

PREPARED BY: Robert Milkovits	<div data-bbox="540 233 1010 373" data-label="Image"> </div>	DOCUMENT NUMBER: NPB PS-0419
APPROVALS: SEE QSI FOR ELECTRONIC APPROVALS	<div data-bbox="375 455 1169 493" data-label="Text"> 4321 Jamboree Road, Newport Beach, CA 92660-3095 </div> <div data-bbox="526 609 1018 758" data-label="Text"> <p>Proprietary Information No Dissemination Or Use Allowed Without Prior Written Permission</p> </div>	<div data-bbox="1192 491 1421 527" data-label="Text"> REVISION: 05 </div> <div data-bbox="1192 567 1484 604" data-label="Text"> DATE: 09/22/2008 </div> <div data-bbox="1192 609 1396 646" data-label="Text"> Page 1 of 28 </div> <div data-bbox="1192 651 1408 760" data-label="Text"> TYPE OF DOCUMENT: Design Kit </div>
<div data-bbox="716 840 824 875" data-label="Text"> TITLE: </div> <div data-bbox="498 915 1045 991" data-label="Text"> <p>Design Kit Installation Instructions: Jazz Cadence PDK</p> </div> <div data-bbox="378 970 1336 1260" data-label="Text"> <p>TowerJazz Confidential Downloaded by: Sanjay Raman Date: 08/14/2012 08:14 IP: 128.173.89.96</p> </div>		

REV	Revision Description	Date
01	Initial release in QSI.	07/23/2003
02	Updated sections 2, 3, 4: Section 2: Added Linux and IC50, Added section numbers, reworded tool list Section 3: Reworded description Section 4: Cleaned up formatting	08/08/2003
03	Updated sections 2, 3, 4, 6, 7: Section 2: Updated minimum tool version requirements Section 3: Reworded description Section 4: Updated Ebizz site, Created table of needed information, Updated vendor websites and minimum versions, Corrected Assura sample Section 6: Added preferred kit switching method Section 7: Added new section for advanced setup	01/21/2005
04	Updated sections 2, 3, 4, 6, 7: Section 2: Remove support for HPUX; Updated minimum tool version requirements and vendor websites Section 3: Reference NPB-PS-0404 Section 4: Add new section for "superset" kit process variant selection; Update kit names/versions Section 7: Update sample shell scripts for advanced installation	05/01/2007
05	Updated sections 2, 3, 4 Section 2: Add details on CDBA versus OA support; Update minimum tool version requirements Section 3: Update documentation list Section 4: Add MMSIM, rename LDV to IUS; Add release notes as reference; Update vendor URLs	09/22/2008

Jazz Semiconductor		DOCUMENT NUMBER: NPB PS-0419	
	PROPRIETARY INFORMATION	REVISION: 05	PAGE 3 OF 28
TABLE OF CONTENTS			
1	OVERVIEW		5
2	REQUIREMENTS		6
2.1	UNIX OPERATING SYSTEM REQUIREMENT:		6
2.2	OPERATING SYSTEM PATCHES.....		6
2.3	MINIMUM TOOL REQUIREMENTS:.....		6
2.4	OPTIONAL TOOL REQUIREMENTS:		6
2.5	MINIMUM LICENSE FEATURE REQUIREMENTS:		7
2.6	OPTIONAL LICENSE FEATURE REQUIREMENTS:		7
2.7	SYSTEM/UNIX REQUIREMENTS:		8
2.7.1	<i>Kit installation directory:.....</i>		8
2.7.2	<i>License setup for all needed software tools.....</i>		8
2.7.3	<i>Perl programming language version 5.....</i>		8
2.7.4	<i>Gzip compression tool</i>		8
2.7.5	<i>Supported shells.....</i>		8
3	DOCUMENTATION		9
4	INSTALLATION PROCEDURE.....		10
4.1	INFORMATION NEEDED		10
4.2	CREATE INSTALLATION DIRECTORY.....		11
4.3	DOWNLOAD THE KIT INSTALLATION FILES		11
4.4	UNPACK DESIGN KIT		11
4.5	LOCATE YOUR TECHNOLOGY LIBRARY ("TECHFILE")		11
4.6	LOCATE YOUR PROCESS VARIANT OR VERIFICATION TECHNOLOGY ("TECHVER")		12
4.7	CREATE A ROOT LOCATION FOR ALL OF YOUR JAZZ DESIGN PROJECTS ("PROJECTS").....		12
4.8	CREATE AND CUSTOMIZE INDIVIDUAL USER JAZZ.CSHRC		12
4.8.1	<i>Create the JAZZ.cshrc file</i>		12
4.8.2	<i>Cadence IC Tools.....</i>		12
4.8.3	<i>Cadence MMSIM Tools</i>		13
4.8.4	<i>Agilent ADS RFDE or Dynamic Link.....</i>		13
4.8.5	<i>Mentor Graphics Calibre.....</i>		13
4.8.6	<i>Cadence Assura tools</i>		14
4.8.7	<i>Cadence IUS Tools</i>		14
4.8.8	<i>Custom Spectre simulation directory</i>		14
4.9	MAKE A PROJECT TREE		15
4.9.1	<i>Go to project directory</i>		15
4.9.2	<i>Determine project name</i>		15
4.9.3	<i>Create project user list.....</i>		15
4.9.4	<i>Make project directory tree</i>		15
4.9.5	<i>Examine the files and directories created.....</i>		15
4.10	LAUNCH CADENCE		16
4.11	ADDITIONAL SETUP REQUIRED FOR "SUPERSET" PROCESS VARIANTS		16
4.12	ADDING USERS TO AN EXISTING PROJECT		18
4.12.1	<i>Update project user list.....</i>		18

Jazz Semiconductor		DOCUMENT NUMBER: NPB PS-0419	
	PROPRIETARY INFORMATION	REVISION: 05	PAGE 4 OF 28
4.12.2	Create another project user list.....	18	
5	TROUBLESHOOTING	20	
5.1	DESIGN KIT EXPIRATION	20	
5.2	WRAPPER SCRIPTS.....	21	
6	UPDATING AN EXISTING KIT AND/OR PROJECT DIRECTORY.....	22	
6.1	KIT UPGRADE — SAME PROCESS TECHNOLOGY AS PREVIOUS KIT	22	
6.1.1	Install new design kit	22	
6.1.2	Using the new design kit.....	22	
6.2	ADDITIONAL KIT — NEW PROCESS TECHNOLOGY	24	
6.3	ADDITIONAL OPTIONS FOR THE CAD ADMINISTRATOR	24	
7	ADVANCED KIT INSTALLATION OPTIONS FOR THE CAD ADMINISTRATOR	26	
7.1	ADVANCED KIT INSTALL SHELL SCRIPT.....	26	
7.2	ADVANCED KIT INSTALL CDS.LIB CONTENTS	27	
7.3	ADVANCED KIT INSTALL .CDSINIT FILE CONTENTS	27	
7.4	SETTINGS REQUIRED FOR THE JAZZ CALIBRE INTEGRATION.....	27	
7.4.1	What variables need to be set?	27	
7.4.2	What directories must exist?.....	28	
7.4.3	How to change the variables?	28	

TowerJazz Confidential
Downloaded by: Sanjay Raman
Date: 08/14/2012 08:14
IP: 128.173.89.96

1 Overview

The Jazz Cadence AMS design kit provides the following functionality:

- Components (fet, resistor, cap, etc.) with symbol, layout (programmable “pcell” or fixed cell), and simulation models
- Schematic capture using Cadence Virtuoso Schematic entry (Composer)
- Simulation using Cadence Analog Design Environment (“ADE”) and Spectre
- Layout using Cadence Virtuoso and VirtuosoXL
- Layout verification (drc, lvs, erc, stress, etc.) using Mentor Graphics Calibre and RVE, or using Cadence Assura DRC/LVS (for select processes)
- Layout parasitic extraction (pex) and Back-Annotation/Post-Layout simulation in Cadence Virtuoso with extraction performed by Mentor Xcalibre, Mentor XRC (select processes) or Cadence Assura QRC or RCX (select processes)

TowerJazz Confidential
Downloaded by: Sanjay Raman
Date: 08/14/2012 08:14
IP: 128.173.89.96

Jazz Semiconductor	DOCUMENT NUMBER: NPB PS-0419		
	PROPRIETARY INFORMATION	REVISION: 05	PAGE 6 OF 28
<p>2 Requirements</p> <p>2.1 UNIX Operating System requirement:</p> <p>System requirements are based on EDA tool vendor OS requirements. Please consult with the tool vendors for more details on OS and platform requirements.</p> <ul style="list-style-type: none"> • Sun/Solaris 5.8 or later • IA32/Red Hat Enterprise Linux 3.0 <p>2.2 Operating System Patches</p> <p>In order for the Jazz design kit to work properly, you must have all the Operating System Patches required by Cadence and other EDA tool vendors. Cadence supplies a command called <code>checkSysConf</code> that can be used to determine if the required patches are installed. Cadence recommends you download the latest version of the <code>checkSysConfig</code> command from the Cadence SourceLink website:</p> <p>http://sourcelink.cadence.com/docs/files/download/sys_conf_check/</p> <p>Documentation on how to use <code>checkSysConfig</code> is provided by Cadence. A sample command line is below:</p> <pre>checkSysConf [CadenceRelease] -d [OSPatchDir]</pre> <p>You will also need to check with other tool vendors to verify that you have the appropriate Operating System patches installed for other tools that you use.</p> <p>2.3 Minimum Tool requirements:</p> <p>The Jazz Cadence AMS Design Kit supports either the Cadence software release IC5.1.41 for CDBA (Cadence Database architecture) based design kits, or the Cadence software release IC6.1.2 and IC6.1.3 for OA (Open Access) based design kits. Version IC5.1 USR5 or later is recommended for CDBA design kits, and version IC6.1.2 ISR 14 (6.1.2.500.14) or later is recommended for OA design kits. You can determine your Cadence software version by typing "<code>icfb -w</code>" at a Unix command prompt.</p> <p>Jazz provides the CDBA based design kits as separate downloads from the OA based design kits. You must obtain the correct Jazz Design Kit to support the Cadence database format that you use in your design environment (IC5.1/CDBA or IC6.1/OA). Not all Jazz Design Kits are available for both database formats. Please review the Design Kit release notes, or consult with your Jazz Support representative if you have questions about the database format supported for a particular design kit.</p> <p>The tool version requirements listed above are for new design kits released after August 2008. Please review the Design Kit release notes and any Design Kit Patches to find the specific software versions supported by the design kit you use.</p> <p>2.4 Optional Tool requirements:</p> <ul style="list-style-type: none"> • Mentor Graphics Calibre Interactive/XRC version v2008.01 or later for layout verification (including DRC, LVS, and PEX) using the Mentor Calibre Interactive with Calibre XRC / CalibreView flow on selected process variants only. 			

- Cadence Assura version 3.1.7 or later for DRC and LVS, and Cadence QRC 7.1 for extraction. Assura and QRC are supported only on selected process variants.
- Cadence IUS 5.8 or later for Spectre-Verilog cosimulation or AMS Designer simulation.
- Agilent ADS RFDE or Dynamic Link v2008A or later.
- Mentor Graphics Calibre/XCalibre version 2003.03 or later for layout verification (including DRC, LVS, PEX, RVE) on selected process variants only. For parasitic extraction, the last Mentor product release to ship with the Xcalibre software is the 2003.04 release.

The tool version requirements listed above are for new design kits released after August 2008. Please review the Design Kit release notes to find the specific software versions supported by the design kit you use.

2.5 Minimum License feature requirements:

The Jazz Design Kit requires certain third-party EDA tool features to function. These third-party tools and licenses are not provided by Jazz and must be obtained directly from the tool vendor. Here is a list of the product vendor and the features required. Many, but not all, of these features have a direct correspondence to the product name or product number with the tool vendor.

Cadence: Composer schematic editor, Analog Artist, Spectre, Virtuoso, VirtuosoXL

Mentor Graphics: Calibre and XCalibre

calibredrc, calibrehdrc (DRC)

calibrelvs, calibrehlvs (LVS)

calibreqdb (RVE)

calibrexrc (Mentor Calibre Interactive PEX)

xcalibrec, xcalibrerc (Jazz PEX)

2.6 Optional License feature requirements:

These additional optional features may be required depending on your design flow:

Cadence: Spectre RF, Analog Artist Statistics, Analog Artist Mixed-Signal Interface, Assura DRC/LVS (selected process variants only), Cadence QRC Extraction (selected process variants only), Analog Artist HspiceS Interface, Analog Artist OASIS (if using Agilent ADS RFDE/Dynamic Link)

Cadence divaLVS is required for the Jazz PEX flow using Xcalibre for parasitic extraction

Synopsys: Hspice and Hspice PSF Writer for Analog Artist

Mentor Graphics: Calibre and XRC (selected process variants only). The Cadence DivaLVS license is not needed for the Mentor CalibreView PEX flow using XRC for parasitic extraction.

calibredrc, calibrehdrc (DRC)

calibrelvs, calibrehlvs (LVS)

calibreqdb (RVE)

calinteractive (Calibre Interactive GUI)

calibrexrc, calibrexrc (CalibreView PEX)

Jazz Semiconductor	DOCUMENT NUMBER: NPB PS-0419		
	PROPRIETARY INFORMATION	REVISION: 05	PAGE 8 OF 28
<p>2.7 System/UNIX requirements:</p> <p>2.7.1 <u>Kit installation directory:</u> The design kit must be installed in one central location with a consistent mount point available for all users across all hosts.</p> <p>2.7.2 <u>License setup for all needed software tools</u> All needed software and licenses must be set up and operational before installing the Jazz design kit.</p> <p>2.7.3 <u>Perl programming language version 5</u> For Solaris 8 or later – Already installed by Operating System For Linux – Already installed by Operating System</p> <p>2.7.4 <u>Gzip compression tool</u> For Solaris 8 or later – freely available at http://www.sunfreeware.com For Linux – Already installed by Operating System</p> <p>2.7.5 <u>Supported shells</u> The Jazz design kit supports the csh and tcsh Unix shells only. The design kit will not function properly with any other shell when using the standard installation procedure. If you use a different Unix shell, please review the contents of Section 7 of this document.</p>			

TowerJazz Confidential
Downloaded by: Sanjay Raman
Date: 08/14/2012 08:14
IP: 128.173.89.96

3 Documentation

The Jazz design kit includes several pieces of documentation that provide an overview of the design kit and components included in the kit. Here is a list of important documents and their location. Several documents are only available once the kit is installed and in use.

Jazz Ebizz Website: The Jazz Ebizz website provides you with access to the latest design kit, documentation, specifications, and design kit patches. You can also submit helpticket requests and submit tapeout data through the Jazz Ebizz website.

- <https://online.jazzsemi.com>

Design Kit Release Notes: Information on changes contained in the latest design kit release as well as the past history of this design kit. Also contains: a list of software versions used to develop and test the design kit, and the process variants supported by the design kit.

- Available through the Jazz eBizz website
- Available from the Cadence CIW using the Jazz Custom Menu:
JAZZ -> Application Notes -> Release Note
- Jazz Cadence AMS Design System Overview NPB-PS-0404: describes the Jazz design kit contents and the design environment. It includes project setup and a simple walkthrough of schematic capture, simulation, layout, verification, and post-layout simulation using the Jazz Calibre flow.
- Available through Jazz eBizz website
- Available from the Cadence CIW using the Jazz Custom Menu:
JAZZ -> Application Notes -> Jazz Design Kit Overview

For details on using Mentor Calibre Interactive or Cadence Assura, please see the documentation provided by the tool vendor.

Technology Component Info: Device/component descriptions in html format

- Available from the Cadence CIW using the Jazz Custom Menu:
JAZZ -> Technology Component Info (VCSD Web)

Jazz Knowledge Base: Frequently asked questions and issues

- Available through Jazz eBizz website

FAQ: Frequently asked questions

- Available from the Cadence CIW using the Jazz Custom Menu:
JAZZ -> Application Notes -> FAQ

Additional documents: Espec (Electrical Specifications), Design Rules, Modeling Design Manual, and Design Application Notes.

- Available through Jazz eBizz website

Jazz Semiconductor		DOCUMENT NUMBER: NPB PS-0419	
PROPRIETARY INFORMATION		REVISION: 05	PAGE 10 OF 28

4 Installation Procedure

In the following documentation, the symbol % designates the command prompt, words written in a typeface like `this` refer to your computer, and the symbols [] with an underline mean you need to substitute a value, for example [my file].

4.1 Information Needed

The following information will be needed. Fill out this table with the proper information as you work through the installation instructions:

Information needed	Section	Value
kit dir	4.2	
Jazz eBizz login username and password	4.3	
techname	4.3	
techfile	4.5	
techver	4.6	
projects directory	4.7	
JAZZ.cshrc file location	4.8.1	
Full path to perl executable	4.8.1	
Cadence IC Install Dir	4.8.2	
Cadence MMSIM Install Dir	4.8.3	
Agilent ADS Install Dir	4.8.4	
Mentor Calibre install dir	4.8.5	
Mentor Calibre Install Dir	4.8.5	
Cadence Assura install dir	4.8.6	
Cadence IUS install dir	4.8.7	
project name	4.9.1	
Process Variant Settings	4.11	

4.2 Create Installation Directory

Create a directory for the design kit to be installed in that is accessible to all the users of the design kit. This kit dir could be the name of the process family, for example, SiGe60, the name of the design kit, for example, sbc35x, or the specific design kit version, for example sbc35x_20061220. Do not install the new design kit over the top of an existing Jazz design kit installation.

You will need to include the whole path of the `[kit_dir]` starting from the root level (the "/" level of the disk). For example, you may name the kit directory "sbc35x_20061220", but the full path you will use in the installation instructions is `/tools/jazz_kits/sbc35x_20061220`. Here are the steps to create this kit installation directory:

```
% mkdir [kit_dir]
% setenv KIT_DIR [kit_dir]
% cd $KIT_DIR
```

4.3 Download the kit installation files

The Jazz design kit is shipped in compressed (gzip) tar file format and named according to the convention:

```
techname_yyyymmdd_hotcode.tar.gz
```

where techname is the name of the technology library (e.g., sbc35x, sbc18, ch25a, ca18, bcd25 etc.), and yyyymmdd is the four-digit year, the month, and the date of kit release.

Retrieve the kit from the Jazz eBizz File Exchange Site using your Internet browser. For example, the following command will open the eBizz site using Netscape or Mozilla. You will then need to log in to the Jazz eBizz site, navigate to the File Exchange file area and download the appropriate design kit.

```
% netscape http://online.jazzsemi.com
```

4.4 Unpack design kit

The design kit must be uncompressed and untared in the `[kit_dir]` directory. Be sure you are in the `[kit_dir]` directory before performing this step. The uncompress and untar can be done in one step using the following command:

```
% gunzip < [techname_yyyymmdd_hotcode.tar.gz] | tar -xf -
```

4.5 Locate your technology library ("techfile")

Identify which technology library you will be using. This is not always the same name as the "techname" that is part of the design kit name. You will need to use the name of the directory, not the full directory path. For example, use the name sbc35x and not the full path

```
/tools/jazz_kits/sige60/HOTCODE/amslibs/cds_default/cdslibs/sbc35x.
```

This command lists the available technology libraries in a Design Kit:

```
% ls $KIT_DIR/HOTCODE/amslibs/cds_default/cdslibs
```

The `[techfile]` will have a name that is the same as or similar to the "techname", for example, sbc18, sbc35x, ch25a, ca18, bcd25 etc. This name will be used in step 4.9.

4.6 Locate your process variant or verification technology (“techver”)

You will also need to identify the process variant (the verification technology or “techver”) that you will be using. This is usually the name of the specific process technology or process variant you will be designing into, for example, `sbc35qtx` or `sbc18hx`. In Design Kits that support “superset” process variants, you will use the “parent” process variant that contains all of the process options. The possible process variants supported by the kit are in the directory `$KIT_DIR/HOTCODE/techs` and are also documented in the Design Kit release notes. Use this command to list the available process variants supported by the kit. Use the name of the directory, not the full path for the `[techver]`.

```
% ls $KIT_DIR/HOTCODE/techs
```

The `[techver]` will have a name that is similar to the “techname”, for example, `sbc18hx`, `sbc18qtd` or `sbc35qtf`. This name will be used in step 4.9.

4.7 Create a root location for all of your Jazz design projects (“projects”)

Create a location for all of your design projects. This can be any directory that is accessible to all the users and is outside the `$KIT_DIR/HOTCODE` directory. This `[project]` directory will be the top-level and all projects you start will be created below this directory. You will need to use the full path for this directory.

```
% mkdir [projects]
```

4.8 Create and customize individual user JAZZ.cshrc

The `JAZZ.cshrc` file is used to set up the design kit environment. Each time you work in the design kit, you will need to source this file. Every user will need access to a copy of this file.

4.8.1 Create the JAZZ.cshrc file

Create a file called `JAZZ.cshrc` in your home directory or some other convenient location. These instructions assume you will place the file in your home directory, but the file can also be placed in a location that is accessible to all users:

```
% vi ~/JAZZ.cshrc
```

Copy the following lines into the `JAZZ.cshrc` file. You will need to customize the lines by replacing the items with underlines with a complete directory path. The order in which these commands are executed is important.

```
#!/bin/csh -f
setenv RDS_ROOT [kit_dir]/HOTCODE
setenv PERL5 [full path to perl executable]
source $RDS_ROOT/etc/RDS.cshrc
source $RDS_ROOT/etc/cdsDesKit.cshrc
source $RDS_ROOT/etc/cdsSystem.cshrc
setenv PROJ_ROOT [projects]
```

4.8.2 Cadence IC Tools

The Jazz Design Kit supports Cadence's IC tools like Virtuoso. You must install and test these tools before installing the Jazz kit. This includes setting all required environment variables and paths that are needed by Cadence's IC tools. If you already have the Cadence IC tools (`icfb`, `icms`, etc.) available in your path, you can skip this step. If you do not have the Cadence IC tools available in

your path, you will need to follow the installation instructions provided by Cadence. Documentation on installing and configuring the Cadence IC tools can be found on the Cadence Sourcelink website:

http://sourcelink.cadence.com/en/infomgmt/DisplayStaticLink.jhtml?/docs/files/Release_Info/Docs/Landing/Rel_List_Full.html

As an example of the setup required by the Cadence IC tools, you can add and customize these lines into the `JAZZ.cshrc` file:

```
set cadence_root=[Cadence IC install dir]
set path = ( ` $prepend $cadence_root/tools/dfII/bin $path` )
set path = ( ` $prepend $cadence_root/tools/bin $path` )
```

4.8.3 Cadence MMSIM Tools

The Jazz Design Kit supports Cadence's MMSIM tools like Spectre. You must install and test these tools before installing the Jazz kit. This includes setting all required environment variables and paths that are needed by Cadence's IC tools. If you already have the Cadence MMSIM tools (`spectre`, etc.) available in your path, you can skip this step. If you do not have the Cadence MMSIM tools available in your path, you will need to follow the installation instructions provided by Cadence. Documentation on installing and configuring the Cadence MMSIM tools can be found on the Cadence Sourcelink website:

http://sourcelink.cadence.com/en/infomgmt/DisplayStaticLink.jhtml?/docs/files/Release_Info/Docs/Landing/Rel_List_Full.html

As an example of the setup required by the Cadence MMSIM tools, you can add and customize these lines into the `JAZZ.cshrc` file:

```
set mmsim_root=[Cadence MMSIM install dir]
set path = ( ` $prepend $mmsim_root/tools/dfII/bin $path` )
set path = ( ` $prepend $mmsim_root/tools/bin $path` )
```

4.8.4 Agilent ADS RFDE or Dynamic Link

The Jazz Design Kit supports Agilent's RFDE or Dynamic Link integration with the Cadence Virtuoso software. You must install and test RFDE before installing the Jazz kit. This includes setting all required environment variables and paths that are needed by Agilent ADS RFDE. Documentation on installing and configuring Agilent ADS with Cadence can be found in the ADS installation manual, or at Agilent's website:

<http://edocs.soco.agilent.com/display/ads2008/Home>

Jazz recommends executing the Agilent-provided `rfde` script in place of executing `icfb` when you are starting the Cadence Virtuoso executable with Agilent integration.

4.8.5 Mentor Graphics Calibre

The Jazz Design Kit supports Mentor's verification tools Calibre and XCalibre. You must install and test these tools before installing the Jazz kit. This includes setting all required environment variables and paths that are needed by the Mentor verification tools. If you already have the Mentor tools (`calibre`, etc.) available in your path, you can skip this step. If you do not have the Mentor verification tools available in your path, you will need to follow the installation instructions provided by Mentor. Documentation on installing and configuring the Mentor verification tools can be found on the Mentor SupportNet website:

http://www.mentor.com/products/ic_nanometer_design/lv/calibre_drc/

As an example of the setup required by the Mentor verification tools, you can add and customize these lines into the `JAZZ.cshrc` file. The directory `$MGC_HOME` should contain the directory `bin/calibre`:

```
setenv MGLS_LICENSE_FILE [path to your mentor license file]
setenv MGC_HOME [Mentor Calibre install dir]
set path = ( ` $prepend $MGC_HOME/bin $path` )
```

4.8.6 Cadence Assura tools

The Jazz Design Kit supports Cadence's Assura Verification tools in most process technologies. You must install and test these tools before installing the Jazz kit. This includes setting all required environment variables and paths that are needed by Cadence's Assura Verification tools. If you already have the Cadence Assura Verification tools (`assura`, etc.) available in your path, you can skip this step. If you do not have the Cadence Assura Verification tools available in your path, you will need to follow the installation instructions provided by Cadence. Documentation on configuring the Cadence Assura Verification tools can be found on the Cadence Sourcelink website:

http://sourcelink.cadence.com/en/infomgmt/DisplayStaticLink.jhtml?/docs/files/Release_Info/Docs/Landing/Rel_List_Full.html

As an example of the setup required, you can add and customize these lines into the `JAZZ.cshrc` file:

```
setenv ASSURAHOME [Cadence Assura install dir]
set path = ( ` $prepend $ASSURAHOME/tools/assura/bin $path` )
set path = ( ` $prepend $ASSURAHOME/tools/bin $path` )
```

4.8.7 Cadence IUS Tools

The Jazz Design Kit supports Cadence's Incisive Unified Simulator (IUS) tools. You must install and test these tools before installing the Jazz kit. This includes setting all required environment variables and paths that are needed by Cadence's IUS tools. If you already have the Cadence IUS tools (`verilog`, `verilog.vmx`, etc.) available in your path, you can skip this step. If you do not have the Cadence IUS tools available in your path, you will need to follow the installation instructions provided by Cadence. Documentation on configuring the Cadence IUS tools can be found on the Cadence Sourcelink website:

http://sourcelink.cadence.com/en/infomgmt/DisplayStaticLink.jhtml?/docs/files/Release_Info/Docs/Landing/Rel_List_Full.html

As an example of the setup required, you can add and customize these lines into the `JAZZ.cshrc` file:

```
setenv IUS_INST_DIR [Cadence IUS install dir]
set path = ( ` $prepend $IUS_INST_DIR/tools/bin $path` )
```

4.8.8 Custom Spectre simulation directory

This optional step is needed only if you wish to change the location where Spectre simulation results are stored. By default simulation results will be stored in the `simulation` directory inside your current project work directory. If you want to change this result directory for Spectre simulation results, add and customize this line into the `JAZZ.cshrc` file.

```
setenv RDS_ASI_DIR [spectre simulation dir]
```


4.9 Make a project tree

You must use the Jazz-provided scripts to create the project trees where all of the design team members will work.

4.9.1 Go to project directory

Set up the Jazz design kit and go to the top of the project tree you selected earlier by executing these steps:

```
% source ~/JAZZ.cshrc
% cd $PROJ_ROOT
```

4.9.2 Determine project name

Select a project name to be used for this new project. For example, it may be the design name, the project team name, or some other unique identifier. This name will be used later in the instructions when you are asked for the [project_name] value.

4.9.3 Create project user list

Create a user list for your new project that contains each user login name with one user per line. You can use these commands below, or you can edit a file with your favorite text editor. The commands below will create a file named `user_list` that lists the two users, `user1` and `user2`:

```
% echo user1 > user_list
% echo user2 >> user_list
```

4.9.4 Make project directory tree

Use the `makeProjectTree` script to create the files and directory structure for the new project. This command must appear all on one line, even though it may be printed here on multiple lines:

```
% makeProjectTree [project_name] -users user_list -cds -tech
[techfile] -tech_ver [techver] -ver
```

You must add the option `-tech_ver [techver]` to specify the process variant (verification technology) that the project will use. For example, in SiGe60 using the `sbc35x` design kit, you could specify `-tech_ver sbc35qtx`. For SiGe120 using the `sbc18` design kit, you could specify `-tech_ver sbc18hxl`.

If you use Cadence Assura, you must also specify `-assura` to create the Assura `assura_tech.lib` file.

You can type the following command with the `-h` option to display all the available options for `makeProjectTree`:

```
% makeProjectTree -h
```

4.9.5 Examine the files and directories created

- The `work_libs` directory contains a directory for each project user. Inside each user directory is a directory named `cds` which is the working directory for Cadence sessions. The `cds` directory contains a `cds.lib` file and a `.cdsinit` file. These initialize the project specific library components. The `cds` directory also contains the `design` directory, which is the Cadence library directory for the specific user. This library appears in the Cadence Library Manager as [project_name]_[user], for example, as "jazzProj_jdoe".

Jazz Semiconductor	DOCUMENT NUMBER: NPB PS-0419		
	PROPRIETARY INFORMATION	REVISION: 05	PAGE 16 OF 28
<ul style="list-style-type: none"> The <code>cds_master</code> directory contains the project-wide settings in the files <code>cdsPrj.cshrc</code> and <code>project.lib</code>. It also contains the <code>design</code> directory, which is the Cadence library directory shared between all users of the project. This library appears in the Cadence Library Manager as <code>[project_name]_master</code>, for example, as "jazzProj_master". 			
<h4>4.10 Launch Cadence</h4>			
<p>An alias <code>cdsprj</code> is defined when you source the <code>JAZZ.cshrc</code> file in section 4.9.1. This alias sources your project specific <code>cdsUsr.cshrc</code> and changes your working directory to be your user-specific <code>cds</code> directory within your project. To begin working in Cadence, type the following:</p>			
<pre>% source ~/JAZZ.cshrc % cdsprj [project name] % icfb</pre>			
<h4>4.11 Additional Setup required for “superset” process variants</h4>			
<p>Some Jazz Design Kits provide a “superset” process variant selection. Each “superset” process variant provides all possible compatible process layers and Design Kit features for a specific number of metal layers. A full list of process variants supported by the design kit is documented in the Design Kit release notes. A complete description of the process and Design Kit features for each process variant is documented in the Design Rules document and any associated Design Application Notes. All of these documents are available from the Jazz Ebizz website.</p>			
<p>A Design Kit that supports superset process variants will offer only a few choices for the “techver” kit setting (see section 4.6). Instead of multiple process variants, environment switches are used to control whether a process or Design Kit feature will be supported or flagged as unsupported. The design kit may support multiple process variants, but you must configure the design kit to use the “superset” process variant as the “techver” setting. Then you can choose to enable or disable process and Design Kit features to match the process variant of interest.</p>			
<p>A utility is provided in the Design Kit that allows you to select the process variant to use, and to set the Unix environment variables that control the Design Kit and the DRC verification deck. To open the utility, choose the menu item from the Cadence CIW Jazz Menu titled “Application Notes” and then the sub-menu item titled “Jazz Process Variant Selection and Setup”.</p>			
<p>If you receive a warning stating “This design kit does not have any 'superset' process variants” then the design kit you are using does not support “superset” process variants and you can skip this section of Design Kit configuration.</p>			
<p>The Setup form shows a list of process variants supported by the Design Kit and the Parent Variant that must be used for the “techver” kit setting. It also shows a list of all of the process features that the process variant supports. Each process feature line has a checkbox to enable/disable the feature, a description of the feature, and a list of the cells in the Design Kit that are enabled/disabled by this feature. If a feature is non-editable (grayed out), this indicates the superset parent process variant supports the feature, but the process variant you have selected does not support this feature.</p>			
<p>The Setup form is shown below as an example. In this example, the process variant desired by the user is “ca18hr”. The form shows that this is a subset of the “ca18hd” superset parent process variant. The form also shows that the ca18hr process variant will never allow the “highvt” and “hvres”</p>			

process options, while the “deepnwell”, “topmm”, and “topmm2” process options can be enabled and disabled. All of these process features are available in the ca18hd superset parent variant. Each of the process options shows a description of the layer/feature, and also shows a list of cells in the Design Kit that will be disabled if this process option is disabled. In this example, the form shows that the user chose to disable the “deepnwell” process option, and therefore all of the cells associated with the “deepnwell” option.

Jazz Process Variant Selection and Setup

OK Cancel Defaults Apply Help

Process Variant: ca18hr Parent Variant: ca18hd

☐ deepnwell = Deep Nwell
Cells: ddnw diso nfet3p3_id pfet3p3_id

☒ topmm = MIM on M2
Cells: cmim2 cmimw2 cmimw2_4 csmim4 csmimw4 csmimw4_4

☒ topmm2 = MIM on M3
Cells: cmim2_m3 cmimw2_m3 cmimw2_4_m3 csmim4 csmimw4 csmimw4_4

☐ highvt = High VT F01
Cells: nfet_vth pfet_vth

☐ hvres = High Value Poly Resistor
Cells: rppoly_hi

Once you have selected the process variant and options you will use on the Setup form, click the OK or Apply button. A popup message will appear showing you what environment settings must be made for this set of process options to take effect. The message in the popup is also printed to the Cadence CIW so that you can more easily copy/paste the settings. An example of the popup message is shown below:

Message

To make these settings persistent
Set these Environment Variables before starting your session:

```
setenv RDS_CDS_VERIFY_TECH ca18hd
setenv JAZZ_BAD_LAYER_deepnwell t
setenv JAZZ_BAD_LAYER_highvt t
setenv JAZZ_BAD_LAYER_hvres t
setenv JAZZ_BAD_LAYER_topmm nil
setenv JAZZ_BAD_LAYER_topmm2 nil
```

Close

In order to make these settings permanent, you must include them in your `cdsPrj.cshrc` file (see section 4.9.5) or otherwise set the Unix environment variables before you start the Design Kit.

4.12 Adding users to an existing project

This section describes what to do when you have already created a project and now wish to add more users to the project. There are two different ways to perform this type of project update: add users to an existing project user list, or create a new project user list. The first approach may work better when directory access permissions allow project members to modify each other's data. The second approach may work better when directory access permissions only allow the project members to view each other's data but not modify it.

4.12.1 Update project user list

Add the new users, one per line, to the end of the existing `user_list` file you created in step 4.9.3. You can use these commands below, or you can edit a file with your favorite text editor. The commands below will update the file named `user_list` by adding two new users, `user3` and `user4`:

```
% echo user3 >> user_list
% echo user4 >> user_list
```

Use the `makeProjectTree` script again using the same options you used previously to update the files and directory structure for the existing project and create the new user directories. You will need to use the same options you used from section 4.9.4 when you first ran the script.

This command will also create the two new work areas in the current project for the new users added to the `user_list` file. It will not delete any existing user's directory or data. It will re-attach the design library for ALL users of the project listed in the `user_list` file if your Unix access permissions allow this change. This command must appear all on one line, even though it may be printed here on multiple lines:

```
% makeProjectTree [project_name] -users user_list -cds -tech
[techfile] -tech_ver [techver] -ver
```

4.12.2 Create another project user list

Create one or more different files that contain only users to be added to the project. This is done in the same way you created the `user_list` file in step 4.9.3. For example, you could create a separate file for each individual user, or each set of users within the same Unix permissions group. You can use these commands below, or you can edit a file with your favorite text editor. The commands below will create the file named `new_user_list` with two new users, `user3` and `user4`:

```
% echo user3 > new_user_list
% echo user4 >> new_user_list
```

The commands below will create two files, `user3_file` and `user4_file` each one containing only the single user name `user3` or `user4`, respectively:

```
% echo user3 > user3_file
% echo user4 > user4_file
```

Use the `makeProjectTree` script again to update the files and directory structure for the existing project. This time you will use each of the newly created user list files instead of the original `user_list` file. You will need to use the same options you used from section 4.9.4 when you first ran the script. Run the `makeProjectTree` script multiple times, each time using a different `new_user_list` file you created.

This command will create only the new work areas in the current project for the new users added to the `new_user_list` file. It will not delete any existing user's directory or data. It will re-attach the

design library for all users listed in the `new_user_list` file if the permissions allow this change. This command must appear all on one line, even though it may be printed here on multiple lines:

```
% makeProjectTree [project_name] -users new_user_list -cds -tech  
[techfile] -tech_ver [techver] -ver
```

TowerJazz Confidential
Downloaded by: Sanjay Raman
Date: 08/14/2012 08:14
IP: 128.173.89.96

5 Troubleshooting

For first time project directory or Design Kit setup, make your project directory using the utility `makeProjectTree`. If there are warning messaged when you try to make a project tree, try logging out and back in to your Unix terminal session, and then repeating the process for making a project tree and launching Cadence as described in section 4.9 and 4.10.

5.1 Design Kit Expiration

Each Jazz Design Kit is compiled with an expiration date based on when the kit is built. The expiration is 12 months (12 * 30 days) for kits built after August 2005, and 9 months (9 * 30 days) for kits built earlier. For the last 30 days prior to expiration, a warning is printed in the CIW and log files that the kit will expire soon. Once a design kit expires, the device parameters in the schematic and layout will no longer be editable. You can still modify the schematic or layout (other than changing device parameters), run simulations, perform DRC, LVS, or Extraction runs, and stream out data when the design kit has expired. You will not be able to modify device parameters or place new devices.

If a newer Design Kit is available, you should evaluate migrating to the newer kit. If no newer kit is available, or it is not feasible to migrate to a newer Design Kit, a Design Kit Expiration (Design Kit License) patch can be applied to your existing kit to extend the Design Kit expiration date. The Design Kit Expiration patch is generally available on the Jazz Ebizz website, or it can be requested from your Jazz support representative. More information on how to migrate between Design Kit releases can be found in section 6.

The Jazz Design Kit prints this message in the Cadence CIW stating it will expire soon:

```
"Design Kit license will expire in X days. Please contact your Jazz
Customer Solutions representative"
```

The Jazz Design Kit prints this message in the CIW once it has expired:

```
"Design Kit license has expired. Please contact your Jazz Customer
Solutions representative"
```

Other things to try

Start Cadence after executing Jazz setup:

```
% source ~/JAZZ.cshrc
% cdsprj [project_name]
% icfb
```

Rename the `.cdsenv` file in your home directory:

```
% mv ~/.cdsenv ~/.cdsenv_last
```

Rename the `.cdsinit` file in your home directory:

```
% mv ~/.cdsinit ~/.cdsinit_last
```

5.2 Wrapper scripts

If you have your own wrapper scripts for `icfb` or other EDA tools, check to be sure that your scripts do not remove the Jazz design kit paths from either the `$PATH` or `$LD_LIBRARY_PATH` environment variables. Here are some other points that should be considered if you use wrapper scripts:

- The wrapper script must use the `-f` “fast” startup option if it is a `csh` or `tcsh` script. This informs the shell that it should not read the `.cshrc` file and reset environment variables like `PATH`.

For example, use this in the script header:

```
#!/bin/csh -f
```

Do not use this in the script header, as it will reset the `PATH` environment variable

```
#!/bin/csh
```

- The wrapper must not reset the path, it must prepend or append to it.

For example, this should be avoided:

```
set path=( /cadence/tools/bin /bin )
```

Instead, use a command like this:

```
set path = ( $path /cadence/tools/bin )
```

Or like this:

```
set path = ( /cadence/tools/bin $path )
```

TowerJazz Confidential
Downloaded by: Sanjay Raman
Date: 08/14/2012 08:14
IP: 128.173.89.96

6 Updating an existing kit and/or project directory

This section describes the steps that should be taken when you have already created a project directory in a previous kit install, and now want to install and use a new version or an additional kit. There are two approaches to performing this upgrade:

Kit upgrade - same process technology as before, different kit version.

Use this option if you are changing the version of the design kit used and the process technology used remains the same. Many of the Jazz design kits can support more than one process variant (the `[techver]` setting in section 4.6). You should use this option if you intend to add a project that uses the same design kit but a different process variant.

Additional kit - new process technology.

Use this option if you are adding or changing the process technology and the kit used (the `[techname]` from section 4.5). For example, if you have the sbc35x Jazz design kit installed and you are now adding the sbc18d Jazz design kit.

You do not need to rerun `makeProjectTree` if you are upgrading an existing kit or switching between different versions of a design kit. You must run `makeProjectTree` and build a new project tree when you are changing processes and have a new technology library.

6.1 Kit upgrade — same process technology as previous kit

You can maintain multiple Jazz design kits on disk. Existing designs that have already taped out should always be archived with a copy of the kit version used for that particular design. After installing a new design kit, or whenever you switch between design kit versions, you must also update any existing designs from within the EDA tools. For example, there may be changes to process related values like sheet resistance or capacitance, or layout cell changes that can affect your design. Please use the instructions that are provided inside the kit, in the FAQ entitled "What steps should I perform when I switch between kit releases?" Section 3 of this installation guide tells how to locate the FAQ from within the Kit.

6.1.1 Install new design kit

Select a new directory `[kit_dir2]` to hold the new design kit. This must be a different directory than any previous kit installations. Never install a design kit on top of another existing design kit. For example, if you previously used sbc35x for the `[kit_dir]` you could use the name `sbc35x_kit2` or use a name like `sbc35x_20061220` that incorporates the new kit datestamp. The new kit directory should be parallel to the existing kit dir, for example, if you used `/tools/jazz/sbc35x` for the first kit, use the directory `/tools/jazz/sbc35x_20061220` for the second kit

Install the new kit in a new directory, `[kit_dir2]` using the instructions from section 4.2 - 4.4.

6.1.2 Using the new design kit

Two different setups can be used to switch users between design kits:

- In the first setup, users must edit their `JAZZ.cshrc` file to point to the newly installed design kit, or you must create multiple `JAZZ.cshrc` files to use the different design kit directories. This setup should be used if there is a need to use a particular design kit version with a particular project. For example, one project may be nearing completion when a new design kit is delivered, but another

project is just starting. The first project should continue to use the older design kit to prevent impact to the project, while the second project should upgrade to the latest design kit to obtain the latest kit features.

- In the second setup, the CAD administrator modifies the installation so that users are automatically switched to the newly installed design kit. This setup should be used if there is little or no concern about the impact of design kit changes on project schedule. For example, if all projects are still in the early design stages, they should all be updated to the latest design kit. If needed, a specific project or user can be modified using the first technique while everyone else is updated using this second technique.

For the first setup, update the `RDS_ROOT` environment variable in the `JAZZ.cshrc` file to use the new `[kit_dir2]` as described in section 4.8.1. Every copy of this file will need to be updated. Users will need to log out and back in to reset the environment. Then users can use the new design kit as usual.

For the second setup, you will use a symbolic link to point to a particular revision of the kit. In this way, the kit directory referred to in the `JAZZ.cshrc` file does not change, but the actual directory on disk can be changed. Users that need to refer to a specific kit version and do not want to be automatically updated will need to follow the instructions above for the first setup technique. Here are step-by-step instructions for this second technique:

Go to the original `[kit_dir]` you used in step 4.2

```
% cd [kit_dir]
```

Go up one directory:

```
% cd ..
```

Rename this original `[kit_dir]`. For example, use the first kit's downloaded kit name and datestamp as part of the name:

```
% mv [kit_dir] sbc35x_20051021
```

Go to the second `[kit_dir2]` you used in step 6.1.1

```
% cd [kit_dir2]
```

Go up one directory:

```
% cd ..
```

Rename this second kit if needed. For example, use the second kit's downloaded kit name and datestamp as part of the name:

```
% mv [kit_dir2] sbc35x_20061220
```

Make a symbolic link from this new second kit to a `[kit_link]`. For example, use the Jazz `[techname]` like `sbc35x` for the `[kit_link]`.

```
% ln -s sbc35x_20061220 [kit_link]
```

If the `[kit_link]` is not the same as the original `[kit_dir]` that you used when you created the `JAZZ.cshrc` file, you will need to edit the `JAZZ.cshrc` file and update the `RDS_ROOT` environment variable to use this new `[kit_link]`.

Be sure all users log out and back in, then source the environment files as usual:

```
% source ~/JAZZ.cshrc
```

```
% cdsprj [my_project]
```

```
% icfb
```

You can switch back to the first kit if needed by changing the symbolic link `[kit_link]` from the `[kit_dir2]` to point back to your original `[kit_dir]`. Be sure all users log out and back in after you make this change.


```
% rm [kit_link]
% ln -s sbc35x_20030815 [kit_link]
```

6.2 Additional kit — new process technology

These instructions apply when you are adding a different technology library. For example, you are adding the sbc18d (SiGe90) library, but already have the sbc35x (SiGe60) kit installed. Select a new directory `[kit_dir2]` to hold the new design kit. This must be a different directory than any previous kit installations. Never install a design kit on top of another existing design kit. For example, you used the directory name `sbc35x_20061220` for the SiGe60 `[kit_dir]` so you could use the directory name `sbc18d_20050708` for the SiGe90 `[kit_dir]`.

Install the new kit in a new directory, `[kit_dir2]` using the instructions from section 4.2 - 4.8 and just sections 6.1.1 and 6.1.2 of the Kit Installation Procedure above. Use a different name for the `JAZZ_kit2.cshrc` file so that the `[kit_dir]` and `[kit_dir2]` can be set independently. You will also need to remember the setup information for the new kit that is shown in the table in section 4.1. You can use the same project root directory with the different kits, but the project names must be different.

Make a new project tree using commands similar to the ones you used for the first kit install in section 4.9.4. Use the information for kit 2 wherever needed:

```
% source JAZZ_kit2.cshrc
% cd $PROJ_ROOT
% vi user_list
% makeProjectTree [project_name_kit2] -users user_list -cds -tech
[techfile_kit2] -tech_ver [techver_kit2] -ver
```

Then source the remaining environment files and run Cadence:

```
% source JAZZ_kit2.cshrc
% cdsprj [project_name_kit2]
% icfb
```

Each time you wish to work with kit 2, you will need to use the new setup file. If you wish to use kit 1 again, you will need to open a new terminal window and source the original setup files.

6.3 Additional options for the CAD Administrator

It is possible to control some aspects of the design kit version used from within the project directory itself. These options are discussed below as a way for a CAD administrator to switch specific users or projects on a limited basis between kit versions. However, the safest way to perform this switch without creating a potential for version-to-version compatibility issues is to use one of the two methods previously described in section 6.1. More details on the environment variables described below can be found in the document “Jazz Cadence AMS Design System Overview”, NPB-PS-0404. This document can be obtained from the Jazz Ebizz website.

There are two Jazz design kit environment variables that can be used to override the design kit base directory (the environment variable `RDS_ROOT`) on either a user-specific or project-specific basis. In each case, the setting for the design kit base directory (`RDS_ROOT`) used in the `JAZZ.cshrc` file will be overridden.

In the first method, the design kit base directory is reset on a project-specific basis. Edit the file `[project]/cds_master/cdsPrj.cshrc` and set the environment variable `PROJ_RDS_ROOT` to a kit install directory. This will set to the design kit base directory for all users of this `[project]`.

In the second method, each project user resets the design kit base directory. Edit the file `cdsUshr.cshrc` located in the project-specific work directory for a particular user of the project and set the environment variable `USER_RDS_ROOT` to a kit install directory. This will reset the design kit base directory for this one user of this specific project without affecting other users and without affecting this user's other projects.

TowerJazz Confidential
Downloaded by: Sanjay Raman
Date: 08/14/2012 08:14
IP: 128.173.89.96

7 Advanced kit installation options for the CAD Administrator

This section describes an alternative setup configuration from the standard Jazz setup scripts described in section 4.8 and the standard Jazz project directory setup described in section 4.9. It is strongly recommended that you use the standard setup configuration the first time you install and use the Jazz design kits. Once you have verified that the standard Jazz setup configuration works, you could switch to using this alternative setup.

This section includes an alternative setup that is a non-standard configuration meant for use by customers who have their own project directory structure and setup, and are also very familiar with the requirements of all EDA tools in use. The setup described here consists of a revised C-shell script file, a Cadence `cds.lib` file, and a Cadence `.cdsinit` file. Using this setup, you can integrate the Jazz design kit into any project or directory structure and use any Unix shell for your command line.

This section also information on how to alter where the Jazz Calibre integration stores its data.

7.1 Advanced Kit install shell script

The following C-shell (`csh`) script shows the environment variables that must be set for the Jazz kit to function. You can rewrite this script so that it is compatible with your preferred Unix shell (e.g., Born or Korn shell). We assume you will use the `source` command to execute this script, but it can also be used as a part of a larger wrapper script to perform the needed Jazz kit setup and then start a specific EDA vendor tool.

The information shown in the script is taken from the table shown in section 4.1. You will need to edit the script to set the `[kit_dir]`, the platform (Sun or Linux), the Jazz technology library `[techfile]`, and the Jazz process variant `[techver]` being used.

```
##### Environment variables
# Set RDS_ROOT to the top of the Jazz Kit installation
setenv RDS_ROOT [kit_dir]/HOTCODE

# Set the process and design kit selected
setenv RDS_CDS_TECH [techfile]
setenv RDS_CDS_VERIFY_TECH [techver]

# Set the appropriate platform
# SunOS = sun5
# Linux = lnx86
setenv RDS_PT sun5
# setenv RDS_PT lnx86

# Set location of the perl executable
setenv PERL5 [full path to perl executable]

##### Do not change any lines below here #####
# Set variables needed for different subdirectories in kit.
# These are derived directly from RDS_ROOT and
# do not need to be changed.
setenv RDS_CDSWARE $RDS_ROOT/amslibs/cds_default/cdsware
setenv RDS_CDSLBS $RDS_ROOT/amslibs/cds_default/cdslbs
```

```
setenv RDS_ETC $RDS_ROOT/amslibs/cds_default/etc
setenv RDS_CDS_INIT_FILES $RDS_ETC
setenv RDS_CDS_LIB_VERS $RDS_CDS_INIT_FILES/system.lib
setenv RDS_TECH $RDS_ROOT/techs
```

```
# Set path
set path=( $RDS_ROOT/bin/rfbin $RDS_CDSWARE/bin $path)
```

7.2 Advanced Kit install cds.lib contents

The following Cadence `cds.lib` file shows the minimum library definitions that must be present for the Jazz kit to function. You must create the `softlibs` Cadence library if it does not exist, and you must attach it to the appropriate Cadence Technology Library. The `softlibs` library may be populated with intermediate copies of many Cadence Virtuoso Pcells used in the Jazz design kit. The library can be shared among users if access permissions allow, or it can be unique to each user. The `work_lib` is optional and is used in the standard Jazz project directory structure to contain the user's design library. You can use any libraries you like for the user work library.

```
DEFINE softlibs ./softlibs
DEFINE work_lib ./work_lib
INCLUDE $RDS_CDS_LIB_VERS
```

7.3 Advanced Kit install .cdsinit file contents

The following Cadence `.cdsinit` file shows the minimum SKILL commands that must be executed for the Jazz kit to function. Cadence has several methods for loading `.cdsinit` files. You will need to ensure that each user will load these required commands when starting the Cadence framework.

```
if( InitPath = getShellEnvVar( "RDS_CDS_INIT_FILES" ) then
    load(strcat(InitPath "/system.cdsinit"))
else
    printf("ERROR: system variable 'RDS_CDS_INIT_FILES' is not set.\n")
)
```

7.4 Settings required for the Jazz Calibre integration

In order to use the Jazz Calibre integration, you must also set the following environment variables. These settings are not needed for the Cadence Assura tool or when using the Mentor Graphics Calibre Interactive integration. These variables are automatically set if you use the standard Jazz project directory setup from section 4.9. If you use the advanced kit installation script from section 7.1, you must add these variables to the script you create ONLY if you use the Jazz Calibre integration. If you are using the standard Jazz project directory structure, these variables can be reset on a per-project or per-user basis.

7.4.1 What variables need to be set?

You need to set/reset these four environment variables. Each one is set to a directory where the generated data files will be stored.

```
GDS_DIR
VER_DIR
CBR_DIR
PEX_DIR
```

The default values for these variables are derived from other existing project variables. The values are shown below:

```
GDS_DIR  ${PROJ_ROOT}/${PROJ_ID}/gds_dir
VER_DIR  ${PROJ_ROOT}/${PROJ_ID}/verification
CBR_DIR  ${VER_DIR}/cbr_dir
PEX_DIR  ${VER_DIR}/pex
```

7.4.2 What directories must exist?

You must create the directory structure that is automatically created by the `makeProjectTree` script. Here is the list of all of the directories that must exist for the Jazz Calibre integration to function:

```
$GDS_DIR
$VER_DIR
$CBR_DIR
$PEX_DIR
$VER_DIR/ant
$VER_DIR/drc
$VER_DIR/drclvs
$VER_DIR/esd_lup
$VER_DIR/lup
$VER_DIR/lvs
$VER_DIR/softerc
$VER_DIR/street
$VER_DIR/stress
```

TowerJazz Confidential
Downloaded by: Sanjay Raman
Date: 08/14/2012 08:14
IP: 128.173.89.96

7.4.3 How to change the variables?

In the standard Jazz project directory structure, these variables are set in the project `cschrc` file located here:

```
${PROJ_ROOT}/${PROJ_ID}/cds_master/cdsPrj.cschrc
```

You can add the commands to set each of the variables by editing this file. For example, to set the directories to the user's home directory, you can use these lines. You must ensure these directories exist prior to using the Jazz Calibre integration.

```
setenv GDS_DIR ~/gds_dir
setenv VER_DIR ~/verification
setenv CBR_DIR ${VER_DIR}/cbr_dir
setenv PEX_DIR ${VER_DIR}/pex
```

To set the directories to each user's project directory, you can use this example. Again, you must ensure these directories exist prior to using the Jazz Calibre integration.

```
setenv GDS_DIR ${PROJ_ROOT}/${PROJ_ID}/work_libs/`whoami`/cds/gds_dir
setenv VER_DIR
${PROJ_ROOT}/${PROJ_ID}/work_libs/`whoami`/cds/verification
setenv CBR_DIR ${VER_DIR}/cbr_dir
setenv PEX_DIR ${VER_DIR}/pex
```