Learn the OpenAccess API

Using Python

Initial Contribution By

James Masters
Intel - 2013
Updates & Additions
by Silicon Integration Initiative - 2020

Section 5 - The tech file

- Create a tech.db file within an OpenAccess library
- Understand Database Units and user units
- Define layers and purposes
- Define Vias

Overview

- OA design data uses a technology definition to identify how the design is used
- OA technology information is stored in an OA library
  - OA design library can have technology information directly embedded or the OA design library can attach or reference another library for technology information
    - See “Using Technology Databases” in the Programmers Guide
- Among other things, the OA “tech” library contains:
  - Layer and purpose definitions
  - Via definitions
  - Database/user unit conversion

Database Units and User Units

- For measurements, the OA tech defines DBU/UU
  - DBU = Database units (integers stored internally)
  - UU = User units (exposed to the user – e.g. microns)
  - DBU/UU = conversion factor of database units per user units
- The DBU/UU conversion factor is defined for each viewType

  (see [get/set]DBUPerUU method calls in oaTech class)
  - Default for “maskLayout” is “microns”
  - Default for “schema,cSymbol” and “symbol” is “inches”
- The oaTech class offers convenience methods to convert between DBU and UU (see dbuToUU and uuToDBU methods)
- This is FYI only, we will not alter the DBU/UU factor in this training

Layers and Purposes

- A design layer is represented with a class below the abstract base oaLayer class like oaPhysicalLayer
  - Layers in a design are stored with a layer number and the tech has more information behind that layer like the layer's name and material
  - A physical layer's material may optionally be specified; see the oaPhysicalLayer and oaMaterial classes in the API doc for more info
- A design "purpose" is a kind of sub-category for a layer
  - Default purpose is "drawing" which is the basic input for a layer's shapes
  - Other purposes may exist for different layer usage like a "pin" purpose
- There is no default definition of valid layer-purpose pair (LPP) combinations in OA but this may be defined by an application as an extension

Creating a Tech Library

- A “tech library” is established by simply adding technology information to an OA library

```
libname   = "mylib"
lib = oa.oaLib.find(libname)
if not lib:
print "Cannot find library %s" % libname  sys.exit(1)

tech = oa.oaTech.create(lib)

# Add other tech attributes here like layers, purposes,
# etc...

tech.save()
```

Creating a Layer / Purpose

- To create a layer and purpose, you first need an oaTech object created (in an oaLib)
- Note, the material type (e.g. oa.oacMetalMaterial) is optional but may be useful in some OA--based applications

```
oa.oaPhysicalLayer.create(tech, "m1", 5, oa.oacMetalMaterial)
  oa.oaPurpose.create(tech, "resistorID", 0)
```

Defining Standard Vias

- Vias can be handled as tech file definitions or can be instantiated from a library
- The oaStdViaDef class is used to define standard vias
- An array of oaViaParam enums are set to specific values and are used to define specific dimensions of the via.

```
stdviaparams = oa.oaViaParam()

stdviaparams.setCutLayer(v1.getNumber())
stdviaparams.setCutHeight(350)
stdviaparams.setCutWidth(350)
L1Enc = oa.oaVector(40,200)
stdviaparams.setLayer1Enc(L1Enc)
L2Enc = oa.oaVector(200,40)
stdviaparams.setLayer2Enc(L2Enc)

viadef = oa.oaStdViaDef.create(tech,"testvia", m1, m2,stdviaparams)
```

Setting Database Units From The Tech File

- The oaTech class has methods for setting the database unit to user unit ratio for a given view type

```
oa.oaTech.setDBUPerUU(<techfile pointer>, <view type>, <integer>)
```

Lab 5.1

- Goals:
  - Learn how to add technology information to an existing OA library
  - Learn how to create layers and purposes
  - Learn how to define vias and set DBU from the tech file
- Create a script to:
  1. Open an existing library (../mylib)
  2. Add layers: nwell, ndiff, pdiff, poly, cont, m1, v1, m2
  3. Add purposes: resistorID, pin
  4. Define a via
  5. Set DBU per UU to:
     - maskLayout = 2000, schematic = 100, schemticSymbol = 100
- Compare your script to labs/5.1/makeTechFile.py

Lab 5.2: Create an oaTech “dump” Utility

- Goals:
  - Become more familiar with tech files
  - Learn to work with tech file contents
- Create a script to:
  1. Read the library name from the command line
  2. Show the layers, purposes, vias, DUBperUU

     Note: there are more purposes than you defined

     These predefined purposes are actually positive integer representations of negative numbers

  Use the library created in lab 5.1

- Compare your script to labs/5.2/dumpTechInfo.py

Section 5 Summary

- Create a tech file in an OpenAccess library
- Add technology information to the tech file
- Extract the contents of a tech file from another application

Silicon Integration Initiative

www.si2.org

For details contact Marshall Tiner

Director of Production Standards

mtiner@si2.org