

ClearSpeed[™]



The ClearSpeed Amber Installation Guide

July 2007

Table of contents

Table of contents	1
1 Before you start	3
1.1 Prerequisites	3
1.2 Content of patch	3
2 Installation	5
2.1 Preliminary steps	5
2.2 Installation instructions	5
2.3 Test installation	6
2.4 Using sander.CS	6
2.5 Reversing the ClearSpeed patch	7
Appendix A Details of scripts provided	9
A.1 The patch_amber9_ClearSpeed.bash script	9
A.2 The build_amber9_ClearSpeed.bash script	9
Appendix B Files added or modified by the ClearSpeed patch	11
B.1 Modified files	11
B.2 Added files host side	11
B.3 Added files on card side	12
Appendix C Building card-side code from source	13
Appendix D Notes on configuration for pgf90	15

1 Before you start

Before installing, read through this Installation Guide to ensure that the installation goes smoothly.

1.1 Prerequisites

To install the Amber patch, you need a system with:

- ClearSpeed runtime version 2.50, installed as `rpm`. Available from the ClearSpeed Support website: <http://support.clearspeed.com>
- Amber 9 software:
 - Fortran compiler:
 - Intel Fortran, ifort 9.0 or 9.1.
 - Portland Group, pgf90 version 6.1-1, 6.1-2, or 7.0-2.
- Other Fortran 90 compliant compilers may work, but have not been tested.
- Optional: ClearSpeed SDK version 2.50 installed as an `rpm`. This is only needed to recompile the card-side code. Precompiled binaries are provided in this distribution.

1.2 Content of patch

The contents of `amber_implicit_cs_1.1_${platform}.tgz` are shown in [Table 1](#). `$platform` can be any of the following: RHE3, RHE4, RHE4_64 or SLES9.3_64.

File name	Description
<code>AmberClearSpeedSetup.pdf</code>	A PDF version of this Installation Guide.
<code>bugfix.all</code>	Amber 9 patches to patch level 36. This is an unaltered copy from the AMBER web site: "http://amber.scripps.edu/bugfixes90.html"
<code>CS_amber9.patch</code>	ClearSpeed's Amber patches.
<code>amber_csboard.tgz</code>	Tarball of card-side code.
<code>patch_amber9_ClearSpeed.bash</code>	Script that patches the Amber 9 distribution, first with the Amber bug fixes, and then with ClearSpeed's modifications
<code>build_amber9_ClearSpeed.bash</code>	Script to build the modified Amber 9 version for Advance™ X620 Accelerator card.
<code>amber_cs_env.bash</code>	Script that sets up part of the environment needed to run ClearSpeed's version of <code>sander</code> . This file should also be sourced before running <code>sander.CS</code> , possibly as part of a login script.

Table 1. Contents of Amber patch

2 Installation

This chapter describes how to install the Amber 9 patch.

2.1 Preliminary steps

This patch can be installed on top of your existing Amber 9 installation. The ClearSpeed enabled version of Amber creates a new binary, `sander.CS`.

It may be easier to first try out the patch on a clean copy of Amber 9. If installing on top of an existing Amber 9 installation, it is recommended that you save a copy of the current `src/config.h` configuration file:

```
cd $AMBERHOME/src
cp config.h config.h.orig
```

If anything goes wrong when compiling the host code, the original configuration file will help you find out what went wrong.

The following must be set before running the script `patch_amber9_ClearSpeed.bash`:

- `$AMBERHOME` should point to the top directory in the Amber 9 source tree. If you are using a fresh copy of Amber 9, reset `$AMBERHOME` accordingly.
- A Fortran compiler must be in the path, either `ifort` or `pgf90`.
- An existing ClearSpeed installation. It is expected to be in `$CSHOME`. `$CSHOME` is usually set to `/opt/clearspeed/csx600_m512_le`, and defined by sourcing `$CSHOME/bin/bashrc`

The following is optional. You only need to set this when compiling the card-side code. Card-side binaries are distributed.

- `$CLEARSP_LICENSE_FILE` should point to the ClearSpeed license server.

2.2 Installation instructions

The following steps are typically run from the account that maintains `AMBER`. A detailed description of the scripts is available in [Appendix A: Details of scripts provided on page 9](#). To install the patch, follow these procedures:

1. Patch the code using the Amber 9 patch and ClearSpeed's Amber patch by moving the `amber_implicit_cs_1.1_${platform}.tgz` tarball to the Amber home directory, untar and running the setup script:

```
mv amber_implicit_cs_1.1_${platform}.tgz $AMBERHOME
cd $AMBERHOME
tar -zxvf amber_implicit_cs_1.1_${platform}.tgz
source ./patch_amber9_ClearSpeed.bash
```

2. Configure the build for the Advance X620 by adding the `-mtap` flag to a serial configuration. Look at the top of an existing `src/config.h` for previously used settings. When comparing the old settings, make the following changes, add the

```
-mtap flag, and remove all parallel flags (-mpi, -lam, -mpich, -mpich2, -openmpi,
-scali)
cd src
./configure -mtap pgf90 (as an example)
cd ..
```

If you are using `pgf90` on a 64-bit operating system, see [Appendix D: Notes on configuration for pgf90 on page 15](#).

3. Build the new version. You do not need an Advance X620 to build the software but you will need one for running the tests.

```
./build_amber9_ClearSpeed.bash
```

The build step usually finishes in less than 15 minutes. It is useful to keep the output from the `build_amber9_ClearSpeed.bash` script, for example:

```
./build_amber9_ClearSpeed.bash >& | tee build_log
```

On successful execution of the `build_amber9_ClearSpeed.bash` script, the following files will become available:

```
$AMBERHOME/exe/sander.CS
$AMBERHOME/exe/csx/gb_hi.csx
$AMBERHOME/exe/csx/gb_lo.csx
$AMBERHOME/exe/csx/gb6_hi.csx
$AMBERHOME/exe/csx/gb6_lo.csx
```

2.3 Test installation

To confirm that this build of `sander.CS` uses the Advance X620 and gives the correct results, run the tests for the Advance X620. This requires that an Advance X620 is present which may not be the case on the build machine.

```
cd test
make test.sander.CS
```

There are three tests that will finish in about 40 seconds and should report that the tests “PASSED”.

If there is no card available, an error message will be written to the screen and the test will be run on the host instead.

2.4 Using sander.CS

To set up the Clearspeed environment and extend paths to include both host and card executables, run the `amber_cs_env.bash` setup script. It may be useful to add this to a login script.

```
${AMBERHOME}/amber_cs_env.bash
```

To use the `sander` executable for the Advance X620, use `sander.CS` rather than `sander` as the executable. For example:

```
sander.CS -i mdin -o mdout -p prmtop -c inpcrd
```

`sander.CS` can use more than one Advance X620 on the host machine. The number of cards and processors are controlled through environment variables. For example, to use four processors on two cards:

```
export CS_MTAP_TOTAL_PROCS=4
```

```
export CS_MTAP_PER_CARD=2
```

The default setting is to use one card and two processors. The number of cards and processors used is written to the `sander` output file.

2.5 Reversing the ClearSpeed patch

The ClearSpeed patches can be reversed with the following command:

```
patch -p1 -R -N -r cs_reverse <CS_amber9.patch
```

This is useful if you are repeating the patch and build process to avoid spurious messages.

Appendix A Details of scripts provided

A.1 The patch_amber9_ClearSpeed.bash script

1. Check that the `$AMBERHOME` environment variable exists. Exit if not found.
2. Go to the `$AMBERHOME` directory.
3. Check that the two patch files `bugfix.all` and `CS_amber9.patch` exists. The former is the `AMBER` patch file, the latter is ClearSpeed's patch to the existing Amber source tree
4. Set up more of the environment. `./amber_cs_env.bash`
5. Apply the Amber 9 patches

```
patch -p0 -N -r patch_rejects <bugfix.allfile bugfix.all
```
6. If some of the patches have already been applied, warning messages will be seen. They can safely be ignored. For example:

```
Reversed (or previously applied) patch detected! Skipping
patch. 1 out of 1 hunk ignored -- saving rejects to file
patch_rejects patching file test/jar_multi/dist_vs_t.001.save
```
7. Apply ClearSpeed patch of host code

```
patch -p1 -N -r cs_rejects <CS_amber9.patch
```
8. Unpack card-side code

```
cd src
tar -zxvf ../amber_cs_board.tgz
```

A.2 The build_amber9_ClearSpeed.bash script

1. Check that the `$AMBERHOME` environment variable exists. Exit if not found.
2. Go to the `$AMBERHOME` directory.
3. Check that `configure` has been run with `-mtap`. Exit if that is not the case.
4. Setup additional environment - this script should be sourced or incorporated by users when using the `sander.CS` executable.

```
source ${AMBERHOME}/amber_cs_env.bash
```
5. Check that card-side executable directory exists. Exit if not found.
6. The card-side code is distributed as executables. The files are copied to the `csx` subdirectory in `${AMBERHOME}/exe` and then linked to the names that expected by `sander.CS`. For instructions on how to build the executables from source see [Appendix C](#).
7. Remove any remains of previous builds. The `make clean` step does not remove any previously built executables, the build system is set up to be used multiple times with different configurations.

```
make clean
```
8. Build host code. This last step is the time consuming one. It creates a file `sander.CS` in `$AMBERHOME/exe`.

```
make sander.CS
```


Appendix B Files added or modified by the ClearSpeed patch

B.1 Modified files

Filename	Description
src/configure	Added <code>-mtap</code> flag and alternative settings for <code>pgf90</code> to allow a 64-bit built on <code>x86_64</code> platforms
src/Makefile	Added <code>sander.CS</code> target to exclusively build <code>sander.CS</code> . (<code>sander.CS</code> is also built as part of the serial target if <code>configure</code> was run with <code>-mtap</code>)
src/sander/depend	Included the output from <code>makedepend</code> with <code>-mtap</code> configuration and the lower optimization level for <code>qm_link_atoms.f</code> .
src/sander/Makefile	Added target <code>sander.CS</code> .
src/sander/makedepend	Added support for <code>sander.CS</code> .
src/sander/qm_link_atoms.f	Lowered optimization level so that Amber's test suite pass on 64-bit builds with <code>pgf90</code> .
src/sander/force.f	Use alternative calls to <code>cs_egb</code> rather than <code>egb</code> , for the two main GB functions.
src/sander/sander.f	Added function call to check if simulation is supported and that at least one Advance X620 is available.
test/Makefile	Added target <code>test.sander.CS</code> for tests for the <code>sander.CS</code> executable.

Table 2. Files modified by ClearSpeed patch

B.2 Added files host side

Filename	Description
src/sander/cs_egb.h	Header file for GB functions in <code>cs_mtap_connections.c</code> .
src/sander/cs_egb.f	Layer that converts data structure to suit CS architecture. Handles the intercepted function calls from <code>force.f</code> .
src/sander/cs_exs_host.h	Data structures for explicit solvent simulations—not used in this version.
src/sander/cs_mtap.f	Check if simulation can be run on the Advance X620, call to initial connection to card(s).
src/sander/cs_mtap_connection.c	Communication layer between host and card(s).
test/cs/lbf7_gb/lbf7_gb1.out.save	Baseline for GB 1 test.

Table 3. Files added by ClearSpeed patch

Filename	Description
test/cs/1bf7_gb/1bf7_gb5.out.save	Baseline for GB 5 test.
test/cs/1bf7_gb/1bf7_gb6.out.save	Baseline for GB 6 test.
test/cs/1bf7_gb/inpcrd	Input coordinates for the GB 1, GB 5 and GB 6 tests.
test/cs/1bf7_gb/prmtop	Topology and parameters for the GB 1, GB 5 and the GB 6 tests.
test/cs/1bf7_gb/Run.1bf7_gb1	Script to run test and compare results with baseline for the GB 1 model.
test/cs/1bf7_gb/Run.1bf7_gb5	Script to run test and compare results with baseline for the GB 5 model.
test/cs/1bf7_gb/Run.1bf7_gb6	Script to run test and compare results with baseline for the GB 6 model.

Table 3. Files added by ClearSpeed patch

B.3 Added files on card side

Filename	Description
src/amber_cs/explicit_600/cs_exs_defs.h	Constants for explicit solvent code shared between host and card—not used in this version.
src/amber_cs/gb_600/cs_gb.cn	Source code GB1, 2, & 5 models.
src/amber_cs/gb_600/cs_gb_compthread.is	Redefines stack sizes.
src/amber_cs/gb_600/cs_gb_defs.h	Constants & structs shared between host and card GB.
src/amber_cs/gb_600/cs_gb.h	GB declarations.
src/amber_cs/gb_600/cs_igb6.cn	GB 6 frame work and outer loop.
src/amber_cs/gb_600/cs_igb6inner.cn	GB 6 inner loop.
src/amber_cs/gb_600/makefile	GB makefile.
src/amber_cs/gb_600/reduce_sum_djk.h	Reduction function in assembler.
src/amber_cs/gb_600/swazzle_tuned.inc	Additional code for reduction.

Table 4. Files added on card side

Appendix C Building card-side code from source

The card-side code is provided in two formats: as precompiled binaries and as source code. The binaries are installed by running the `build_amber9_ClearSpeed.bash` script. The following instructions are for building the same executables from source. The executables have different names to distinguish between the provided binaries and the binaries created by the build procedure. In addition, links are used to point to the versions that are used at runtime.

```
cd ${AMBERHOME}/src/amber_cs/gb_600
```

```
make
```

```
make install
```

The executables shown in [Table 5](#) will be created.

Filename	Description
<code>gbcn_lo.csx</code>	Executable for GB models 1,2, and 5 for chip 0.
<code>gbcn_hi.csx</code>	Executable for GB models 1,2, and 5 for chip 1.
<code>gb6cn_lo.csx</code>	Executable for GB model 6 for chip 0.
<code>gb6cn_hi.csx</code>	Executable for GB model 6 for chip 1.

Table 5. Executables created from source code

Appendix D Notes on configuration for pgf90

The configure script distributed with Amber 9, enforces a 32-bit build for the pgf90 compiler.

On 64-bit operating systems, ClearSpeed's libraries and runtime environment is set up for 64-bit. The configure script distributed with ClearSpeed's patch sets up a 64-bit build on 64-bit operating systems for pgf90. In order to get the build to pass the Amber test suite, the optimization level was lowered for the file `src/sander/qm_link_atoms.f`.

The default installation for the pgf90 compiler has also been changed. It used to be `/usr/pgi` and it is now `/opt/pgi`. The configure script has been made more flexible so that it can find the associated BLAS and LAPACK libraries. Previously, the libraries were expected to be in `/usr/pgi/linux86/lib`.

ClearSpeed Technology, Inc.

3031 Tisch Way, Suite 200
San Jose, CA 95128
United States of America

Tel: +1 408 557 2067
Fax: +1 408 557 9054

Email: info@clearspeed.com

Web: <http://www.clearspeed.com>

Support: <http://support.clearspeed.com>

ClearSpeed Technology plc

3110 Great Western Court
Hunts Ground Road
Bristol BS34 8HP
United Kingdom

Tel: +44 (0)117 317 2000
Fax: +44 (0)117 317 2002

Acknowledgments:

Linpack: This product includes software developed at the University of Tennessee, Knoxville, Innovative Computing Laboratories

1. Information and data contained in this document, together with the information contained in any and all associated ClearSpeed documents including without limitation, data sheets, application notes and the like ('Information') is provided in connection with ClearSpeed products and is provided for information only. Quoted figures in the Information, which may be performance, size, cost, power and the like are estimates based upon analysis and simulations of current designs and are liable to change.
2. Such Information does not constitute an offer of, or an invitation by or on behalf of ClearSpeed, or any ClearSpeed affiliate to supply any product or provide any service to any party having access to this Information. Except as provided in ClearSpeed Terms and Conditions of Sale for ClearSpeed products, ClearSpeed assumes no liability whatsoever.
3. ClearSpeed products are not intended for use, whether directly or indirectly, in any medical, life saving and/ or life sustaining systems or applications.
4. The worldwide intellectual property rights in the Information and data contained therein is owned by ClearSpeed. No license whether express or implied either by estoppel or otherwise to any intellectual property rights is granted by this document or otherwise. You may not download, copy, adapt or distribute this Information except with the consent in writing of ClearSpeed.
5. The system vendor remains solely responsible for any and all design, functionality and terms of sale of any product which incorporates a ClearSpeed product including without limitation, product liability, intellectual property infringement, warranty including conformance to specification and or performance.
6. Any condition, warranty or other term which might but for this paragraph have effect between ClearSpeed and you or which would otherwise be implied into or incorporated into the Information (including without limitation, the implied terms of satisfactory quality, merchantability or fitness for purpose), whether by statute, common law or otherwise are hereby excluded.
7. ClearSpeed reserves the right to make changes to the Information or the data contained therein at any time without notice.

© Copyright ClearSpeed Technology plc 2007. All rights reserved.

Advance is a registered trademark of ClearSpeed Technology plc

ClearSpeed, ClearConnect, Advance and the ClearSpeed logo are trade marks or registered trade marks of ClearSpeed Technology plc. All other brands and names are the property of their respective owners.