Cadence setup manual v1.0

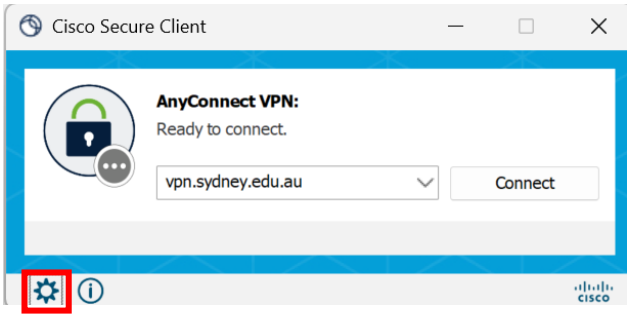
WSL installation

Follow Microsofts WSL Install Guide

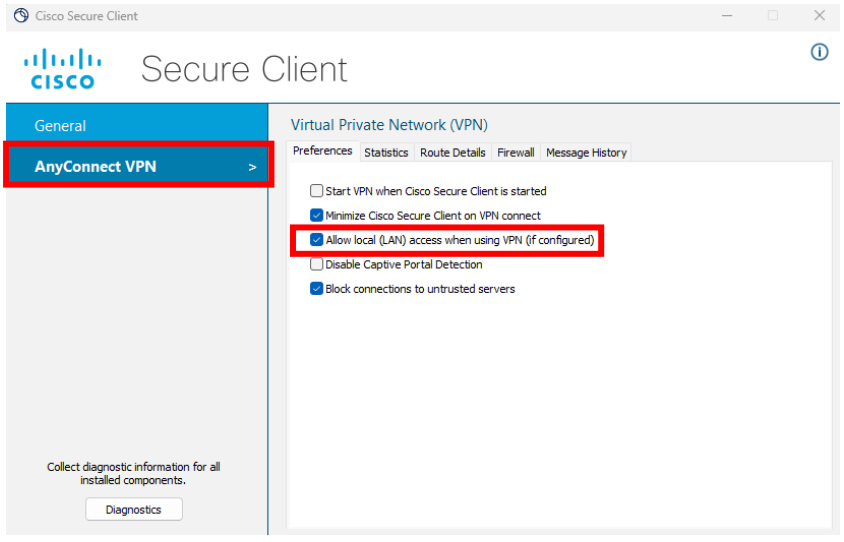
Cisco Secure Client installation

Install Cisco client. Uni credentials will be required.

Once installed, local access isn't enabled by default which means that no virtual machine connected to the LAN through wsl has access to the VPN. To modify it, execute Cisco Secure Client and open settings



Enable “Allow local (LAN) access when using VPN (if configured)



Now your CentOS distribution will have access to the VPN.

CentOs installation

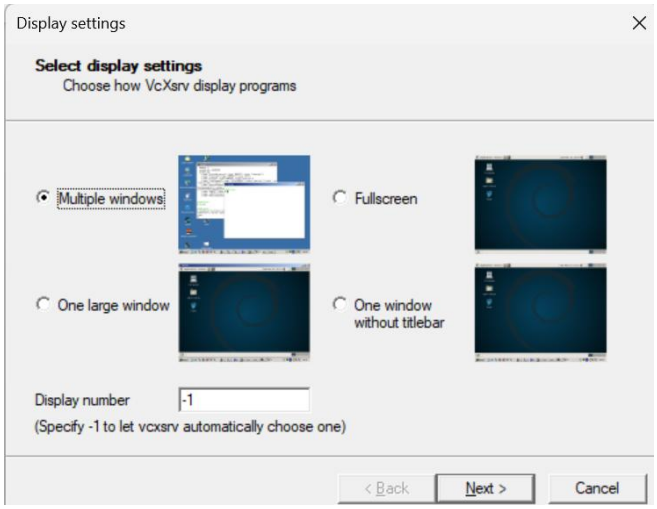
Download CentOS v7 from CentOS-WSL because it’s built with the same kernel as RHEL 8.

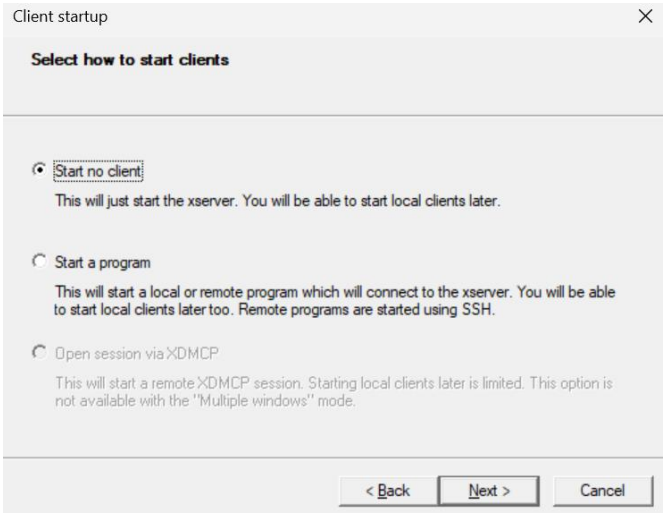
Important, WSL has to be running WSL 2. Execute CentOS.exe. Now CentOS 7 terminal is available in windows.

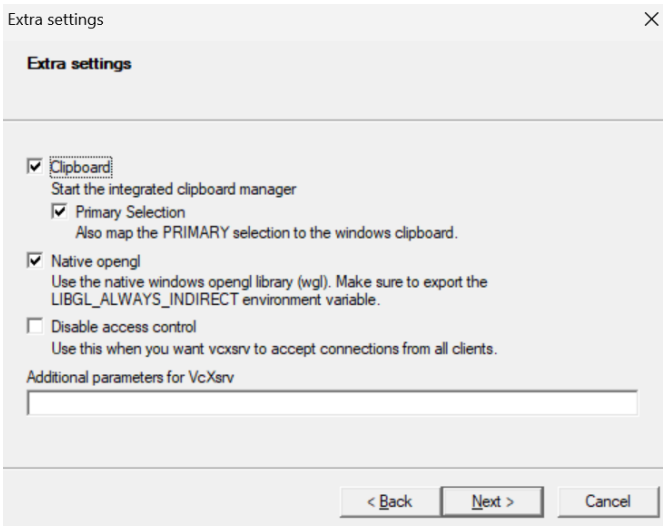
Vcxsrv installation

Install vcXsrv which is an open-source X server for windows. This server allows users to run graphical application from a linux environment on their windows system.

Run “xlaunch” to modify the display settings.







CentOs setup

Run windows PowerShell. On the tab option look for CentOs and open a new terminal.

From first of July 2024 on CentOS 7, it's needed to swith to Vault archive repositories:

```
vi /etc/yum.repos.d/CentOS-Base.repo
```

Replace the file with the following lines:

```
[base]
name=CentOS-$releasever - Base
baseurl=http://vault.centos.org/7.9.2009/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

[updates]
name=CentOS-$releasever - Updates
baseurl=http://vault.centos.org/7.9.2009/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

[extras]
name=CentOS-$releasever - Extras
baseurl=http://vault.centos.org/7.9.2009/extras/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7

[centosplus]
name=CentOS-$releasever - Plus
baseurl=http://vault.centos.org/7.9.2009/centosplus/$basearch/
gpgcheck=1
```

```
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

Now you can run

```
sudo yum update
```

To create a user, follow the next step:

```
sudo adduser user1
```

Next, you'll need to give a password with:

```
sudo passwd user1
```

To give sudo access, modify the sudoers files with:

```
sudo vi /etc/sudoers
```

Add your user:

```
user1 ALL=(ALL) ALL
```

Now you'll have sudo access.

Modify *.bash_profile* to include the correct bin paths and have it as a permanent change:

```
# .bash_profile

# User specific environment and startup programs
PATH=$PATH:$HOME/.local/bin:$HOME/bin

export PATH
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi
```

Now, Cadence will require a license. So, for SoC team permission was granted to use uni license. To add the license variable permanent to your user, modify the source file of your interpreter. CentOS use bash as default, so modify *.bashsrc* with:

```
vi ~/.bashsrc
```

Add the following line to the file:

```
export CDS_LIC_FILE=5280@cadence.lic.sydney.edu.au
```

Additionally, to use VcXsrv, windows x server, add to *.bashrc* the following line:

```
export DISPLAY=$(ip route|awk '/^default/{print $3}'):0.0
```

Now your display will be routed to the local ip address and access to the x server for a GUI.

After adding the lines to the shell file, reload settings with:

```
source ~/.bashrc
```

InstallScape installation

Open your favorite browser and go to Software Downloads. Scroll until you reach “Download and Install with InstallScape”, then click on the link of that section, cadence will ask for your username and password. Then download the tar installscape file.

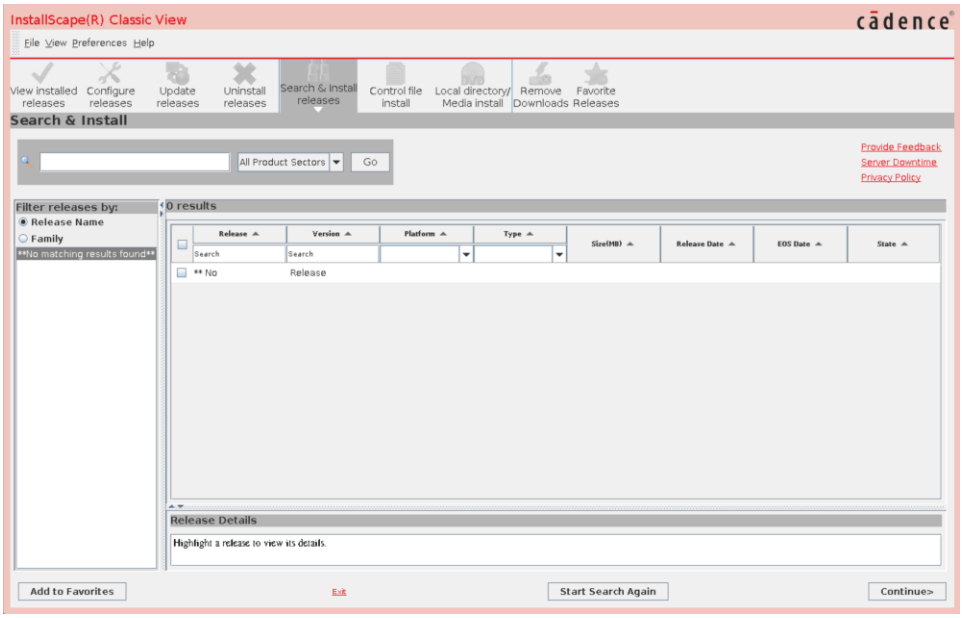
On File Explorer locate the installscape file and move it inside your wsl distro. Common location for wsl will be: *\\wsl.localhost\CentOS7\home\user1\downloads*.

Open a CentOS terminal and untar the file. Go to InstallScape bash file.

```
cd ./iscape-run/iscape/bin
```

BEFORE executing “iscape.sh” file, make sure that VcXsrv is running.

```
sudo bash ./iscape.sh
```



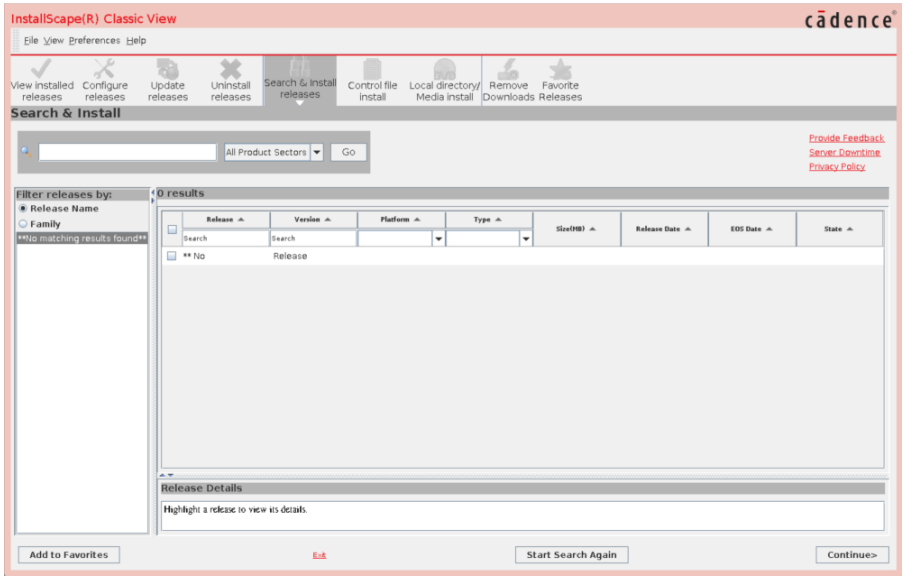
Cadence virtuoso installation

On the terminal install the following packages by running the command:

```
sudo yum update
```

```
sudo yum install -y   libX11 libXext libXtst libXt libXft libXrender libXrandr mesa-libGLU libXmu openmotif libpng xterm gawk ksh tcsh libjpeg-turbo   libxcb qt5-qtbase-gui qt5-qtx11extras java-1.8.0-openjdk-devel epel-release xorg-x11-server-Xvfb
```

Make sure that VcXsrv is running and execute iscape.sh. On InstallScape go to “Search & Install releases”



Your cadence credentials will be requested. Search for “IC23.1” and download & install it. It's recommended to download and install it in */opt/cadence/*

Once the process is completed, don't worry if the installation shows an error. Modify your shell file, by default *.bashrc,* and add the location of virtuoso bin folder to your path with the following line:

```
export CDSHOME=/opt/cadence/installs/IC231
[ -d "$CDSHOME/tools/bin" ]           && export PATH="$CDSHOME/tools/bin:$PATH"
[ -d "$CDSHOME/tools/dfII/bin" ]       && export PATH="$CDSHOME/tools/dfII/bin:$PATH"
```

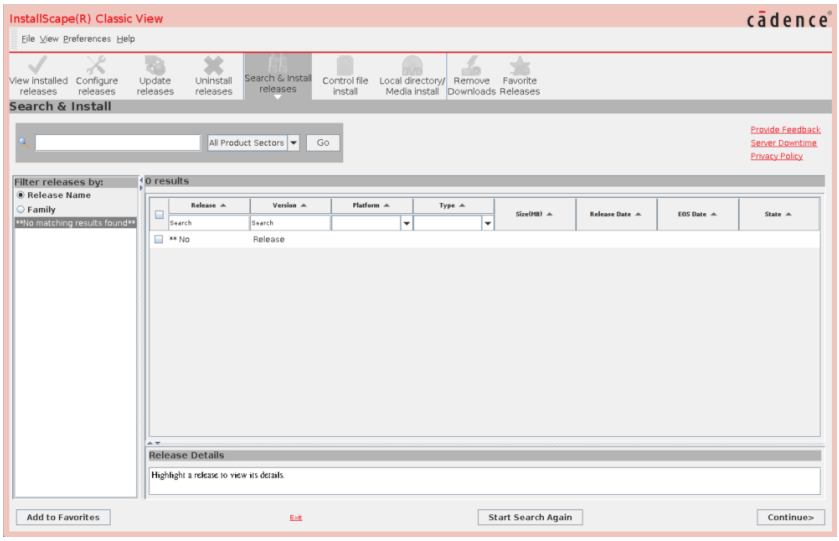
Connect to usyd vpn with cisco client and make sure VcXsrv is running to execute Virtuoso.

```
virtuoso &
```

Cadence Spectre installation

Spectre is for analog simulation.

Make sure that VcXsrv is running and execute iscape.sh. On InstallScape go to “Search & Install releases”



Your cadence credentials will be requested. Search for “SPECTRE 23.1” and download & install it. It's recommended to download and install it in */opt/cadence/*

Once the process is completed, don't worry if the installation shows an error. Modify your shell file, by default *.bashrc,* and add the location of virtuoso bin folder to your path with the following lines:

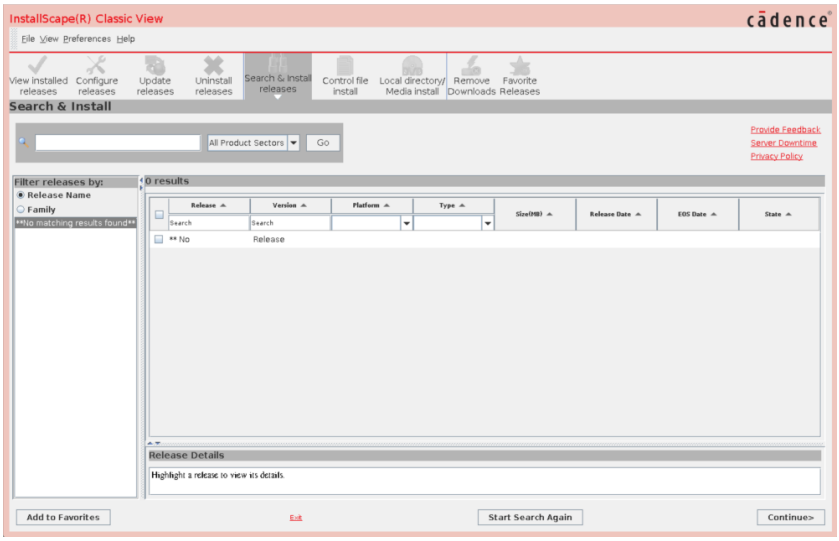
```
export SPECTREHOME=/opt/cadence/installs/SPECTRE231
export MMSIMHOME=/opt/cadence/installs/SPECTRE231
[ -d "$MMSIMHOME/tools.lnx86/spectre/bin" ] && export PATH="$MMSIMHOME/tools.lnx86/spectre/bin:$PATH"
[ -d "$MMSIMHOME/tools/bin" ]           && export PATH="$MMSIMHOME/tools/bin:$PATH"
```

Now you can simulate analog cells on Virtuoso

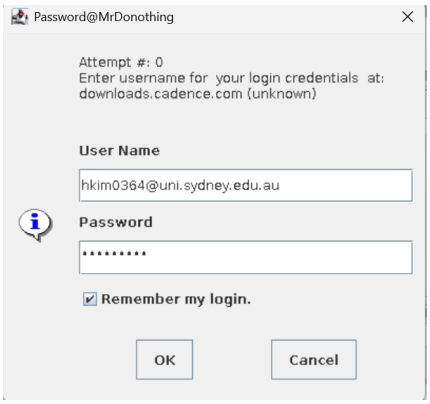
Xcelium installation

Xcelium is for Verilog/SystemVerilog

Make sure that VcXsrv is running and execute iscape.sh. On InstallScape go to “Search & Install releases”



Your cadence credentials will be requested:



Search for “XCELIUM 23.09” and download & install it. It's recommended to download and install it in */opt/cadence/*

Once the process is completed, don't worry if the installation shows an error. Modify your shell file, by default *.bashrc,* and add the location of virtuoso bin folder to your path with the following line:

```
export PATH=/opt/cadence/installs/XCELIUM2309/tools/bin:$PATH
```

Now you can simulate Verilog/SystemVerilog on Virtuoso

SkyWater 130 PDK

The Sky130 Process Design Kits provide for use with Cadence Design Tools and Flows of Virtuoso and Innovus products. You can download the pdk from: https://support.cadence.com/

Before downloading the pdk you have to login to your Cadence account and from the bar menu click on **Resources->PDKs.**

Two different PDKs are provided by Cadence GPDK045 and SKY130. For SKY130, download *Sky130 PDK* and *Sky130 Standard Cell Library*. Then extract *Sky130 PDK* into /home/user1/cadence/lib/ directory, and *Sky130 Standard Cell Library* into /home/user1/cadence/lib/ sky130_release_0.1.0/libs/.

This PDK includes a *cds.lib* file which we will modify:

```
#DEFINE    analogLib   $CDSHOME/tools/dfII/etc/cdslib/artist/analogLib
#DEFINE    basic       $CDSHOME/tools/dfII/etc/cdslib/basic

DEFINE sky130_fd_pr_main ./libs/sky130_fd_pr_main

# SCL 9T
DEFINE sky130_scl_9T     ./libs/sky130_scl_9T_0.1.1/sky130_scl_9T/oa/sky130_scl_9T
DEFINE sky130_scl_9T_HS  ./libs/sky130_scl_9T_0.1.1/sky130_scl_9T_HS/oa/sky130_scl_9T_HS
DEFINE sky130_scl_9T_LP  ./libs/sky130_scl_9T_0.1.1/sky130_scl_9T_LP/oa/sky130_scl_9T_LP
```

Notice that the standard cells (scl) are inside sky130 lib directory. For future projects using this PDK, it must be included in the *cds.lib* file of the project. To include it don't copy and paste the *cds.lib* shown above into the project *cds.lib*, instead include it by adding the following line in the *cds.lib* file of the project.

```
INCLUDE /home/user1/cadence/lib/sky130_release_0.1.0/cds.lib
```

CDS.LIB convention

To facilitate the installation and use of cadence, add to your *.bashrc* each line mentioned in the corresponding installation section. A summary of all required entries (assuming every step was followed) is shown below:

```
#export DISPLAY=$(ip route|awk '/^default/{print $3}'):0.0

export CDS_LIC_FILE=5280@cadence.lic.sydney.edu.au
export CDSHOME=/opt/cadence/installs/IC231
export SPECTREHOME=/opt/cadence/installs/SPECTRE231
export CDS_WS=/home/user1/cadence
export MMSIMHOME=/opt/cadence/installs/SPECTRE231
export PATH=/opt/cadence/installs/XCELIUM2309/tools/bin:$PATH

[ -d "$MMSIMHOME/tools.lnx86/spectre/bin" ] && export PATH="$MMSIMHOME/tools.lnx86/spectre/bin:$PATH"
[ -d "$MMSIMHOME/tools/bin" ]           && export PATH="$MMSIMHOME/tools/bin:$PATH"
[ -d "$CDSHOME/tools/bin" ]             && export PATH="$CDSHOME/tools/bin:$PATH"
[ -d "$CDSHOME/tools/dfII/bin" ]        && export PATH="$CDSHOME/tools/dfII/bin:$PATH"
```

The default cds.lib file located in cadence directory, /home/user1/cadence/cds.lib has been modified to include examples and basic libraries like analogLib.

```
DEFINE analogLib          $CDSHOME/tools/dfII/etc/cdslib/artist/analogLib
DEFINE basic              $CDSHOME/tools/dfII/etc/cdslib/basic
DEFINE sheets             $CDSHOME/tools/dfII/etc/cdslib/sheets
DEFINE aExamples          $CDSHOME/tools/dfII/samples/artist/aExamples
DEFINE ahdlLib            $CDSHOME/tools/dfII/samples/artist/ahdlLib
DEFINE functional         $CDSHOME/tools/dfII/etc/cdslib/artist/functional
DEFINE rfExamples         $CDSHOME/tools/dfII/samples/artist/rfExamples
DEFINE rfLib              $CDSHOME/tools/dfII/samples/artist/rfLib
DEFINE rfTlineLib         $CDSHOME/tools/dfII/samples/artist/rfTlineLib
DEFINE training           $CDSHOME/tools/dfII/samples/artist/training
DEFINE solutions          $CDSHOME/tools/dfII/samples/artist/solutions
```

An example of a *cds.lib* for a particular project is shown below:

```
INCLUDE /home/user1/cadence/cds.lib
DEFINE SoC_probe /home/user1/cadence/SoC/SoC_probe

INCLUDE /home/user1/cadence/lib/sky130_release_0.1.0/cds.lib
```

Verilog AMS Test Bench

This section describes the process of creating a Verilog Ams block and simulates it with a test bench.

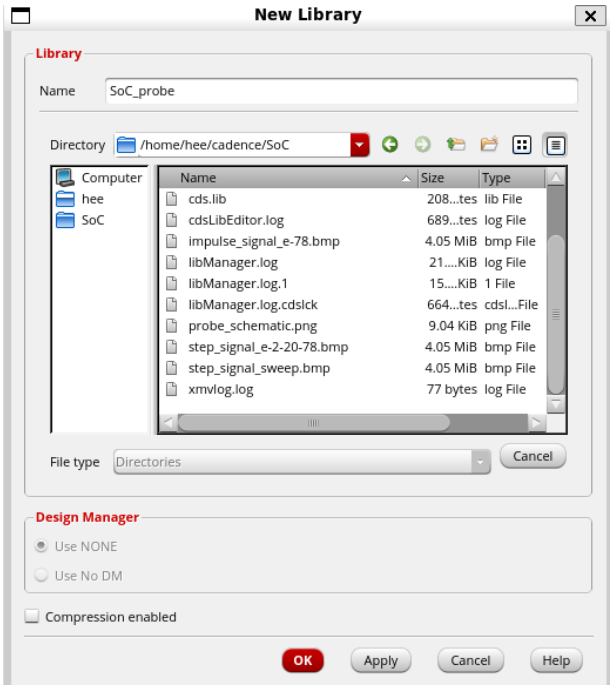
Requirements

- IC23.1
- XCelium23.01
- Spectre23.1

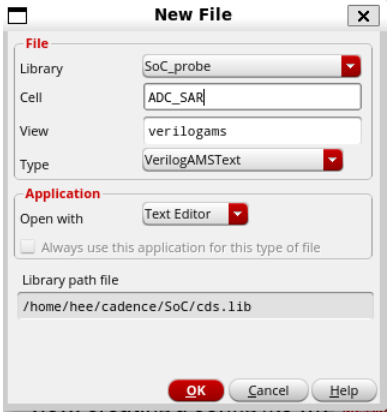
Process

The flow of testing a Verilog Ams block starts from creating a cell with a VerilogAmsText view, creating a config file with the va design attached, and creating a maestro view to run the simulations with ADE Explorer.

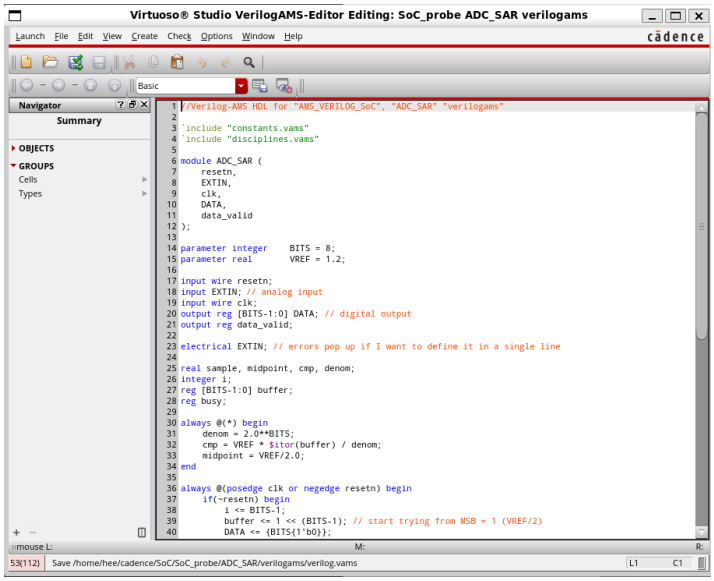
First step will be to create a new library. From the library manager: **File->New->Library.** In the new window choose a name and click “ok”.



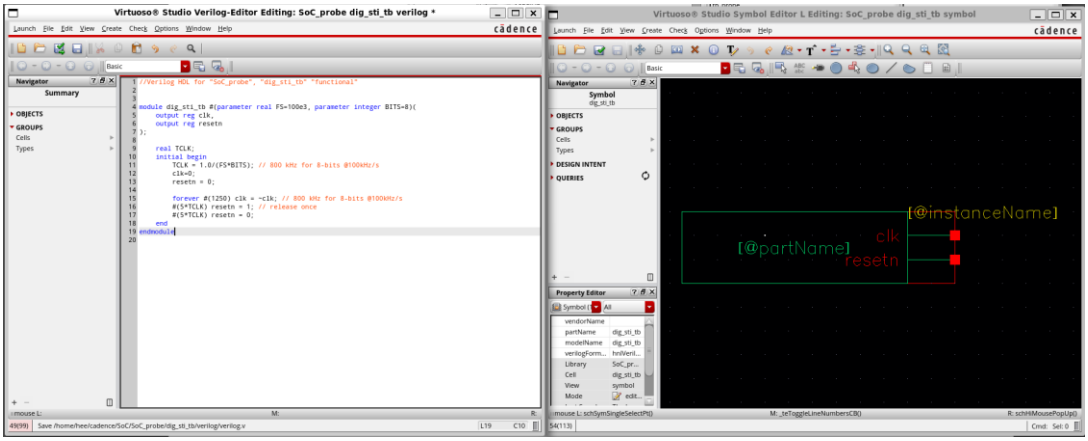
Now to create the Verilog Ams block we will click in **File->New->Cell View.** In the new window we will choose the library we just created, name our new cell and choose the correct view. For a Verilog Ams it is *VerilogAMSText*, choosing *Verilog* or *VerilogA* will create a digital or analog cell, respectively. Finally, click ok.



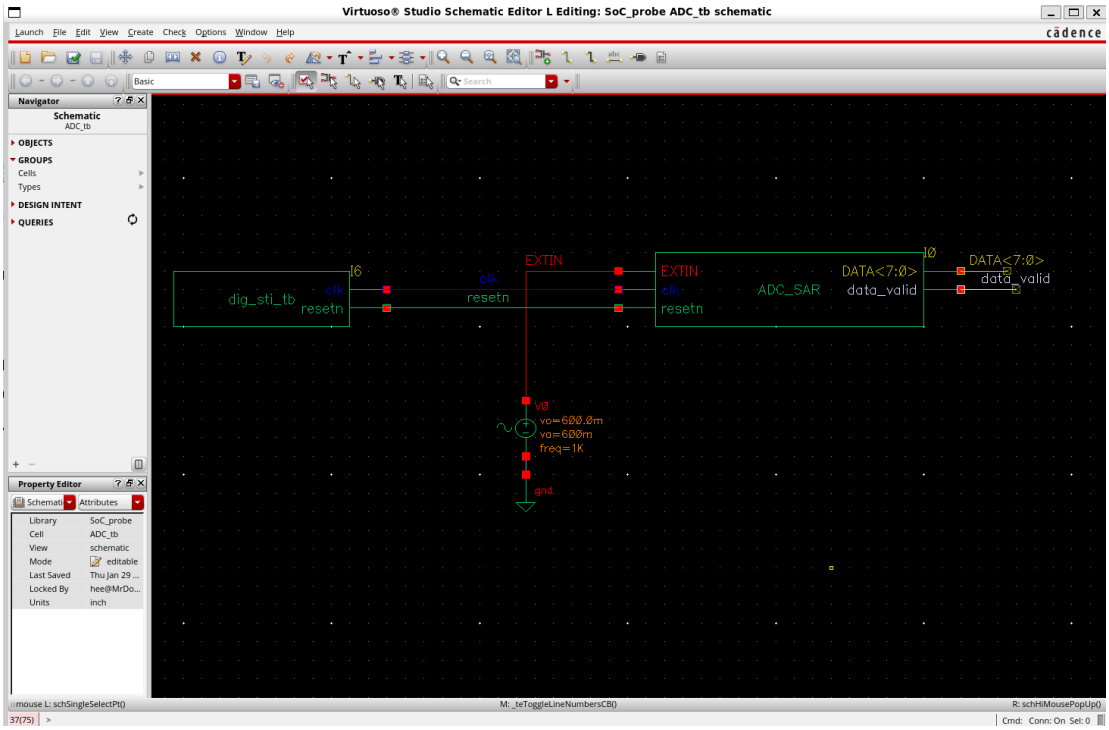
In the new cell view, we will write the behaviour of the block. Once we are done, click extract to save, check for no errors and create a symbol view for future schematics.



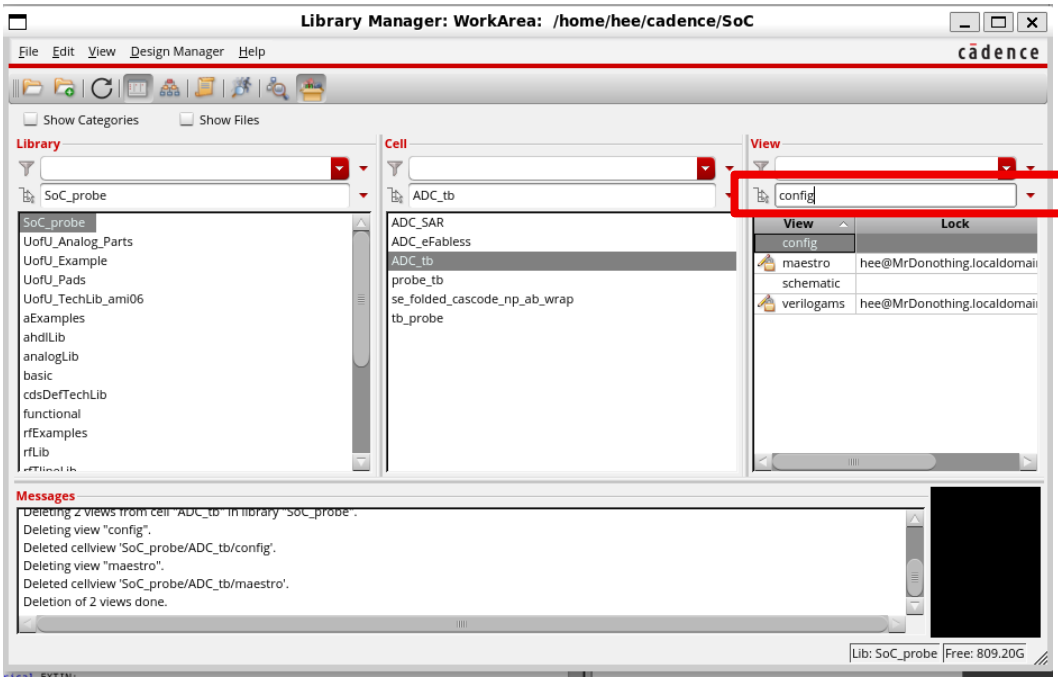
Once the block is finished, we proceed to create the digital block for the test bench. This block includes the clk and reset signals for the ADC. The steps for creating the digital blkc are similar as creating a Verilog Ams block. In the library manager we click **File->New->Cell View** and create a new cell with a Verilog View. Once the cell is done, click *build a database from instance* to create a symbol view.



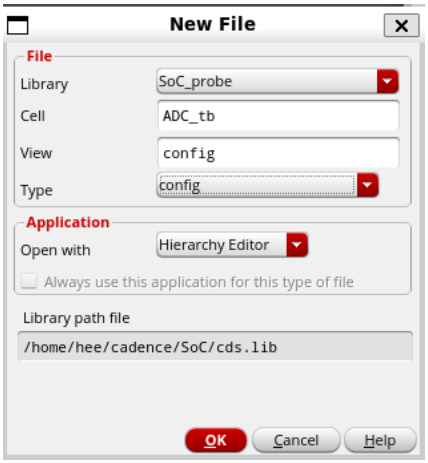
To test the ADC behaviour, we'll proceed to create a test bench. The first step is to create a new cell with a schematic view. In the schematic we will insert the ADC_SAR and the digital stimuli block. Additionally, as an input we will use a *Vsin* to add a 1KHz sine wave as an input.



To simulate the test bench in the Library Manager, we will create a new *config* view attaching the test bench. To create the *config* view, write "config" in the search and select option of the cell view.



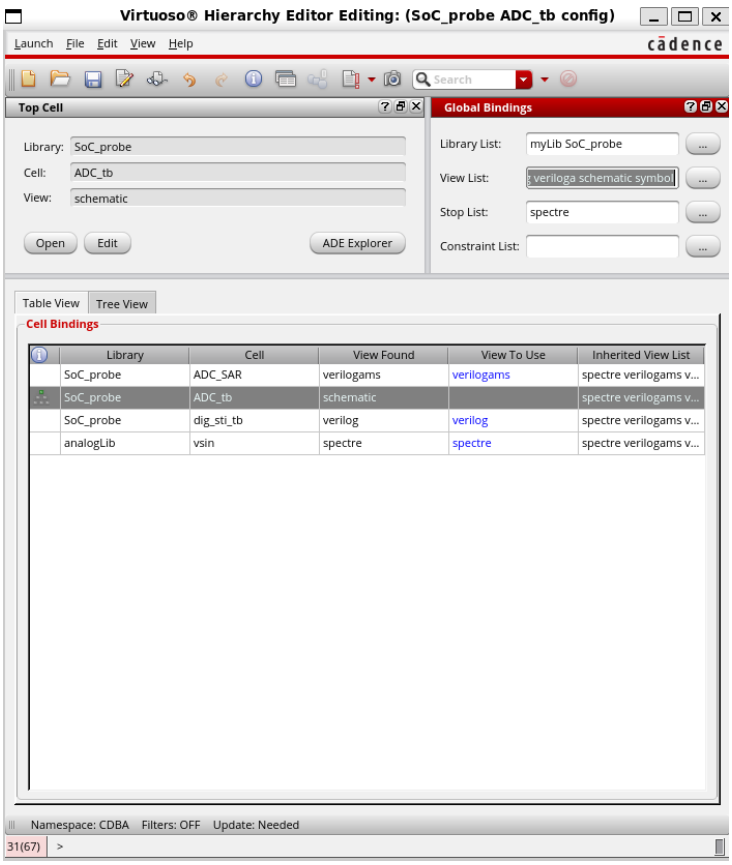
In the new window, choose config and click *ok*.



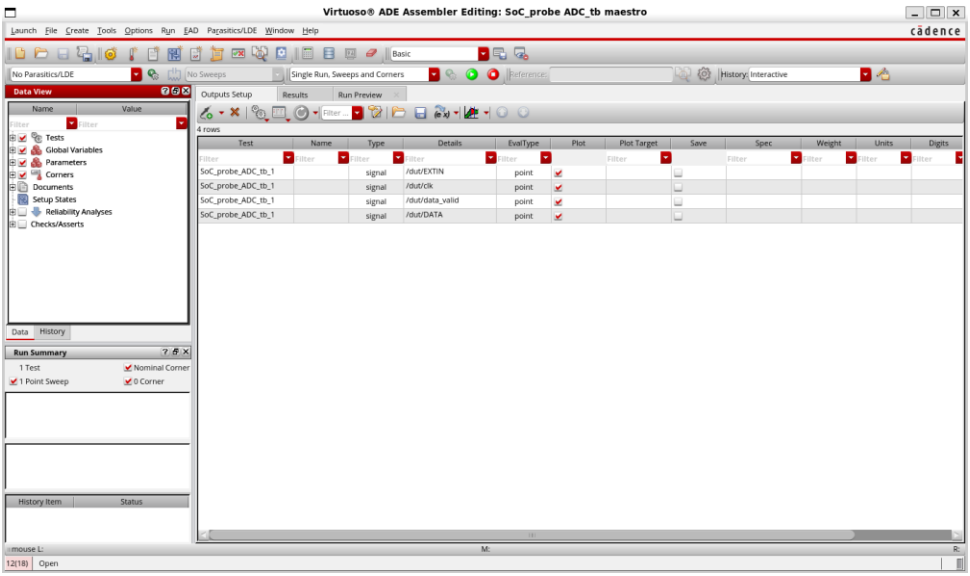
In the new window choose the test bench cell with the verilogams view. To correctly simulate with ADE Explorer, the global bindings must include the libraries that have the cells used in the test bench file. To include them, add the libraries in the *Library List*. To tell the Simulator which implementation of each cell to use and in what priority, in our case:

```
spectre verilogams verilog veriloga schematic symbol
```

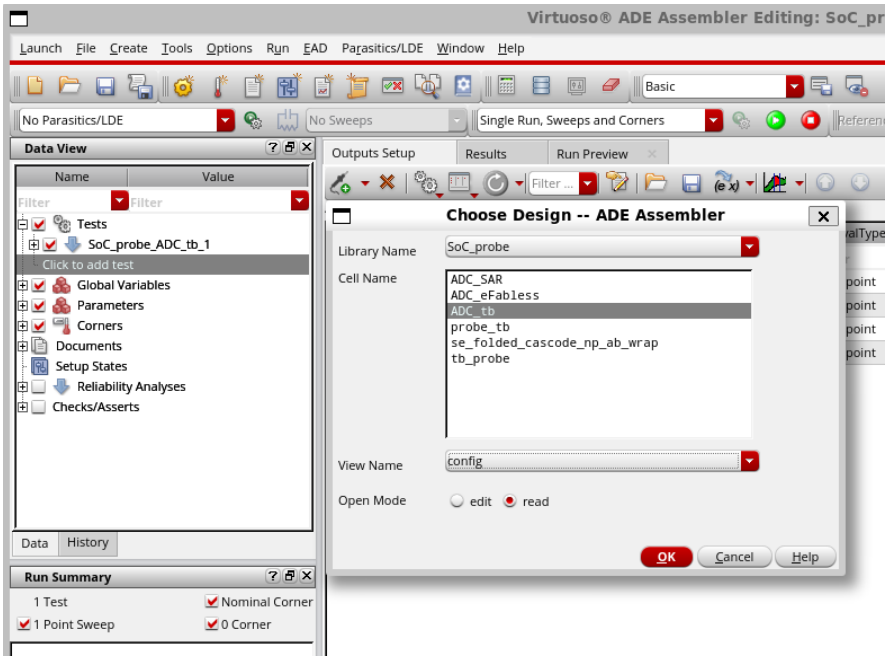
For simulating the behaviour we don't need a constraint list. Once every field is filled, click ok. In the tree view of the config file, it must have the test bench cell and descend in the cells included in the test bench file. Choose the correct view to use for each cell.



Click save and launch the ADE Explorer: **Launch->ADE Explorer**. Because it is the first time, we will create a new maestro view. Click ok and create the new view.

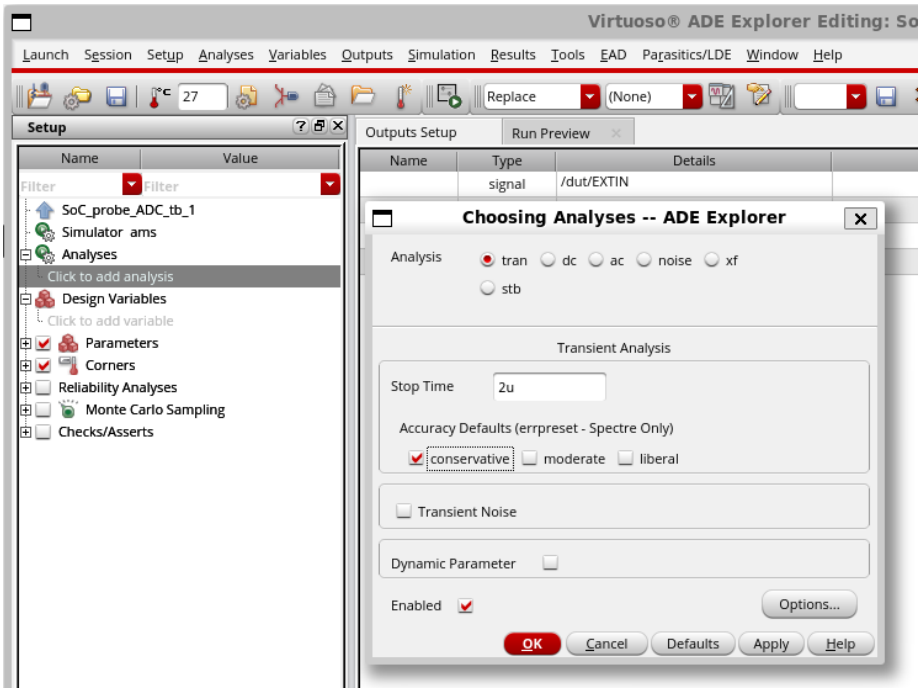


In the simulator click *add test* from the Data View window and choose the test bench cell with the *config* view. Click ok.

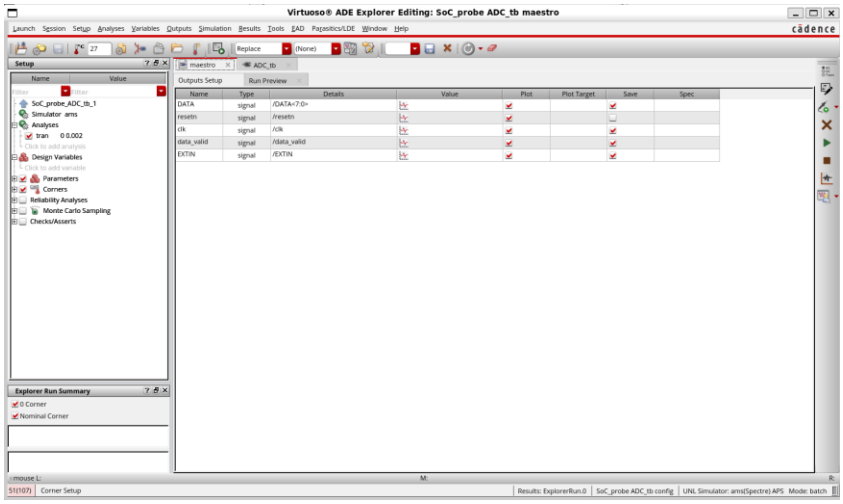


In the new test make sure that the chosen simulator is ams from the setup window. If by default the option is Spectre, make doble click and choose ams. **In the scenario** that choosing ams fails, the test wasn't created from *config* view.

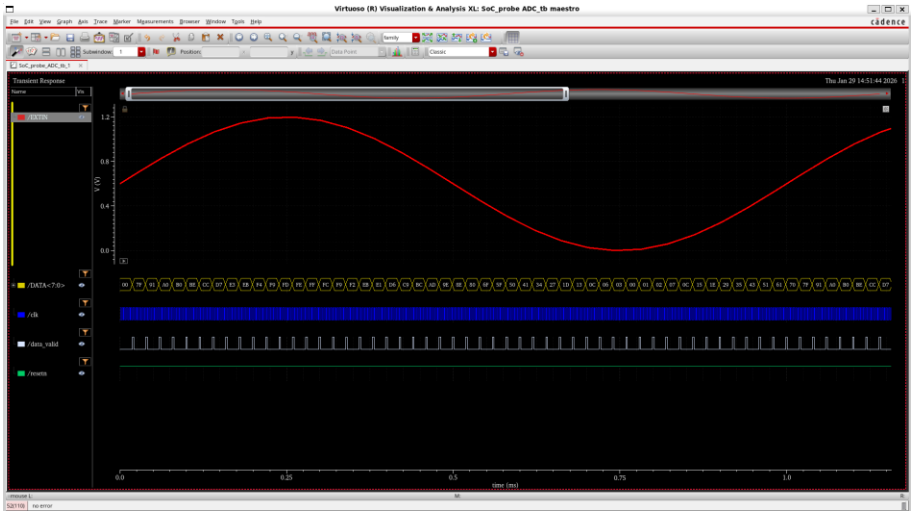
Click on add analysis and create tran analysis for your simulation. Click ok.



To add outputs to your test, go to: **Outputs->To Be Plotted->Select On Design.** In the new window choose the nets from the cell you want.



Run your simulation and plot the output.



Notice that each conversion takes 8 cycles and data_valid reg is asserted when conversion is done.