

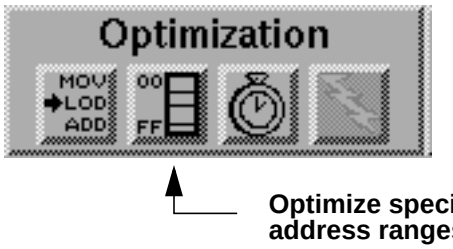
```
25 secs; 656 instr; 26 instr/sec
Hit Breakpoint 1 at 0x00000000 (begin+0x00000000)
(0x00000000) begin              nop
(mdb)
```

In addition, waveforms appear in the hardware simulator waveform window, showing the bus activity that takes place during the simulation run. At this point, the run status of both simulators is interactive.

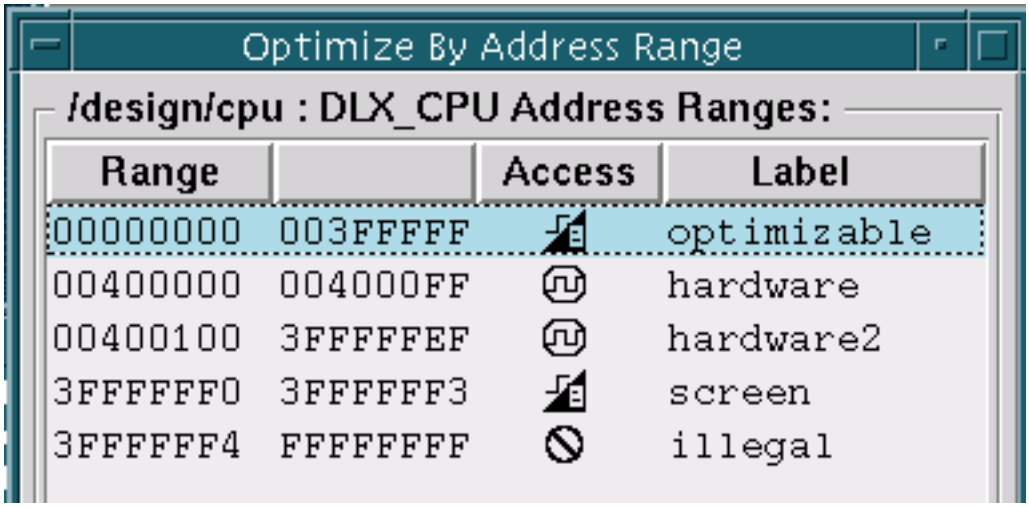
# Enabling Address-Range Optimization

Enable address-range optimization for the optimizable addresses in the processor address space by performing the following steps:

1. Click the Optimize Address Ranges icon in the Seamless CVE Session window (shown below). This activates the Optimize by Address Range dialog box.

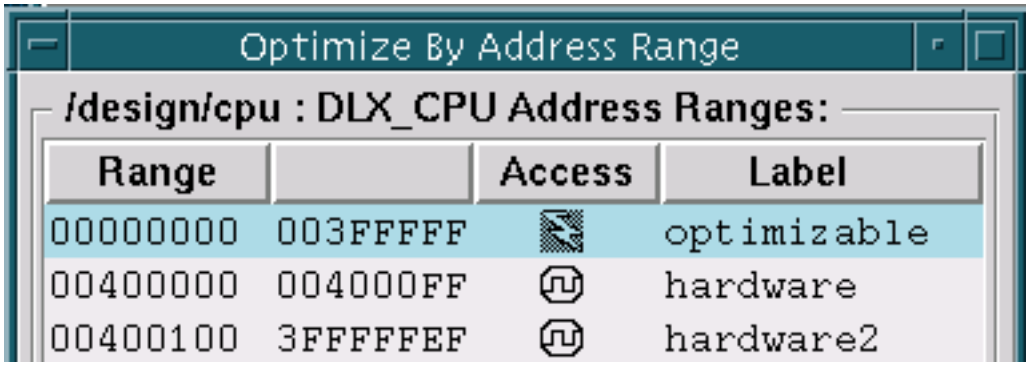



The Address Ranges list contains the ranges you set up in the previous section, as shown below.




2. Click the range labeled “optimizable.” The start and end addresses of the range appear in the entry boxes under Optimization Entry.

3. Click the On button. This modifies the optimizable range in the Address Ranges list as shown below. The “lightning bolt” icon indicates that processor accesses to this range can take place directly through software rather than generating hardware bus cycles.




   You can enable optimized access to any portion of an optimizable address range.

4. Click the Close button to dismiss the Optimize By Address Range dialog box. The Optimize Address Ranges icon in the Seamless CVE session window is now highlighted.

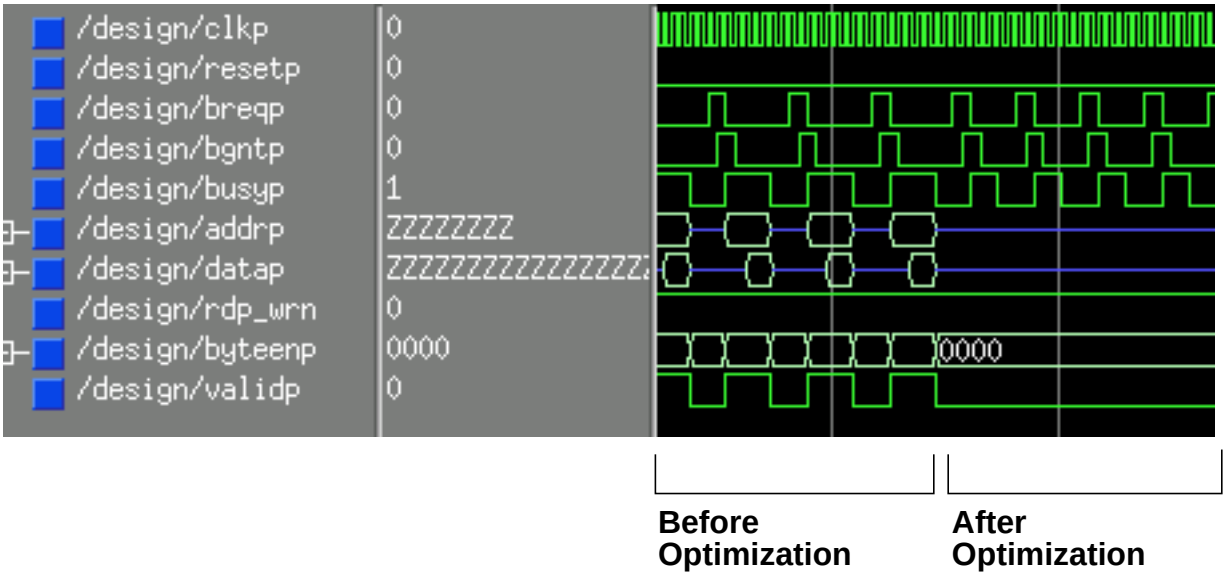
5. Move the cursor to the transcript window in the ModelSim simulator’s session window and enter the following command:

   **run -all**

   This time, the simulation runs faster. A portion of the waveform display that overlaps the first and second runs is shown below. Notice the relative lack of address- and data-bus activity during the second run. The clock runs in the hardware simulation as before. However, the software simulator now accesses memory directly except when writes to the screen, in which case it needs to generate bus activity in the hardware simulation. You can confirm

this by noting that a value of 0x3FFFFFF0 is present on the address bus during these write cycles. (Use the cursor.)

**Figure 2-2. ModelSim Waveform Window**




6. Select the **Optimize > Name Optimization** menu item. This activates the Optimization Set dialog box. This dialog box allows you to save the current optimization state for later recall.

7. Type in a name such as *opt1* in the text-entry box in the Optimization Set dialog box then click Ok to save the optimization set. Notice that the Optimization Set Enable ("lightning bolt") icon under Optimization group in the main Seamless CVE window becomes pressed and is highlighted yellow, meaning that a named optimization set is active.


8. Click the Optimization Set Enable icon to disable the optimization set. The icon becomes "unpressed" and is no longer highlighted. (You can also use the **Optimize > Unoptimize** menu item.) Click the icon again to re-enable the optimization set.

9. Save the current configuration by selecting the **File > Save** menu item.

# Exiting from the Cosimulation

1. Select the **File > Exit** menu item in the Seamless CVE session window or enter “exit” at the command-line. A message box appears saying “Simulator is running. Exit anyway?”

2. Click Yes to exit from the ModelSim VHDL simulator. The Seamless CVE Session window and simulator windows disappear.

# Restarting Seamless CVE using the Configuration File

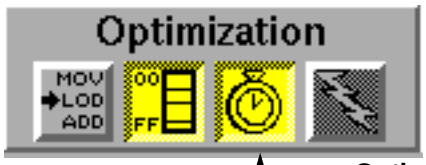
Invoke Seamless CVE using the following command:

```
$CVE_HOME/bin/cve   my_config
```

where *my_config* is the name of the configuration file you saved previously. Seamless CVE, ModelSim VHDL simulator, and software simulator all start up. Notice that the configuration in *my_config* is restored and the cosimulation is ready to start. You can confirm this by clicking on the appropriate CPU Setup and Optimization icons and checking the values that appear in the corresponding dialog boxes.

# Enabling Time Optimization

1. Click the Optimize time icon to activate Seamless time optimizations. The Time Optimizations icon button (shown below) is now “pressed” and highlighted yellow.




**Optimize time by allowing software to run ahead of hardware (using Time Optimization dialog box settings)**

2. Move the cursor to the transcript window in the ModelSim VHDL simulator’s session window and enter the following commands:

   **add wave ***
   **run -all**

   The Run button on the Seamless CVE session window turns green, indicating that cosimulation has begun.

3. Enter ***run*** at the (mdb) prompt in the software simulator’s terminal, as you did earlier in “Starting the Cosimulation.”

   Notice that the same number of instructions are executed as before, but in much less time. The software has run at full speed, decoupled from the hardware simulation, and far fewer clock cycles have occurred in the hardware simulation. The only time hardware clock cycles are required is when the program accesses the screen to print out the message. Although the tutorial program is too simple to show the full effect, usually over 99 percent of hardware clock cycles can be eliminated without affecting simulation results.

4. Select the **Optimize > Name Optimization** menu item. The Optimization Set dialog box appears.

5. Enter a name (different than the one you used previously) for the current optimization set, then click Ok.

6. Select the **Optimize > Recall Optimization** menu item. The Optimization Sets dialog box appears.

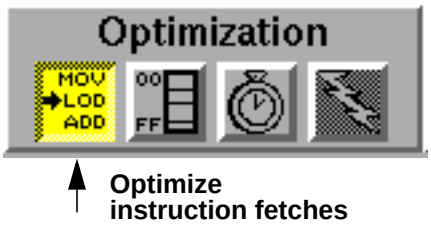
7. Select the first optimization set you saved by clicking on its name, then click the Apply button. Now select the optimization set you saved in this section and click Apply. Notice the effect on the Optimizaton icons.

8. Click the Close button to dismiss the Optimization Set dialog box.

9. Select the **Optimize > Unoptimize** menu item to clear the current optimization set.

# Enabling Instruction Fetch Optimization

1. Click in on the Instruction Fetch Optimizations icon. The icon becomes highlighted as shown below.




   Instruction fetch optimization is a convenient way to automatically enable optimizations for all instruction-fetch bus cycles. This eliminates hardware bus-cycle activity for instruction fetches, allowing the ISS to fetch the instructions directly from memory.

2. Run the hardware and software simulators as before.

3. Select the **Optimize > Unoptimize** menu item.

4. Run the hardware and software simulators again. Notice the difference in run time and bus activity with and without instruction fetch optimizations enabled.

This concludes the *Seamless CVE and ModelSim VHDL Tutorial.* You can experiment with different combinations of optimization settings to see how they

affect cosimulation. For example, try turning on optimizations for single addresses or ranges of addresses to see how they affect simulation speed and bus cycle activity.

# Chapter 3
# Seamless CVE and NC-Verilog Tutorial

This tutorial consists of several sections designed to be performed in sequence, as follows:

- "Setting Up Your Environment" explains how to set up your workstation to run the tutorial.
- "Setting Up the Design Files" explains how to prepare the design components for cosimulation.
- "Invoking Seamless CVE" explains how to invoke Seamless CVE on the design.
- "Invoking the Hardware Simulator" explains how to set up the hardware simulator invocation. You invoke the simulator and allow the Seamless CVE models to register themselves with Seamless CVE.
- "Configuring the Processor" explains how to set up the software simulator invocation for the processor and how to configure the processor's address space and memory instances.
- "Saving the Configuration" explains how to save a Seamless CVE configuration in a file.
- "Starting the Cosimulation" explains how to start the cosimulation process.
- "Enabling Address-Range Optimization" explains how to perform Seamless CVE memory-access optimizations.

- "Exiting from the Cosimulation" explains how to exit from a cosimulation session.
- "Restarting Seamless CVE using the Configuration File" explains how to start up Seamless CVE using a previously saved configuration. It also shows how to use Seamless CVE time optimization to eliminate unnecessary hardware bus cycles.
- "Enabling Time Optimization" explains how to use Seamless CVE time optimization to eliminate unnecessary hardware bus cycles.
- "Enabling Instruction Fetch Optimization" describes a convenient way to enable data-access optimizations for all instruction fetches.

# Setting Up Your Environment

This tutorial assumes that Seamless CVE and NC Verilog are installed and that proper license files and servers are set up as described in the product installation instructions. (See "Related Publications.")

Before starting the tutorial, set the following environment variables:

- CVE_HOME points to the top of the Seamless CVE installation tree for your platform (ss5 or hpu).
- MGLS_HOME points to *$CVE_HOME/mgls* (or to another appropriate location for the license server).
- MGLS_LICENSE_FILE points to the location of your license file.
- CDS_INST_DIR points to the top of the installation directory for the NC Verilog application. (Refer to the product documentation for detailed information.)
- CDS_ROOT points to the top of the installation directory for the NC Verilog application. (Refer to the product documentation for detailed information.)

- SHLIB_PATH (for HPUX): add the current directory as well as $CVE_HOME/lib and $CDS_INST_DIR/tools/lib to the SHLIB_PATH setting.

- LD_LIBRARY_PATH (for Solaris): add the current directory as well as $CVE_HOME/lib and $CDS_INST_DIR/tools/lib to the LD_LIBRARY_PATH setting.

# Setting Up the Design Files

Copy and set up the design files for the tutorial as follows:

1. Create a directory in which you can place the design files and run the tutorial.

2. Change directories to your tutorial directory.

3. Compile the Verilog design files into your tutorial directory by running the *build_design* script:

   ```
   $CVE_HOME/example/tutorial/ncvlog/build_design
   ```

   This compiles the DLX processor, SRAM, screen, reset generator, memory module, clock generator, and top-level tutorial design into your local directory.

4. Set the COMPILER_PATH environment variable:

   ```
   setenv COMPILER_PATH $CVE_HOME/isms/dlxtools/bin
   ```

5. Compile and link the assembly-language program:

   ```
   $CVE_HOME/isms/dlxtools/bin/dlx-gcc -g \
          $CVE_HOME/example/tutorial/common/sw/crt_tutorial.S \
      -c -I. -o crt_tutorial.o

   $CVE_HOME/isms/dlxtools/bin/dlx-ld crt_tutorial.o -Ttext 0x0000 \
           -o crt_tutorial.elf
   ```

6. Copy the debugger “include” file to your tutorial directory:

```
cp $CVE_HOME/example/tutorial/common/sw/tutorial_int.inc .
```

# Invoking Seamless CVE

Invoke Seamless CVE using the following command:

```
$CVE_HOME/bin/cve
```

Figure 3-1 shows the Seamless CVE session window that appears after invocation. The main features of this window are as follows:

- File Menu—allows you to open configurations for editing, save configurations, and exit from Seamless CVE.
- Setup Menu—allows you to perform hardware-simulator setup as well as the CPU Setup operations described below.
- Optimize Menu—allows you to set up optimizations and optimization sets.
- View Menu—allows you to perform configuration and version checks and to view simulator output.
- Preference Menu—allows you to setup online help format, and enable help balloons or Auto Check as well as to save these preferences.
- Help Menu— allows you to gain access to online help.
- Tool bar—contains clickable icons that provide short cuts to all the major setup operations that you need when configuring a Seamless CVE cosimulation. The tool bar is divided into four major sections:

  **General Setup**
  contains icons for opening a configuration file, saving a configuration file, and starting the hardware simulator.

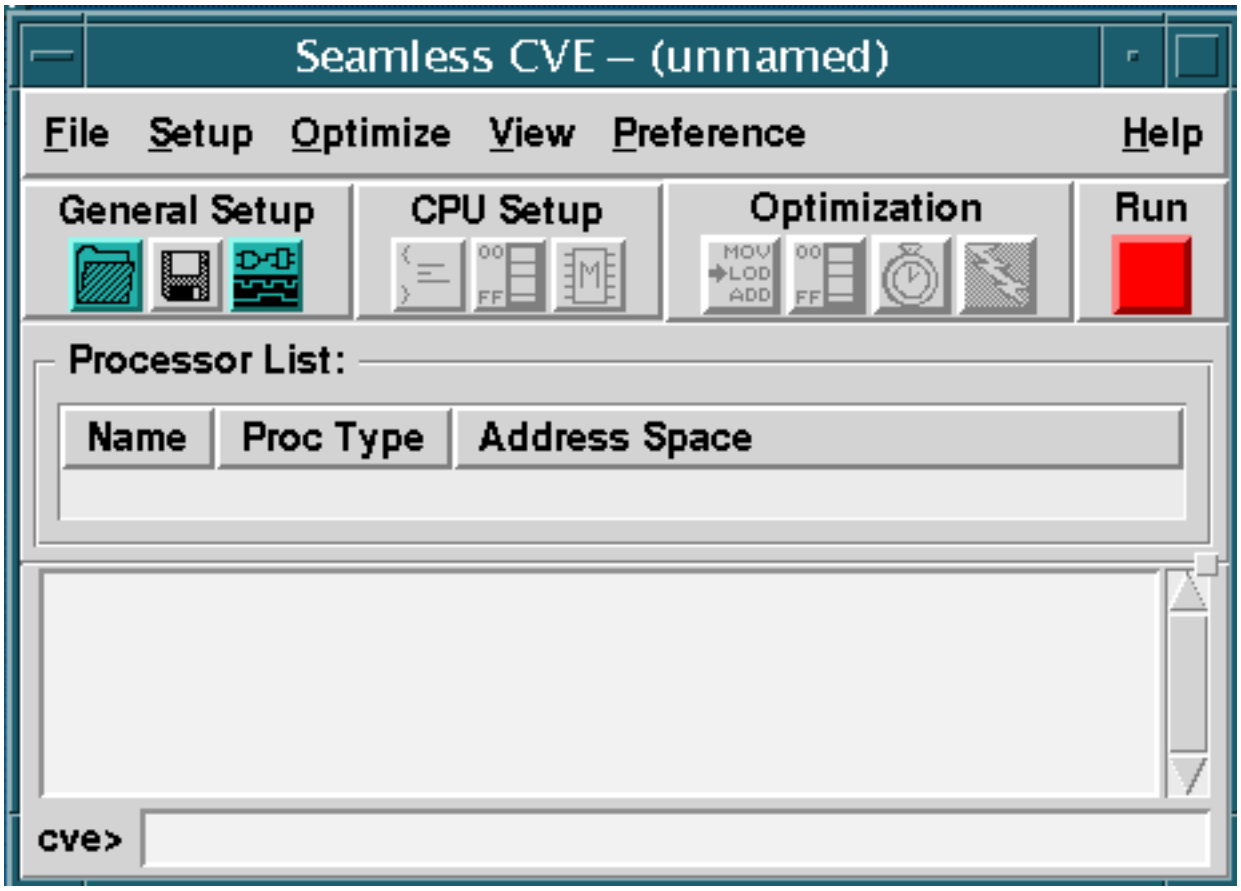
  **CPU Setup**
  contains icons for setting up software-simulator execution, defining memory access ranges, and mapping memory instances.

**Optimization**
contains icons for optimizing instruction fetches, optimizing accesses to address ranges, and optimizing time.

**Run**
contains the Run button, which starts a cosimulation session.

- Processor List window—lists all the registered processors in the design. (There are none registered now.)
- Transcript window— displays Seamless CVE command activity.
- Command line— allows you enter Seamless CVE commands.

**Figure 3-1. Seamless CVE Session Window**



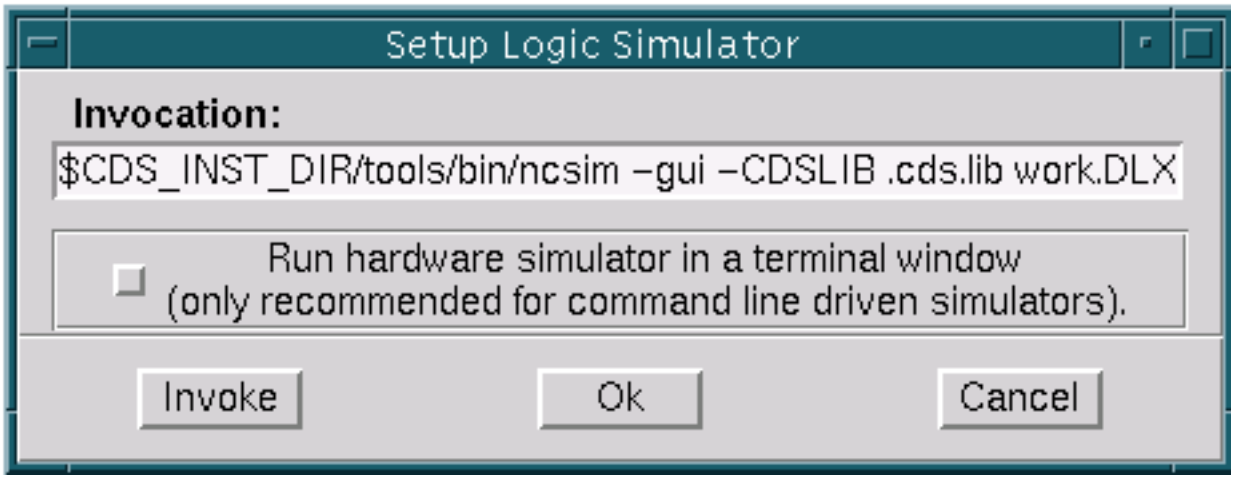
# Invoking the Hardware Simulator

1. Click the Logic Simulation icon in the Session window (as shown below). This activates the Setup Logic Simulator dialog box.


2. Enter the following simulator invocation string in the text-entry box in the Setup Logic Simulator dialog box:

   **$CDS_INST_DIR/tools/bin/ncsim -gui -CDSLIB cds.lib work.DLX_design:snap**

   The variable MODELTECH points to the installation directory for the ModelSim VHDL simulator, as set up at the beginning of this tutorial. Do not enable the “Run hardware simulator in a terminal window” button because NC Verilog runs from a GUI.




3. Click the Invoke button in the Setup Logic Simulator dialog box. This invokes the simulator and allows Seamless CVE memory models and processor models within the design to register themselves with the Seamless CVE. (The Ok button allows setup without invocation.)

The ModelSim VHDL session window comes up. As the simulator loads the design, a listing for the processor model appears in the Processor List window, showing that the model has registered itself.

Notice that once you set up the invocation, the Logic Simulation icon in the Session window is no longer highlighted, indicating that this step is done. If you wish to view the contents of the hardware simulator terminal window, select the **View > Hardware Output** menu item in the Seamless CVE window.

4. Select the **View > Run Status** menu item in the Seamless CVE window to display the Run Status message box. The indicator lists the hardware simulator as Vsim Uninitialized.

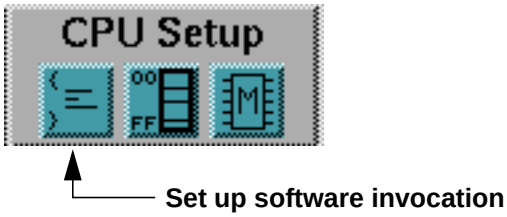
# Configuring the Processor

Setting up a Seamless CVE processor instance for cosimulation is a three-part process consisting of the following steps:

- Setting up the software simulator invocation.
- Mapping memory instances.
- Mapping memory-access ranges.

Set up the software simulator invocation as follows:

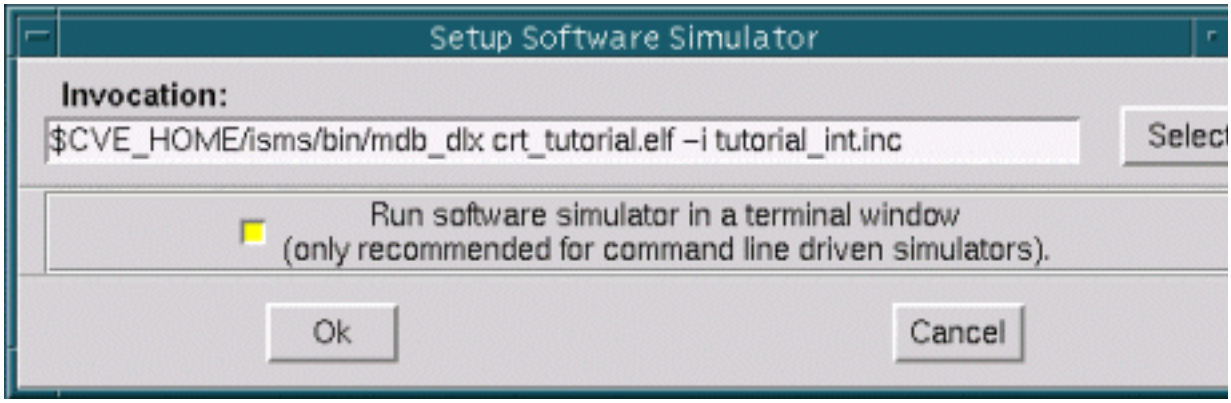
1. Click the Software Simulation icon in the Seamless CVE Session window (shown below). This activates the Setup Software Simulator dialog box.




2. Enter the following simulator invocation string under “Invocation:” in the Setup Software Simulator dialog box:

**$CVE_HOME/isms/bin/mdb_dlx crt_tutorial.elf -i tutorial_int.inc**

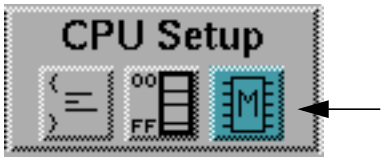
3. Click the button next to “Run software simulator in a terminal window” in the Setup Software Simulator dialog box, as shown below. You do this because the software simulator is not GUI driven.




4. Click the Ok button. Notice that the Software Simulation icon in the Session window is no longer highlighted, indicating that setup is done.

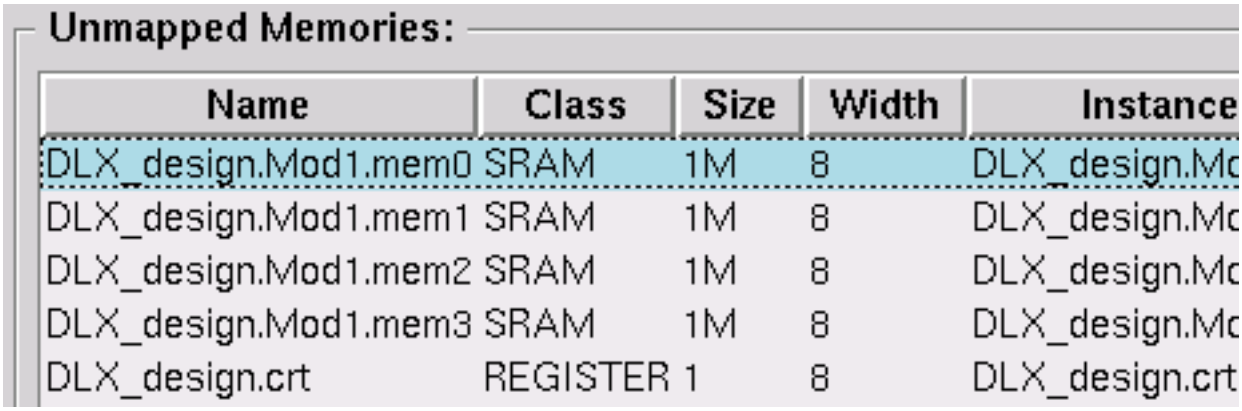
You need to map the Seamless optimizable memory instances into the processor’s address space. Do this by performing the following steps:

1. Click the Map Memory Instances icon in the CVE Session window (shown below). This activates the Memory Map dialog box.



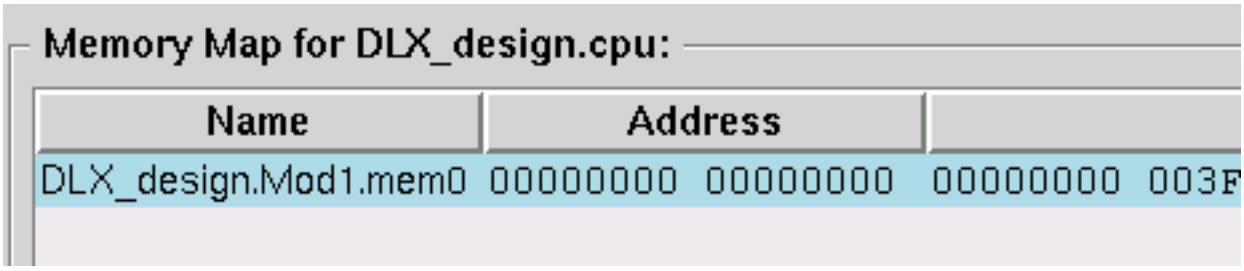

   Each of the four memory instances in the memory module is 1Mbyte long and 8 bits wide. These instances will be mapped across the processor’s 32-bit data bus such that each instance occupies 8 bits of the bus. As shown

below, the logical names of the instances are listed on the far left side and the design instance identifiers are on the right side (and they are the same).




2. Select the *mem0* memory instance by clicking on that instance's entry in the Unmapped Memories list. The instance entry is highlighted when selected. Notice that the Base Address and Bit Position entries are both 0, as shown below.

3. Click the Map button to map *mem0* into the processor address space. The entry for *mem0* appears in the Memory Map list, as shown below.




4. Select the *mem1* memory instance by clicking on that instance's entry in the Unmapped Memories list.

5. Specify the bit position for the *mem1* instance by clicking on the up arrow button to increment the value in the box to 8. This maps the *mem1* instance into bits 8 through 15 of the processor's 32-bit word.

6. Click the Map button to map *mem1* into the processor address space. The entry for *mem1* appears in the Memory Map list with a bit position of 8.

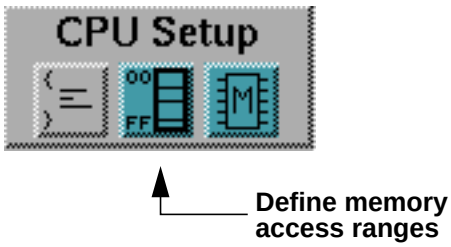
7. Map the *mem2* and *mem3* instances into bit positions 16 and 24, respectively.

8. Select the *crt* entry in the Unmapped Memories list. Enter a base address of 3FFFFFF0 and bit position of 0 then click the Map button to map the instance the processor address space.

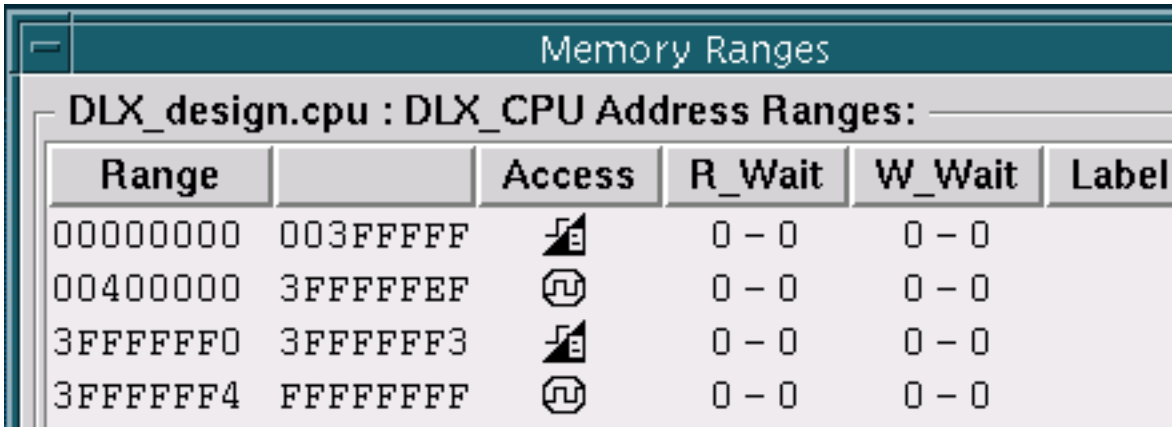
9. Click the Close button to dismiss the Memory Map dialog box.

Set up the memory-access ranges for the processor as follows:

1. Click the Memory Access Ranges icon in the Seamless CVE Session window (shown below).



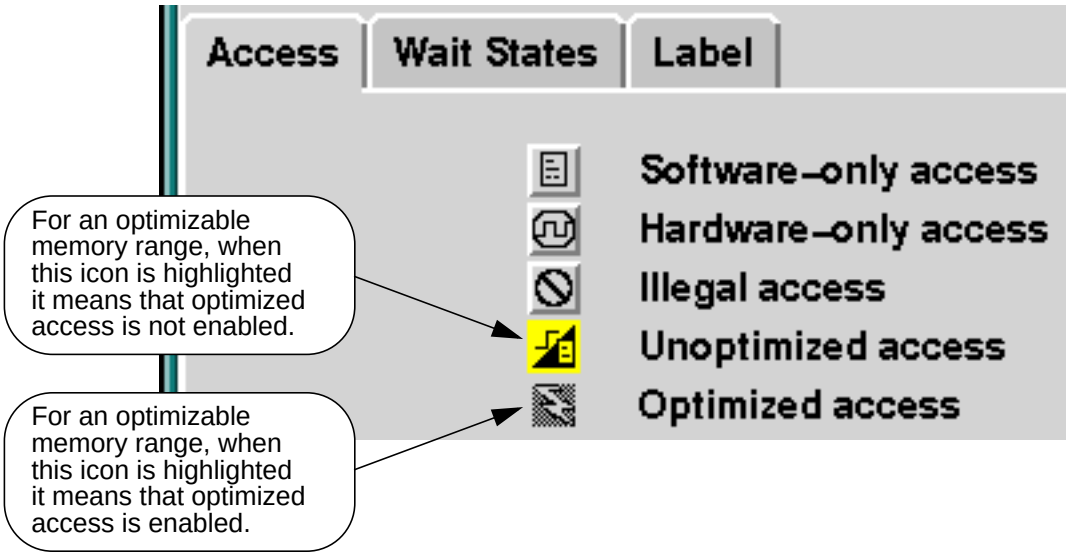

The Memory Ranges dialog box appears. The Address Ranges list box contains four entries, as shown below.




Notice that the first entry, for addresses 0 through 3FFFFF, is the address range of the memory module in the hardware design. Since these instances are Seamless memory models, this address range is optimizable. The icon under “Access” indicates this.

2. Select the first entry (0 to 3FFFFF) in the Address Ranges list box. The start and end addresses appear in the Start Address and End Address entry boxes.

   Notice the icon under the Access heading in the list box and the corresponding icon on the Access tab in the lower portion of the Memory Ranges dialog box (shown below). Optimized access to this range has not been enabled yet, meaning that all accesses to this memory range generate hardware bus cycles.




3. Create a label for the optimizable address range as follows:

   a. Click the Label tab.

   b. Enter a label of “optimizable” in the Range Label box, then press Return or click the Apply button. The label appears in the optimizable entry in the Address Ranges list box.

4. Set up wait-state values for the optimizable memory:

   a. Click the Wait States tab.

   b. Enter values of 2 in the Read and Write boxes under “Initial,” then press Return or click the Apply button. (Leave the Burst boxes blank.) The wait-state values appear in the Address Ranges list box.

5. Select the next unlabeled range (400000 - 3FFFFFFF) listed in the Address Ranges box. Notice that the listed range values appear in the Start and End address boxes.

6. Replace the End Address value by double-clicking on the entry box and entering 4000FF.


**Note**

There are no memory instances, either Seamless CVE or otherwise, in this address range. This and the remaining steps in this section simply demonstrate how to set up memory-access ranges.

7. Click the Label tab and enter “hardware” in the Range Label entry box.

8. Click the Apply button. A new entry, at addresses 400000 - 40000FF, with the label “hardware” appears in in the Address Ranges list box. The former hardware-only range is now split into two ranges.

9. Select the next range on the list and label it “hardware2.”

10. Select the *screen* entry (3FFFFFF0 - 3FFFFFF3), which is next in the list. Although the *screen* is actually based on an optimizable register memory (hence the “optimizable” icon under the Access column), we will pretend it is hardware-only for the present. Enter a label of “screen” for this range then click Apply.

11. Select the unlabeled range at the bottom of the list in the Address Ranges list box (3FFFFFF4 to FFFFFFFF).

12. Enter “illegal” in the Range Label text-entry box, then click the Apply button.

13. Select the Access tab and then click the Illegal access button. This makes addresses 400100 to FFFFFFFF an “illegal” range, as indicated by the icon in the Address Ranges list.

14. Click the Close button to close the window. Notice that the Memory Access Ranges icon in the Session window is no longer highlighted, indicating that this step has been done.

# Saving the Configuration

Save the current configuration by performing the following steps:

1. Click the Save Configuration icon in the Seamless CVE Session window (shown below) or select the **File > Save As** menu item.


**Save Configuration**

2. In the Save As dialog box, enter a configuration filename (and path, if necessary).

3. Click Ok to save the configuration.

# Starting the Cosimulation

1. Click the **Run** button in the Seamless CVE main window. The button changes to yellow and remains pressed, indicating that cosimulation can start. In addition, a terminal window appears showing a software simulator startup message:

   **CVE NOTE: Waiting to connect to hardware process ...**

2. Go to the NC Verilog simulator's main window, activate the Navigator, and set up a waveform display window. Display the signals at the top level of the design, as well as register R30, which is one level down, in the *cpu* module.

3. Click the run button on NC Verilog main window.

   This starts the hardware simulation, but the simulation cannot run until its software begins to execute. Notice that the Run button turns green, indicating that cosimulation has begun. The Seamless CVE models in the design cause Seamless CVE to start up the software simulator. The

software simulator terminal window displays a series of messages followed by this prompt:

**(mdb)**

The Run Status window indicates that the hardware simulator is waiting (for the software simulation to start) and the software simulator is interactive (that is, waiting for you to enter a command to start the software simulation). Neither hardware nor software simulation can proceed until you respond to the software simulator prompt.


4. Enter ***run*** at the **(mdb)** prompt. The Screen window appears and displays the following message.

```
Welcome to the Seamless Tutorial!!
Using the DLX Processor and the MDB Debugger.
```

Notice that the Run Status for both of the simulators changes to “Running” then back to “Interactive” for both simulators. The software simulator runs to a breakpoint and displays a message similar to the following one (the timing may vary):

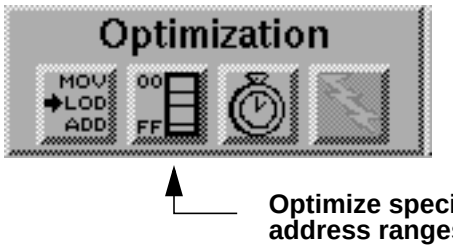
```
14 secs; 656 instr; 46 instr/sec
Hit Breakpoint 1 at 0x00000000 (begin+0x00000000)
(0x00000000) begin              nop
(mdb)
```

At this point, the run status of both simulators is interactive. In addition, waveforms appear in the hardware simulator’s waveform window, showing the bus activity that takes place during the simulation run. Values in register R30, an internal cpu register are also displayed. Display of internal registers as signals is a feature of most PSPs.

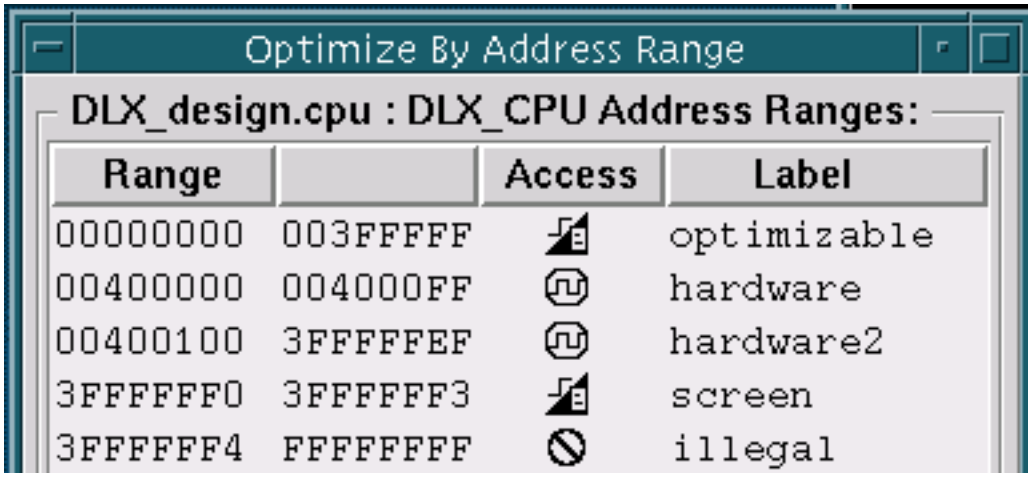
# Enabling Address-Range Optimization

Enable address-range optimization for the optimizable addresses in the processor address space by performing the following steps:

1. Click the Optimize Address Ranges icon in the Seamless CVE Session window (shown below). This activates the Optimize by Address Range dialog box.



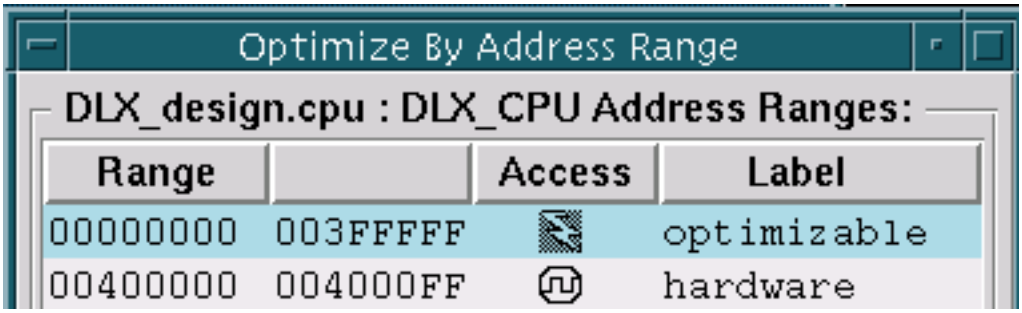

   The Address Ranges list contains the ranges you set up in the previous section, as shown below.




2. Click the range labeled “optimizable.” The start and end addresses of the range appear in the entry boxes under Optimization Entry.

3. Click the On button. This modifies the optimizable range in the Address Ranges list as shown below. The “lightning bolt” icon indicates that

processor accesses to this range can take place directly through software rather than generating hardware bus cycles.




You can enable optimized access to any portion of an optimizable address range.

4. Click the Close button to dismiss the Optimize By Address Range dialog box. The Optimize Address Ranges icon in the Seamless CVE session window is now highlighted.

5. Click the Run button in the NC Verilog simulator session window.

   This time, the simulation runs faster. In the waveform display, notice the relative lack of address- and data-bus activity during the second run. The clock runs in the hardware simulation as before. However, the software simulator now accesses memory directly except when writes to the screen, in which case it needs to generate bus activity in the hardware simulation. You can confirm this by noting that a value of 0x3FFFFFF0 is present on the address bus during these write cycles.

6. Select the **Optimize > Name Optimization** menu item. This activates the Optimization Set dialog box. This dialog box allows you to save the current optimization state for later recall.

7. Type in a name such as *opt1* in the text-entry box in the Optimization Set dialog box then click Ok to save the optimization set. Notice that the Optimization Set Enable (“lightning bolt”) icon under Optimization group

in the main Seamless CVE window becomes pressed and is highlighted yellow, meaning that a named optimization set is active.


8. Click the Optimization Set Enable icon to disable the optimization set. The icon becomes “unpressed” and is no longer highlighted. (You can also use the **Optimize > Unoptimize** menu item.) Click the icon again to re-enable the optimization set.

9. Save the current configuration by selecting the **File > Save** menu item.

# Exiting from the Cosimulation

1. Select the **File > Exit** menu item in the Seamless CVE session window or enter “exit” at the command-line. A message box appears saying “Simulator is running. Exit anyway?”

2. Click Yes to exit from the ModelSim VHDL simulator. The Seamless CVE Session window and simulator windows disappear.

# Restarting Seamless CVE using the Configuration File

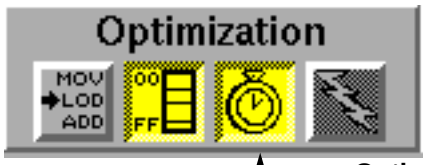
Invoke Seamless CVE using the following command:

```
$CVE_HOME/bin/cve    my_config
```

where *my_config* is the name of the configuration file you saved previously. Seamless CVE, ModelSim VHDL simulator, and software simulator all start up. Notice that the configuration in *my_config* is restored and the cosimulation is ready to start. You can confirm this by clicking on the appropriate CPU Setup and Optimization icons and checking the values that appear in the corresponding dialog boxes.

# Enabling Time Optimization

1. Click the Optimize time icon to activate Seamless time optimizations. The Time Optimizations icon button (shown below) is now “pressed” and highlighted yellow.




**Optimize time by allowing software to run ahead of hardware (using Time Optimization dialog box settings)**

2. Go to the hardware simulator’s main window and set up a waveform viewing window as you did earlier.

3. Click the run button in the hardware simulator.

   The Run button on the Seamless CVE session window turns green, indicating that cosimulation has begun.

4. Enter ***run*** at the (mdb) prompt in the software simulator’s terminal, as you did earlier in “Starting the Cosimulation.”

   Notice that the same number of instructions are executed as before, but in much less time. The software has run at full speed, decoupled from the hardware simulation, and far fewer clock cycles have occurred in the hardware simulation. The only time hardware clock cycles are required is when the program accesses the screen to print out the message. Although the tutorial program is too simple to show the full effect, usually over 99 percent of hardware clock cycles can be eliminated without affecting simulation results.

5. Select the **Optimize > Name Optimization** menu item. The Optimization Set dialog box appears.

6. Enter a name (different than the one you used previously) for the current optimization set, then click Ok.

7. Select the **Optimize > Recall Optimization** menu item. The Optimization Sets dialog box appears.

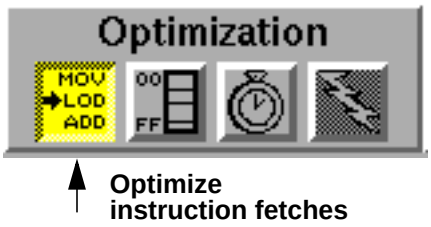
8. Select the first optimization set you saved by clicking on its name, then click the Apply button. Now select the optimization set you saved in this section and click Apply. Notice the effect on the Optimizaton icons.

9. Click the Close button to dismiss the Optimization Set dialog box.

10. Select the **Optimize > Unoptimize** menu item to clear the current optimization set.

# Enabling Instruction Fetch Optimization

1. Click in on the Instruction Fetch Optimizations icon. The icon becomes highlighted as shown below.




   Instruction fetch optimization is a convenient way to automatically enable optimizations for all instruction-fetch bus cycles. This eliminates hardware bus-cycle activity for instruction fetches, allowing the ISS to fetch the instructions directly from memory.

2. Run the hardware and software simulators as before.

3. Select the **Optimize > Unoptimize** menu item.

4. Run the hardware and software simulators again. Notice the difference in run time and bus activity with and without instruction fetch optimizations enabled.

This concludes the *Seamless CVE and ModelSim VHDL Tutorial.* You can experiment with different combinations of optimization settings to see how they

affect cosimulation. For example, try turning on optimizations for single addresses or ranges of addresses to see how they affect simulation speed and bus cycle activity.