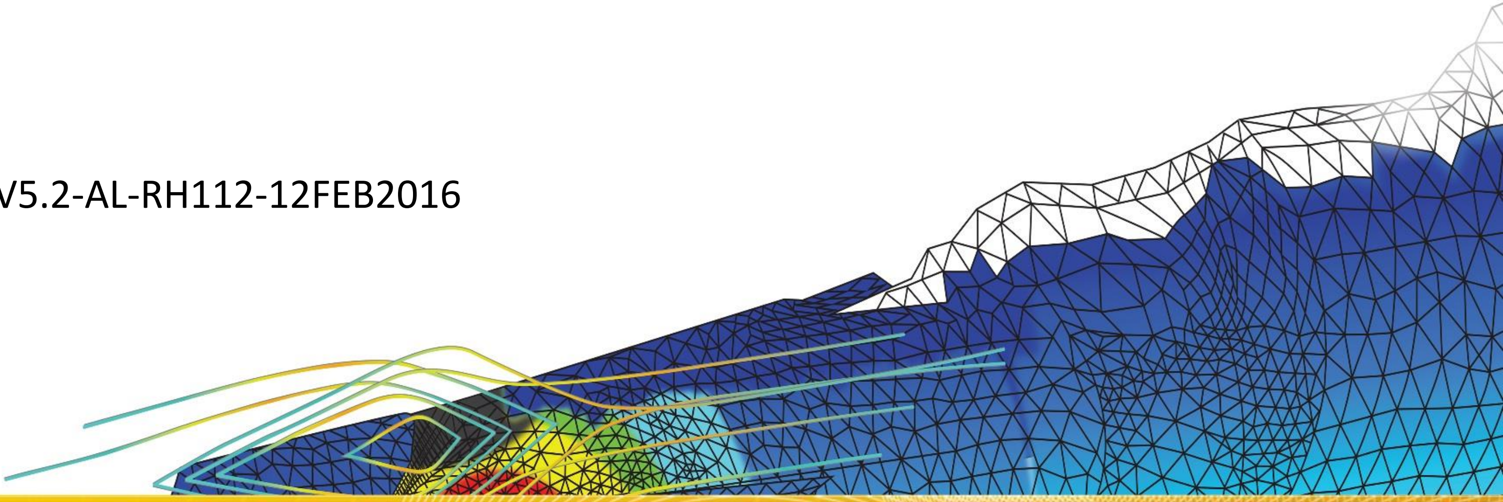
Setting Up For RedHawk™

VERSION: V5.2-AL-RH112-12FEB2016



Agenda

- **Licensing, rh_setup utility**
- **Input Data Requirements**
- **Documentation**

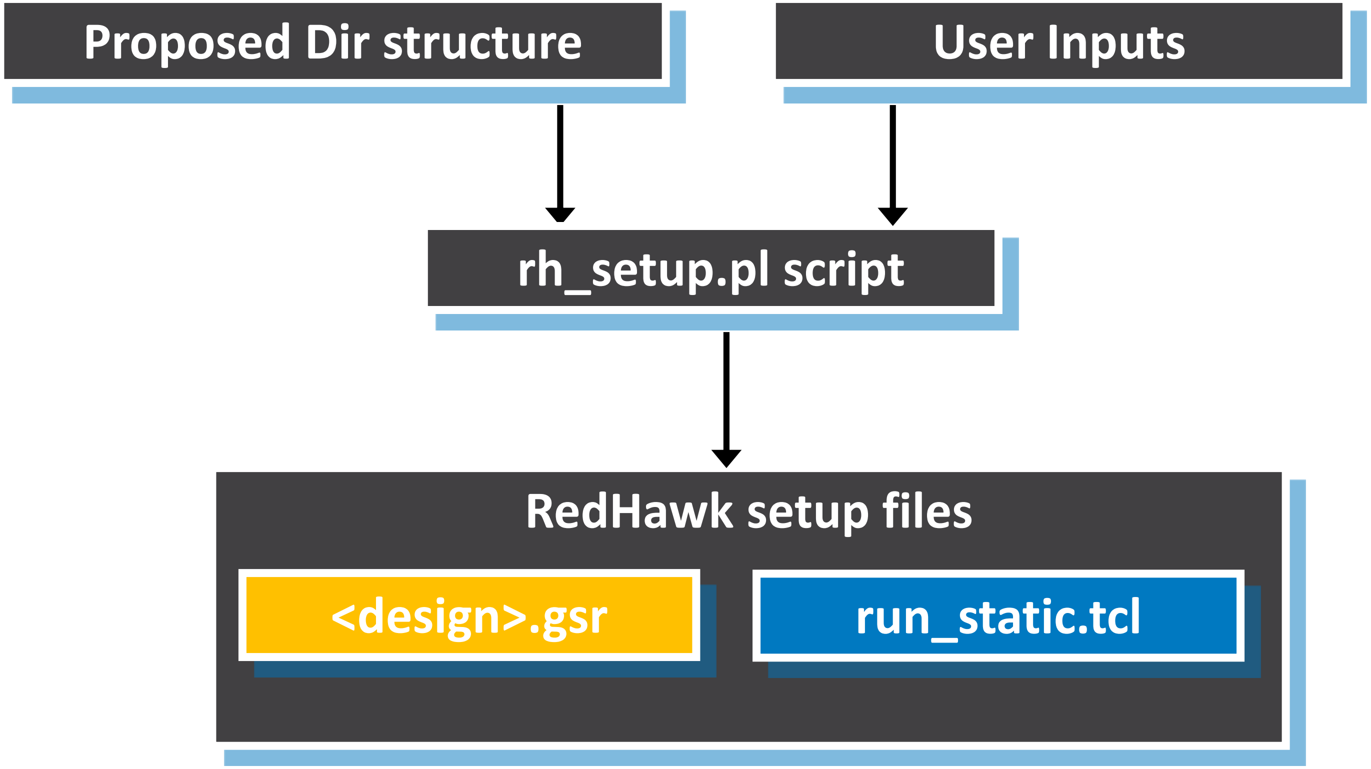
Licensing & Running RedHawk

- **To run RedHawk, set the RedHawk path and license**
  - setenv APACHEROOT <choose the version installed on your server>
  - set path = ($APACHEROOT/bin $path)
  - setenv LM_LICENSE_FILE <To your Apache license>
  - Then execute RedHawk as:
  -     redhawk &

rh_setup Utility

- Utility which simplifies setup tasks for new users – walks you through
- Builds GSR (Global System Requirement) and run command files
- Automatically finds data files if directory structure complies with the recommended one
- Allows information to be added incrementally

rh_setup Utility



Proposed Dir structure

User Inputs

rh_setup.pl script

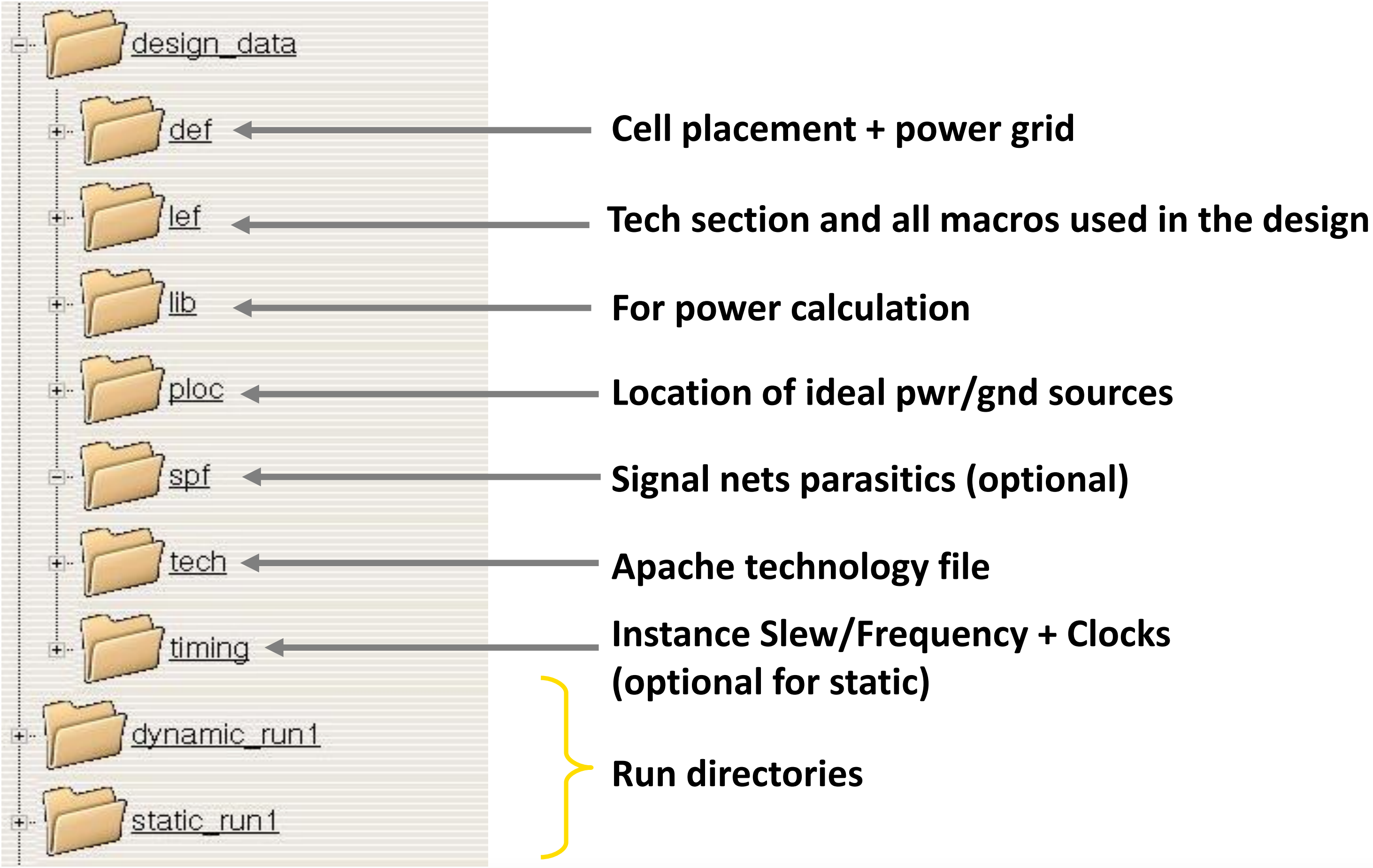
RedHawk setup files

<design>.gsr

run_static.tcl

Setting Up for rh_setup Run

- **Proposed Directory Structure**



- design_data
  - def
  - lef
  - lib
  - ploc
  - spf
  - tech
  - timing
- dynamic_run1
- static_run1

Cell placement + power grid

Tech section and all macros used in the design

For power calculation

Location of ideal pwr/gnd sources

Signal nets parasitics (optional)

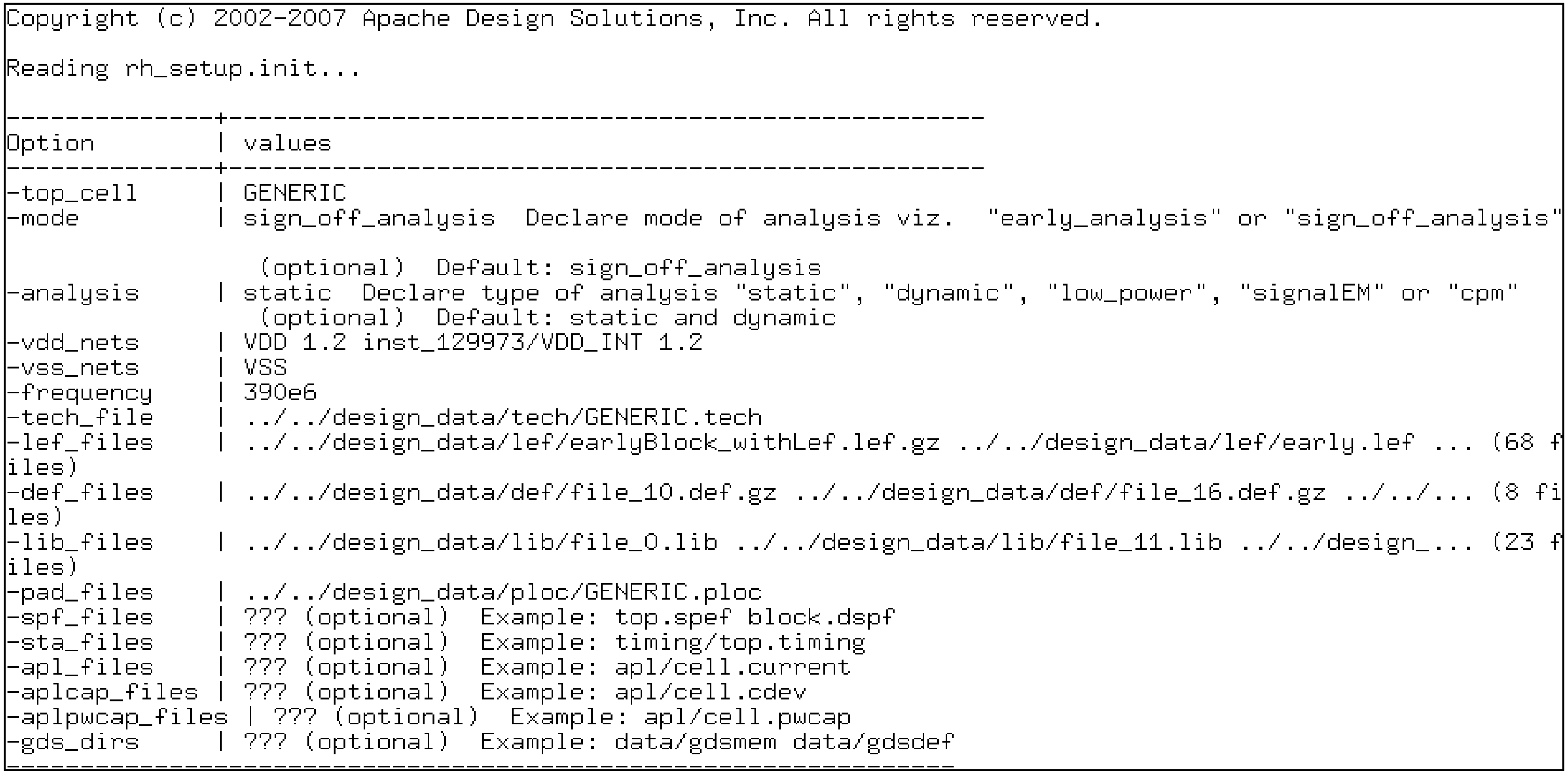
Apache technology file

Instance Slew/Frequency + Clocks (optional for static)

Run directories

rh_setup Script

rh_setup.pl [-h gives detailed explanation]



```
Copyright (c) 2002-2007 Apache Design Solutions, Inc. All rights reserved.

Reading rh_setup.init...

--------------+--------------------------------------------------
Option        | values
--------------+--------------------------------------------------
-top_cell     | GENERIC
-mode         | sign_off_analysis  Declare mode of analysis viz.  "early_analysis" or "sign_off_analysis"

                 (optional)  Default: sign_off_analysis
-analysis     | static  Declare type of analysis "static", "dynamic", "low_power", "signalEM" or "cpm"
                 (optional)  Default: static and dynamic
-vdd_nets     | VDD 1.2 inst_129973/VDD_INT 1.2
-vss_nets     | VSS
-frequency    | 390e6
-tech_file    | ../../design_data/tech/GENERIC.tech
-lef_files    | ../../design_data/lef/earlyBlock_withLef.lef.gz ../../design_data/lef/early.lef ... (68 f
iles)
-def_files    | ../../design_data/def/file_10.def.gz ../../design_data/def/file_16.def.gz ../../... (8 fi
les)
-lib_files    | ../../design_data/lib/file_0.lib ../../design_data/lib/file_11.lib ../../design_... (23 f
iles)
-pad_files    | ../../design_data/ploc/GENERIC.ploc
-spf_files    | ??? (optional)  Example: top.spef block.dspf
-sta_files    | ??? (optional)  Example: timing/top.timing
-apl_files    | ??? (optional)  Example: apl/cell.current
-aplcap_files | ??? (optional)  Example: apl/cell.cdev
-aplpwcap_files | ??? (optional)  Example: apl/cell.pwcap
-gds_dirs     | ??? (optional)  Example: data/gdsmem data/gdsdef
----------------------------------------------------------------
```

Use GSR template with fixed data locations to populate fields easily.

rh_setup

- **Utility Supports following analysis types**
  - Static IR/EM Analysis
  - Dynamic Analysis
  - Low Power Analysis
  - Chip Power Model (CPM)
- **Utility Supports following modes**
  - Early Analysis
  - Sign Off Analysis
- **Example usage:**
  - rh_setup.pl -top GENERIC -vdd VDD 1.0 -vss VSS -freq 100e6

rh_setup

Example usage:

- **Case 1: If the proposed directory structure is followed**

```
rh_setup.pl -top GENERIC –analysis static –mode sign_off_analysis -
vdd VDD 1.0 -vss VSS -freq 100e6
```

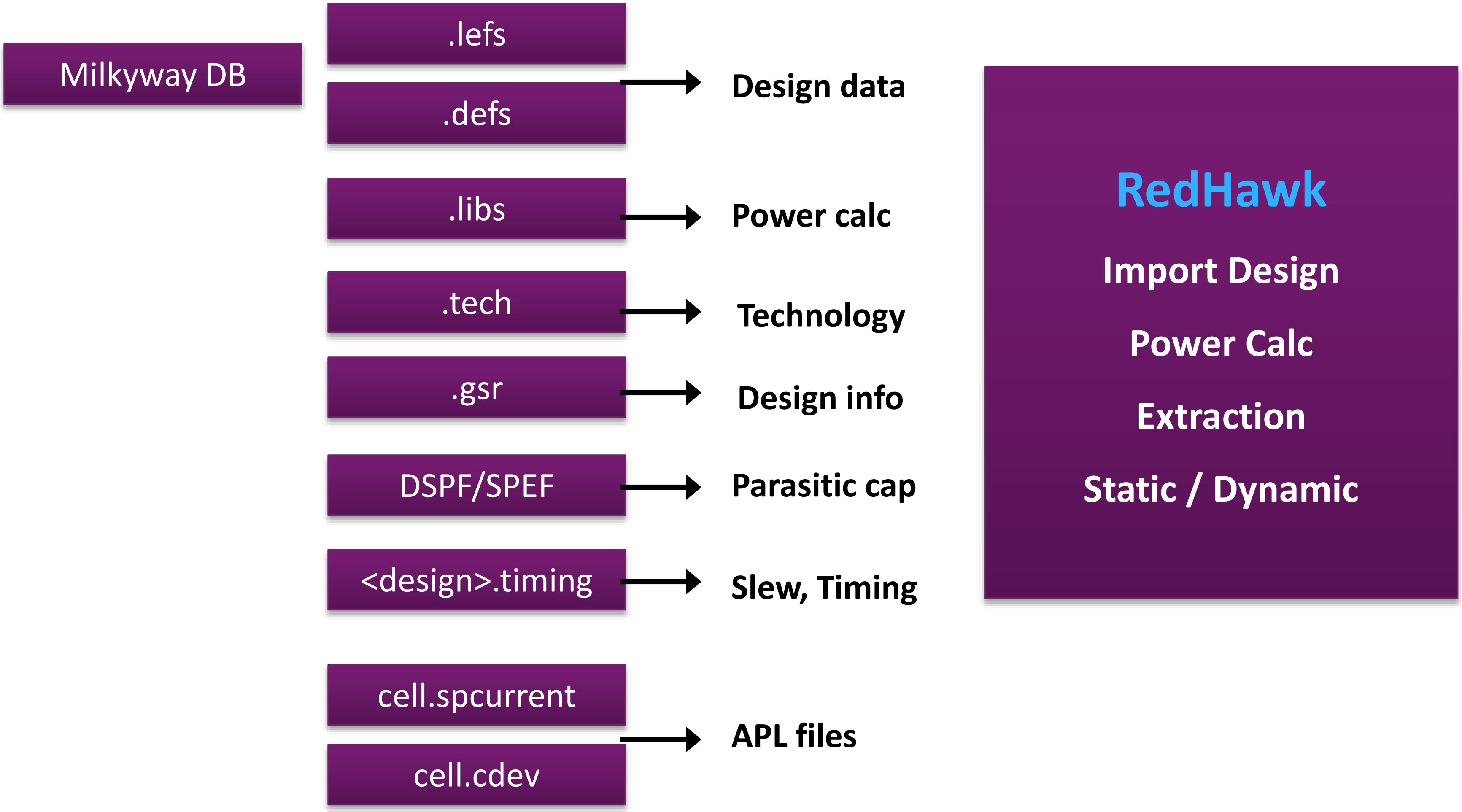
- **Case2: If custom directory structure is followed**

```
rh_setup.pl -top GENERIC –analysis static
                    –mode   sign_off_analysis
                    -def_files /nfs/yamuna.data/GENERIC/def/*
                    -lef_files  /nfs/yamuna.data/GENERIC/lef/*
                    -lib_files /nfs/yamuna.data/GENERIC/libs/*
                    -vdd VDD 1.0 -vss VSS -freq 100e6
```

.lefs, .defs, .libs, .gsr, .tech, .ploc

Input Data Preparation

RedHawk Input Files



Milkyway DB

.lefs

.defs

Design data

.libs

Power calc

.tech

Technology

.gsr

Design info

DSPF/SPEF

Parasitic cap

<design>.timing

Slew, Timing

cell.spcurrent

cell.cdev

APL files

RedHawk

Import Design

Power Calc

Extraction

Static / Dynamic

Data Preparation

Design Data

DEF, LEF, Libs, Pads
Technology
SPEF, STA
Package netlist

Characterization (APL)

Current Profiles
Capacitance, ESR
Leakage
Memory models
Delay/Slew

Simulation Conditions

Voltage
Process, Temperature
Power/Ground nets
Modes

GDS Translations

Memories, IPs
RDL layers

RedHawk Input Files

FILE FORMATS:

1) **LEF: Library Exchange Format:** This is a industry standard format that has the information related to pin description and boundaries of the blocks/instances in the design.

2) **DEF: Design Exchange Format:** This contains logical and physical connectivity between different instances and blocks in the design.

3) **LIB: Synopsys Liberty file format:** This has several electrical and logical properties for a cell like: input and output pin properties, information on distributing power among the different power pins, internal energy of the cell, cell functionality information, etc.

4) **SPEF – SIGNAL Parasitic Exchange Format:** This file contains the parasitic (RC) associated with each nets in the design.

Technology File (.tech)

```
metal <layer> {
    thickness <value>
    must if C and/or L extraction needed>
    resistance <value>
    resistance per square
    EM <value>
    EM current density in (current/length)
    above <above a dielectric_layer_name defined in
    dielectric>
    must if C and/or L extraction needed and no Height is
    defined
    default NA
}
```

Use the “*rhtech*” utility to create the apache tech file from a STARRC-XT file

Technology File Generation

Apache provides several utilities to convert various tech file formats

- ircx2tech : for converting TSMC ircx file to apache format
- rhtech : for converting STARRC-XT NXTGRD/ITF files to apache format

Technology File *(Cont’d)*

```
via <via_layer_name>  {
    width { <width> }
    resistance <value>
    Via resistance
    The resistance is specified as it is for the via
    EM <value>
    Electromigration current limit for the via
    UpperLayer <metal_layer_name; must>
    LowerLayer <metal_layer_name; must>
}

viamodel <viamodel_name>  {
    <metal_layer_name> <X1> <Y1> <X2> <Y2>
    <via_layer_name> <X1> <Y1> <X2> <Y2>
    <metal_layer_name> <X1> <Y1> <X2> <Y2>
}

For vias, “width” or “viamodel” must be specified
```

Technology File *(Cont'd)*

```
dielectric <dname> {
    constant <value; must if C and/or L extraction needed>
    thickness <value defined in length unit above; must>

    height <value defined in length unit above>
    must if Above is not specified

    above <dielectric_layer_name>
    above which dielectric layer
    must if Height is not specified
}

Needed for capacitance and/or inductance extraction
```

Technology File *(Cont'd)*

```
units {
 capacitance 1p
 inductance  1n
 resistance  1
 length      1u
 current     1m
 voltage     1
 power       1
 time        1n
 frequency   1me
}
metal metal1 {
  Thickness 0.18
  T           25
  Tnom        110
  Coeff_RT1   0.00265
  Coeff_RT2   -2.641e-07
  EM          1.509
  EM_ADJUST   0.016
  above       ILD_B
}
 via via1
{
  Width       { 0.1 }
  Resistance  1.5
  T           25
  Tnom        110
  Coeff_RT1   0.0007815
  Coeff_RT2   -2.574e-06
  EM          0.158
  UpperLayer  metal2
  LowerLayer  metal1
}
```

```
ddielectric APACHE_E
{
  constant    3.9
  thickness   1.47
  Height      8.59
}
dielectric APACHE_D6
{
  constant    3.9
  thickness   0.6
  above       APACHE_D5b
}
dielectric APACHE_D5b
{
  constant    3.9
  thickness   0.4
  above       APACHE_D5a
}
dielectric APACHE_D5a
{
  constant    3.9
  thickness   1.45
  above       APACHE_D4
}
dielectric APACHE_D4
{
  constant    3.9
  thickness   0.25
  above       APACHE_D3
}
```

Tech File – Advanced Keywords

- **Most of the advanced technology keywords required especially for 65 nm and below technologies are supported. Example:**
  - ETCH_VS_WIDTH_AND_SPACING
  - RHO_VS_WIDTH_AND_SPACING
  - POLYNOMIAL_BASED_THICKNESS_VARIATION
  - RPSQ_VS_WIDTH_AND_SPACING
  - THICKNESS_VS_WIDTH_AND_SPACING
  - SIDE_TANGENT
  - RPV_VS_AREA

Identification of Voltage Sources

- **Describes the source for the power and ground nets**
  - Can be specified through PAD_FILES keyword in GSR
  - Can also import the pads using “Import pad” command
- **Inside the voltage source file, we can specify the sources in different ways**
  - Pad instance (*PAD section)
  - <pad_cell_name_1> [<pin_name>| <pin_name> <layer_name>]
  - Pad master cell (*PCELL section)
  - <master cell name>
  - <x source loc> <y source loc> <layer> <P/G pad type>
  - Pin location list (*PLOC section)
  - <Net name> <x coord> <y coord> <layer> <POWER | GROUND>
  - Pad location with package (* PLOC_PSS)
  - Pad master cell used along with package (*PAD_PSS)
- **RedHawk will automatically identify the PINS from DEF if you use GSR keyword 'ADD_PLOC_FROM_TOP_DEF 1” .**

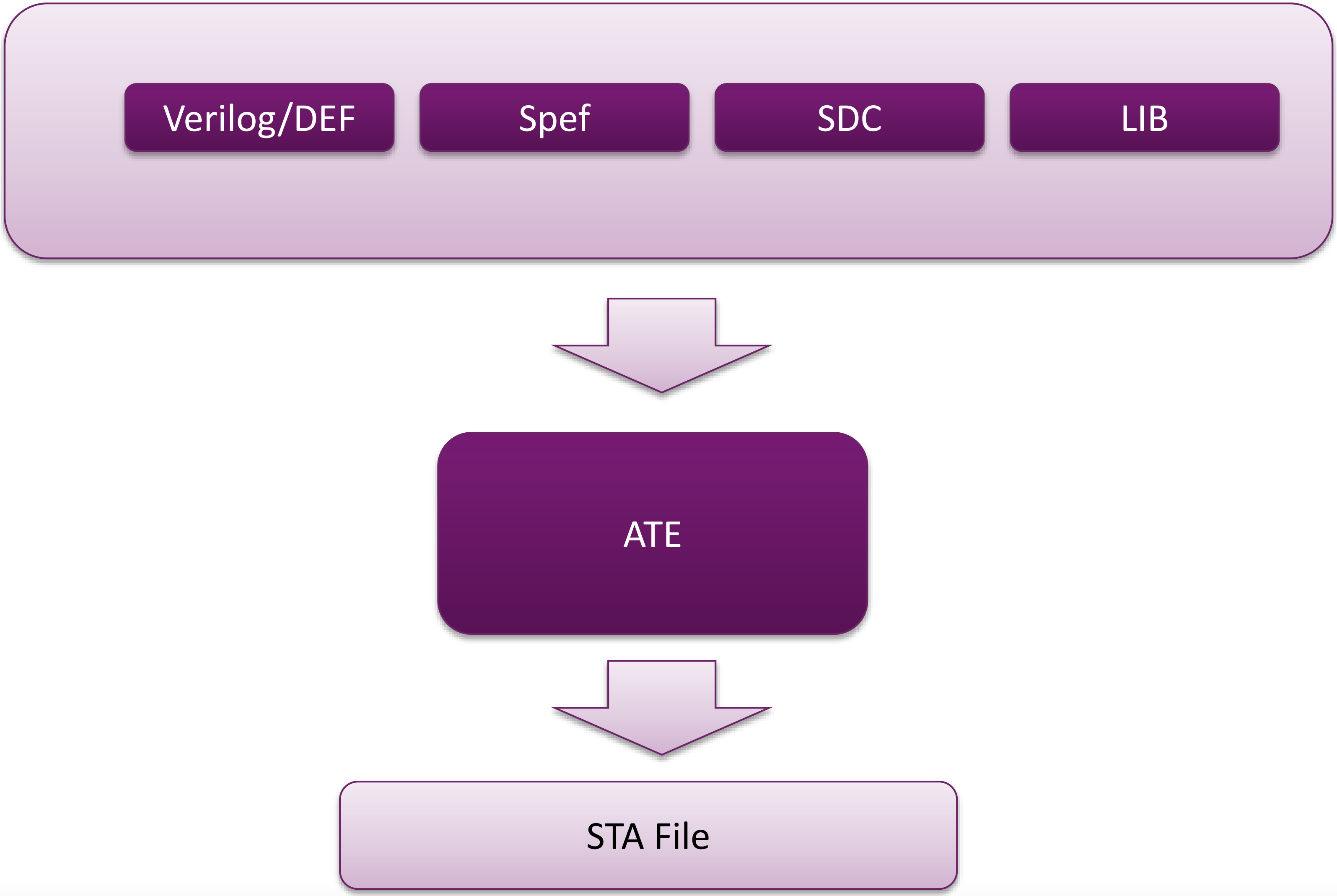
Identification of Voltage Sources *(Cont'd)*

```
*PCELL
    DVDD12
    DVSS
    PASLZ55 VDD
    PADLZ55 VSS

    *PAD
    VDD_PAD1
    VSS_PAD45
    PVDD1DGZ
    17.5 242.0 METAL6 POWER


*PLOC
    DVDD1    4905    878.85  METAL4  POWER
    DVSS1    4880    938.85  METAL4  GROUND
    DVDD2    4905    998.85  METAL4  POWER
```

What is Inside STA File ?

- **Slew**
  - Required for Static (Power calculation uses Slew)
  - Required for Dynamic (Current w/f is dependent on Slew)
- **Timing Windows**
  - Not required for static
- **Instance Frequency**
  - Required for static and dynamic
- **Clock domain info**
  - Required for static and dynamic

STA File Generation



STA File Generation *(Cont'd)*

ATE generates timing file for RedHawk. Usage is shown below :

1. Setting up installation path :

```
setenv APACHEROOT RH_install_dir
setenv PATH $APACHEROOT/bin:$PATH
```

2. Creating the setup file <design_name>.ate

```
set synopsys_lib {
<list of liberty files>
}
set verilog_netlist {
<list of verilog files>
}
set spef {
<list of signal spef files>
}
set timing_constraints {
<list of timing constraint files>
}
```

Note: ATE can also take DEF netlist as the input, instead of Verilog netlist. In order to specify DEF files as design netlist, use '**def_netlist**' keyword, instead of '**verilog_netlist**'.

STA File Generation *(Cont'd)*

3. Creating command file

```
set errorAction     continue

LoadGeneralParam

DataPreparation       -files all
LoadLibrary           -error_action $errorAction
LoadNetlist           -error_action $errorAction
LoadParasiticFile     -error_action $errorAction -ground_coupled_caps
LoadTimingConstraint  -error_action $errorAction

ta_set_clock_delay    -propagated [get_clocks *]

getSTA * -gz
```

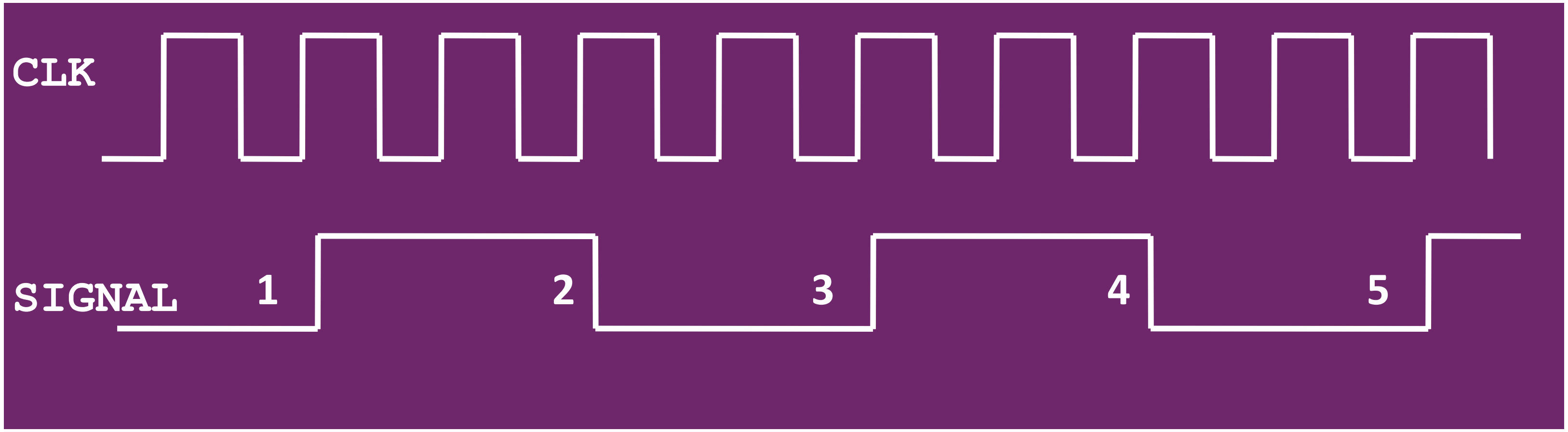
4. Running ATE and getting results

```
ate ate.cmd >& ate.log
```

STA File Usage

	Power Calculation	Static Analysis	Dynamic Analysis
Clock Domain	✓	✓	✓
Instance Slew	✓	✓	✓
Instance Frequency	✓	✓	✓
Instance Timing Window			✓

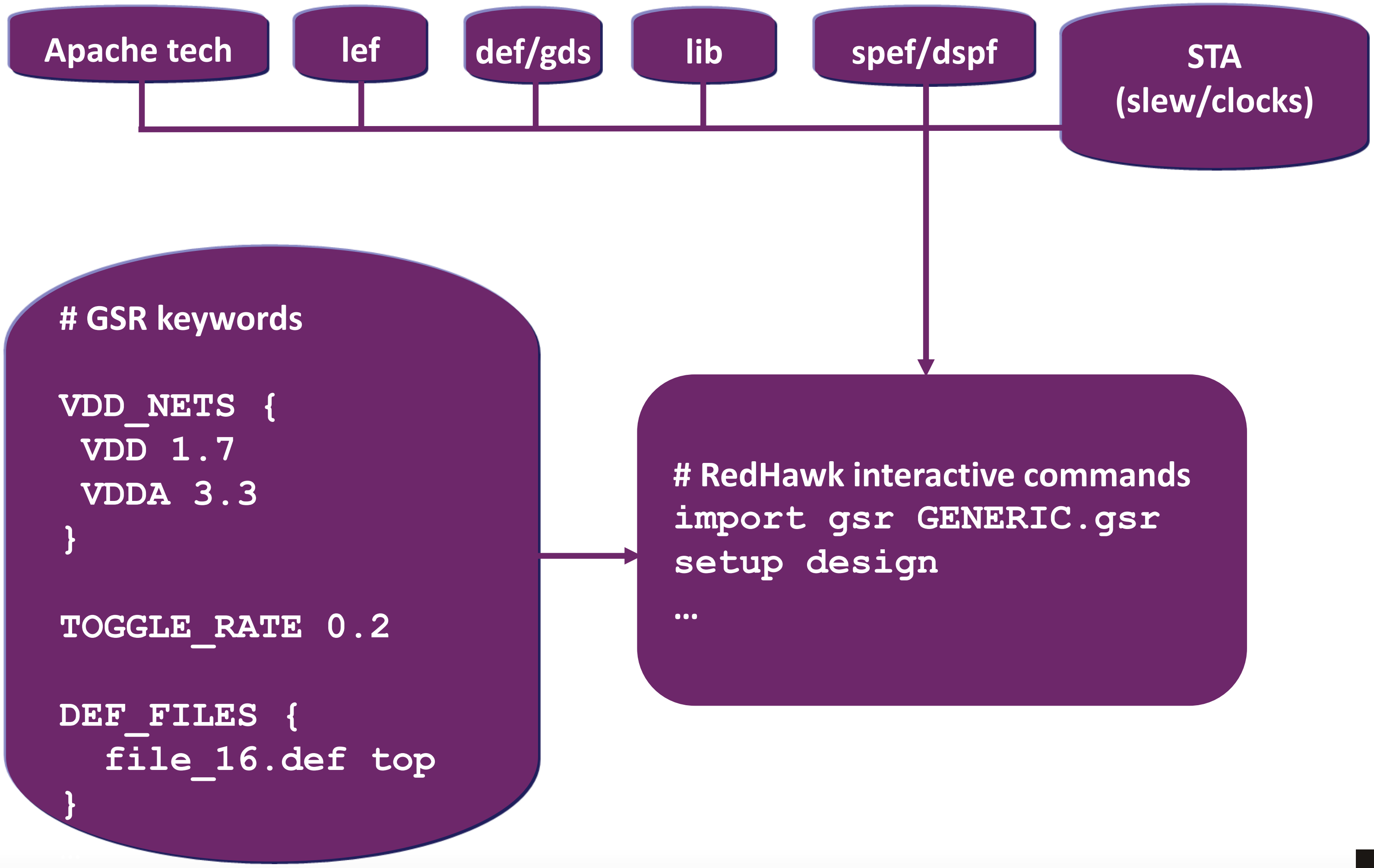
Toggle Rate



- A Toggle is 0→1 or 1→0 transition
- Toggle rate=(no. of transitions)/(no. of cycles)
- Toggle rate CLK=2
- Toggle rate SIGNAL=0.5

```
GSR keyword:

TOGGLE_RATE 0.5 2
```

Global System Requirement File
(aka. user controls)



Apache tech

lef

def/gds

lib

spef/dspf

STA
(slew/clocks)

```
# GSR keywords

VDD_NETS {
 VDD 1.7
 VDDA 3.3
}

TOGGLE_RATE 0.2

DEF_FILES {
  file_16.def top
}
```

```
# RedHawk interactive commands
import gsr GENERIC.gsr
setup design
…
```

GSR File Overview

```
TECH_FILE ads.tech

LIB_FILES {                                                       LIB_FILES {
<path to lib file>                                          OR    <design>.libs
<path to lib directory> (all *.lib files in dir)                  }
<path to custom lib file> custom
}

LEF_FILES {                                                       LEF_FILES {
<lef file path>/name1.lef << tech definition                OR    <design>.lefs
<lef file path>/name2.lef                                         }
}

DEF_FILES {                                                 OR
<def file path>/name1.def                                         DEF_FILES {
<def file path>/name2.def TOP < last one to be TOP DEF            <design>.defs
}                                                                 }
```

GSR File Overview *(Cont'd)*

```
PAD_FILES {
pad file path name/name1.pad
}

GDS_CELLS {
cell_name1 <path to dir where files for cellname1
reside>
cell_name2 <path to dir where files for cellname2
reside>
}

GSC_FILE <path and name of GSC file>
```

GSR File Overview *(Cont’d)*

```
# Net switching activity
TOGGLE_RATE <value>

# Block specific toggle information
BLOCK_TOGGLE_RATE {
<block_name> <value>
...
}

# Obtain toggle from VCD
VCD_FILE {
...
}

# Instance specific toggle
INSTANCE_TOGGLE_RATE {
<name of instance> <toggle rate>
}
```

Order of toggle selection

- VCD_FILE
- INSTANCE_TOGGLE_RATE / INSTANCE_TOGGLE_RATE_FILE
- BLOCK_TOGGLE_RATE / BLOCK_TOGGLE_RATE_FILE
- TOGGLE_RATE

GSR File Overview *(Cont'd)*

```
# Design timing information
STA_FILE {
FREQ_OF_MISSING_INSTANCES <value in Hz>
<name of design> <design timing data>
}

# Dominant frequency of design
FREQUENCY <value in Hz>

# Input transition time
INPUT_TRANSITION <value in s>
```

- **From running TCL program**
- **The frequency value that captures most of the power in the design**

GSR File Overview *(Cont’d)*

```
# Power specification
BLOCK_POWER_FOR_SCALING {
FULLCHIP <design_name/block/instance>
<total power>
CELLTYPE <cell name> <power>
<block name> <instance name> <power>
}

INSTANCE_POWER_FILE {
<name of file>
}
```

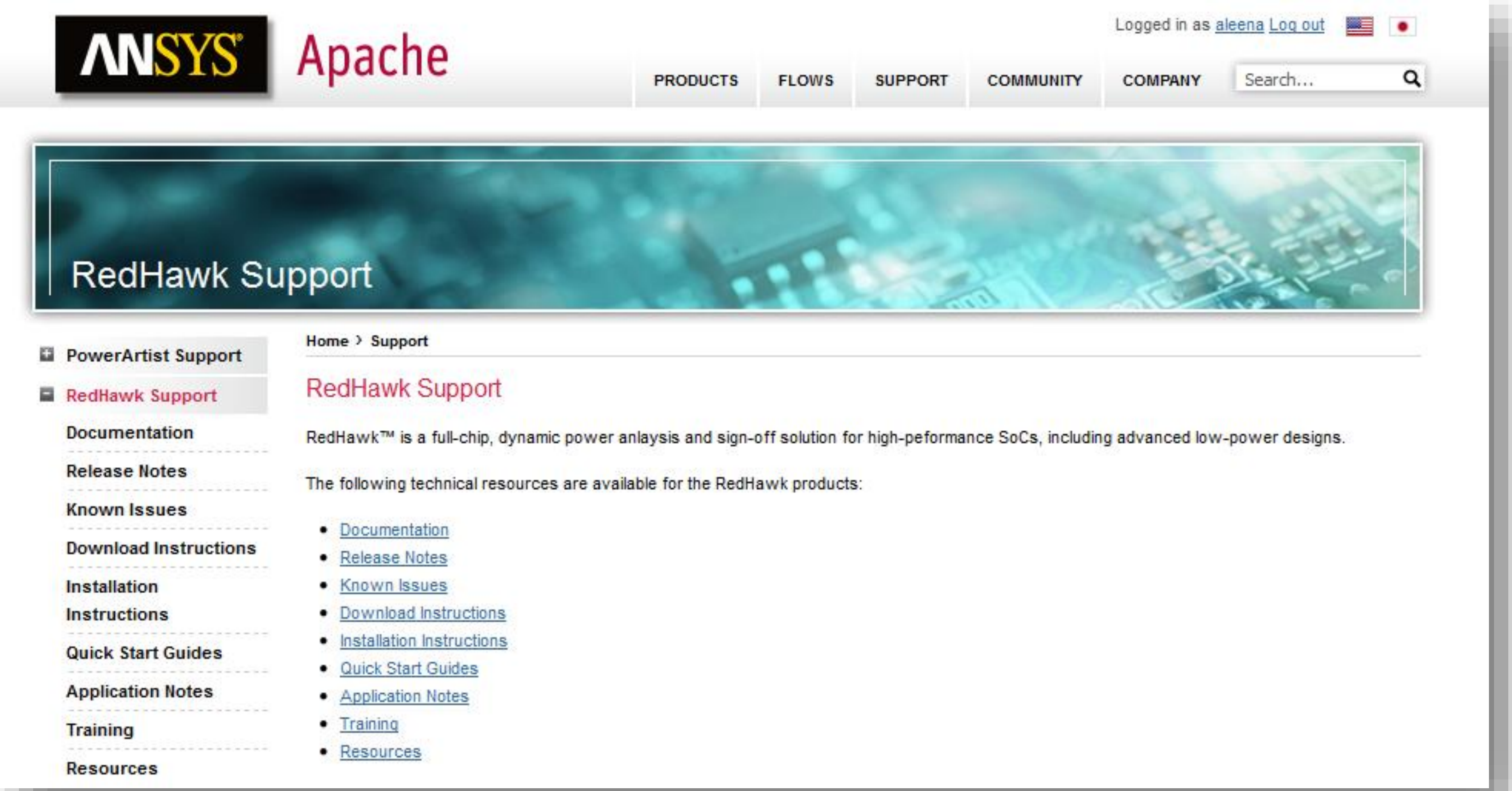
- **Fullchip or block or cell power can be specified**
- **Honor user provided instance specific power**

Run Command File Overview

```
# Import data
import gsr GENERIC.gsr
setup design

# Calculate power
perform pwrcalc


# Power/Ground grid extraction
perform extraction -power -ground

# Lumped resistance (in Ohms)
# for package, wirebond and pads
setup package -power -r 0.005 -l 2.5 -c 5
setup package -ground -r 0.005 -l 2.5 -c 5
setup wirebond -power -r 0.01 -l 2.2 -c 1.42
setup wirebond -ground -r 0.05 -l 1.7 -c 0.2
setup pad -power -r 0.001
setup pad -ground -r 0.001

# Static IR analysis
perform analysis -static
explore design
```

How to get Help!!

- Apache Online Customer Support Center
  - http://support.apache-da.com
  - Email: support@apache-da.com



RedHawk Support

RedHawk™ is a full-chip, dynamic power anlaysis and sign-off solution for high-peformance SoCs, including advanced low-power designs.

The following technical resources are available for the RedHawk products:

- Documentation
- Release Notes
- Known Issues
- Download Instructions
- Installation Instructions
- Quick Start Guides
- Application Notes
- Training
- Resources

Thank You!!!