Incremental Technology Database (ITDB) Usage and Practices

Ted Paone
Core Comp Architect
IC6.1 Adoption
Cadence Design Systems

CDN Live EMEA

Session 1.1

Incremental Technology Data Practices

Constraints and Constraint Groups

What is ITDB?

- Local Technology Data
- Technology references
- Technology graph

How does it operate?

- How to set it up
- How you access technology constraints using ITDB

Splitting up your technology data

- Determining data ownership
- Where should you define constraints
- Avoiding and resolving conflicts

ITDB examples

Constraints - Where did my design rules go?

Design rules are now process technology constraints

- Designed for interoperability
  - Design and electrical rules
  - All tools access the same binary technology data
  - One representation for all tools
- Fixed format
  - Fixed names
  - One, two or three layer types
  - Fixed value types

```
(minSpacing "Metal1" .04)

(viaSpacing "Via1" 3 .5 .03)
```

```
;; Constraint group foundry
( "foundry"      nil
 spacings(
   ( minWidth "well" 0.6 )
   ( minSameNetSpacing "Nwell" 0.4 )
   ( minSpacing "Nwell" 0.6 )
   ( minSpacing "Oxide"
                 "Poly" 0.1 )
   ( viaSpacing "Via1" (3 0.21 0.2))
 ) ; spacings
 orderedSpacings(
   ( minEnclosure "Metal1"
                       "Via" 0.005 )
   ( minOppExtension "Metal1"
                         "Via" 0.01 )
 ) ; orderedSpacings
) ; foundry
```

Constraint Groups

Hierarchical grouping of constraints

- Can reference one or more CGs in another CG
- Called a “member” of another CG
- Multiple levels can be used any purpose such as define DFM rules
- Depth first search for finding specific rules
- **foundry** CG checked if rule is not found
- Some constraint groups are used specifically to store constraints for specific tools
  - `lefDefaultRouteSpec`
  - `virtuosoDefaultSetup`

```
( "M9maximumYeild" t

   memberConstraintGroups(

; listed in order of precedence
; ------------------------------
       "M7M9highYeild"
       "M7M9routingRules"
   ) ;memberConstraintGroups

   spacings(
    ( minSpacing "Metal6" 0.255 )
    ( minWidth   "Metal6" 0.18 )
   );spacings

   memberConstraintGroups(
       "M4M5RoutingRules"
    ) ;memberConstraintGroups

   spacings(
   ( minSpacing "Metal3" 0.265 )
   );spacings

) ;M9maximumYeild
```

Incremental Tech Lib

A library can contain technology data as well as have tech libs bound

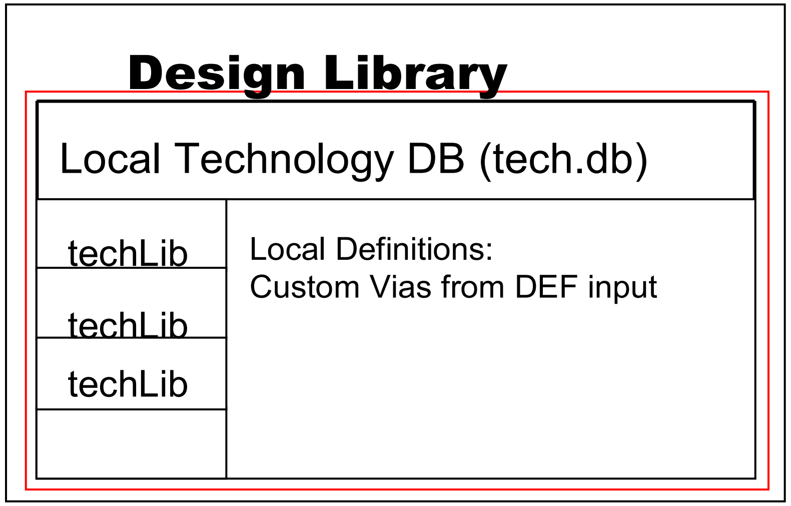
Technology libraries are bound using an ordered hierarchy

Technology library inherits technology data from other tech libs

- Local definitions can override previous ones

Design libraries can inherit technology data from many tech libs

- Design specific technology data (vias) can be stored in design library techDB



Reference vs. Attachment

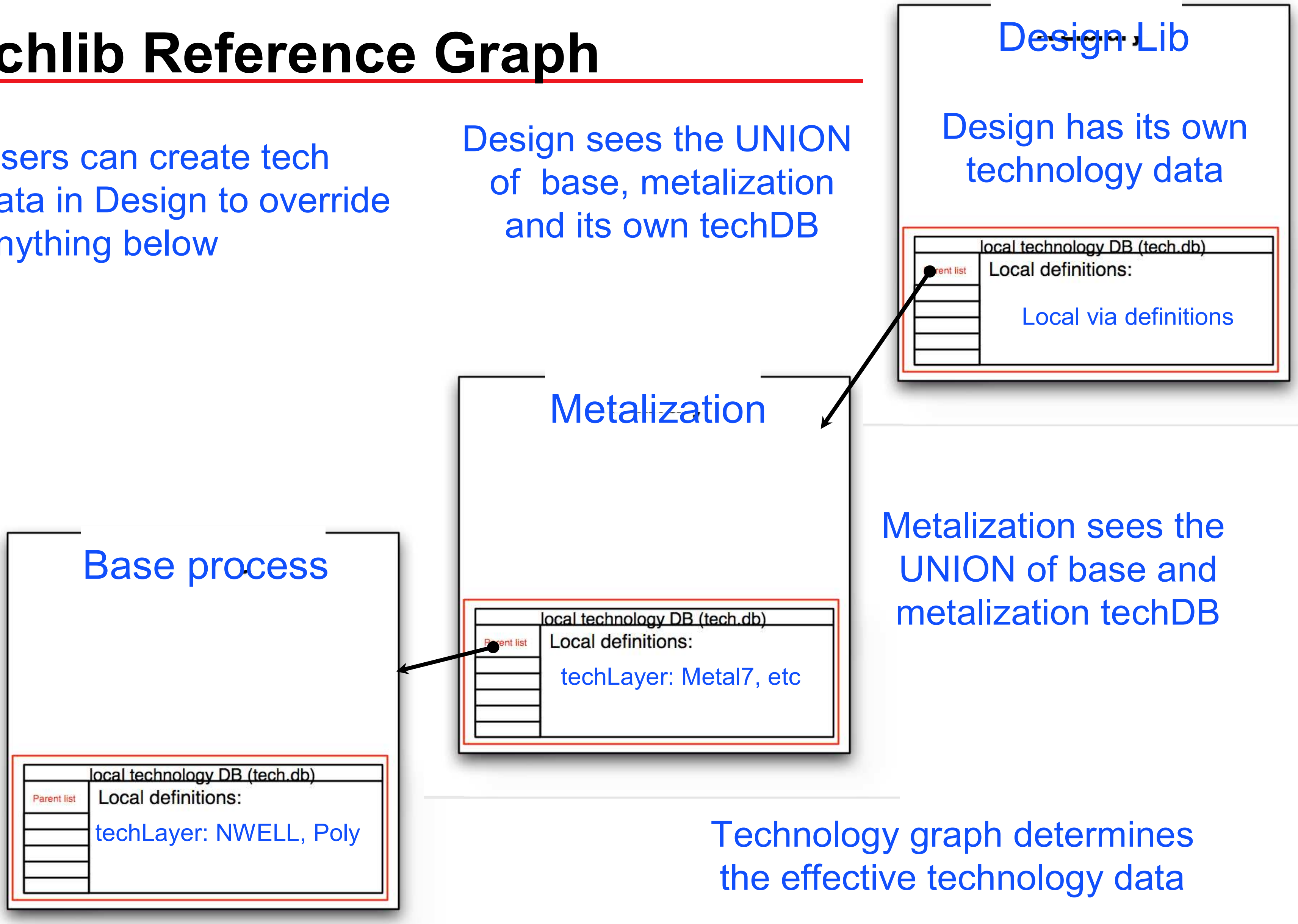
When they create a new library they can either specify to “attach” to or to "reference" an already existing technology library

- **Attachment**: End users access the data from the attached techDB.
  - The attached techDB may be incremental
    - Totally transparent to the final users.
  - No addition/modification can be made to the techDB
- **Reference**: End users inherit the union all the definitions from the various techDB's
  - Local techDB is created
    - Library specific definitions can be added to the techDB

Techlib Reference Graph

Users can create tech data in Design to override anything below

Design sees the UNION of base, metalization and its own techDB



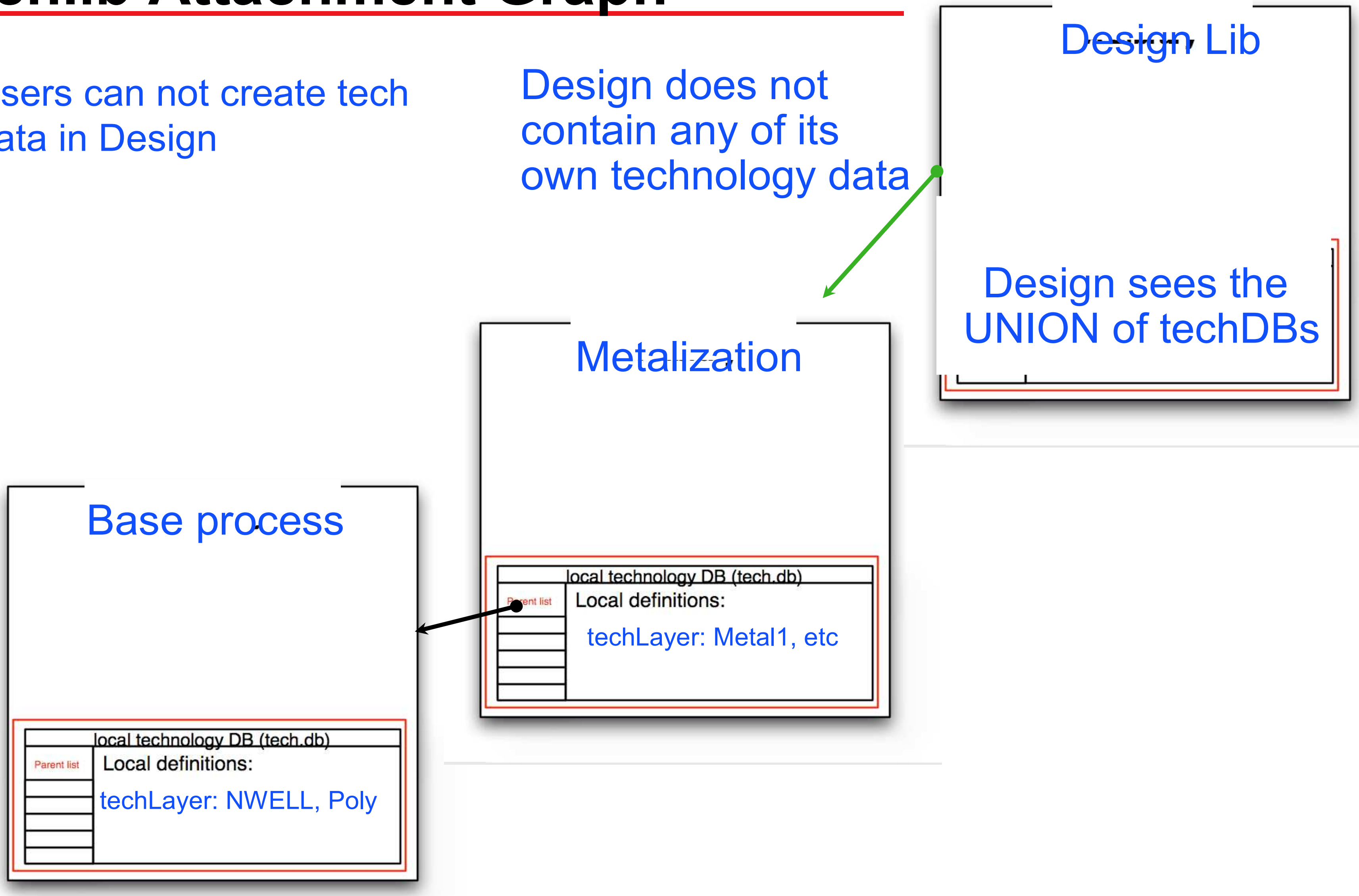
Metalization sees the UNION of base and metalization techDB

Technology graph determines the effective technology data

Techlib Attachment Graph

Users can not create tech data in Design

Design does not contain any of its own technology data



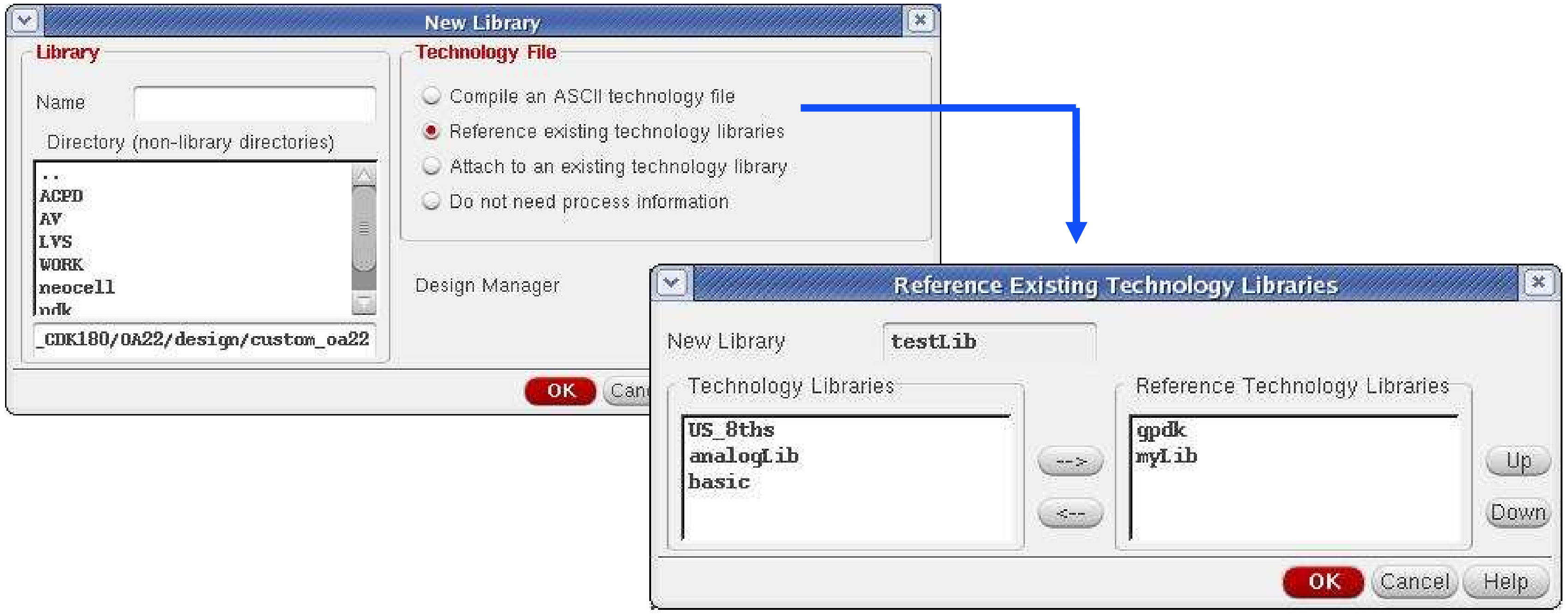
Referencing Other Libraries

To create a techDB with references only (no local techDB rules or constructs):

Create a new library

Select "Reference Existing Technology Libraries"

In the Reference form, move libraries to the right to reference them


New Library

Library

Name

Directory (non-library directories)

```
..
ACPD
AV
LVS
WORK
neocell
pdk
```

_CDK180/OA22/design/custom_oa22

Technology File

- ( ) Compile an ASCII technology file
- (●) Reference existing technology libraries
- ( ) Attach to an existing technology library
- ( ) Do not need process information

Design Manager

Reference Existing Technology Libraries

New Library: testLib

Technology Libraries:

```
US_8ths
analogLib
basic
```

Reference Technology Libraries:

```
gpdk
myLib
```

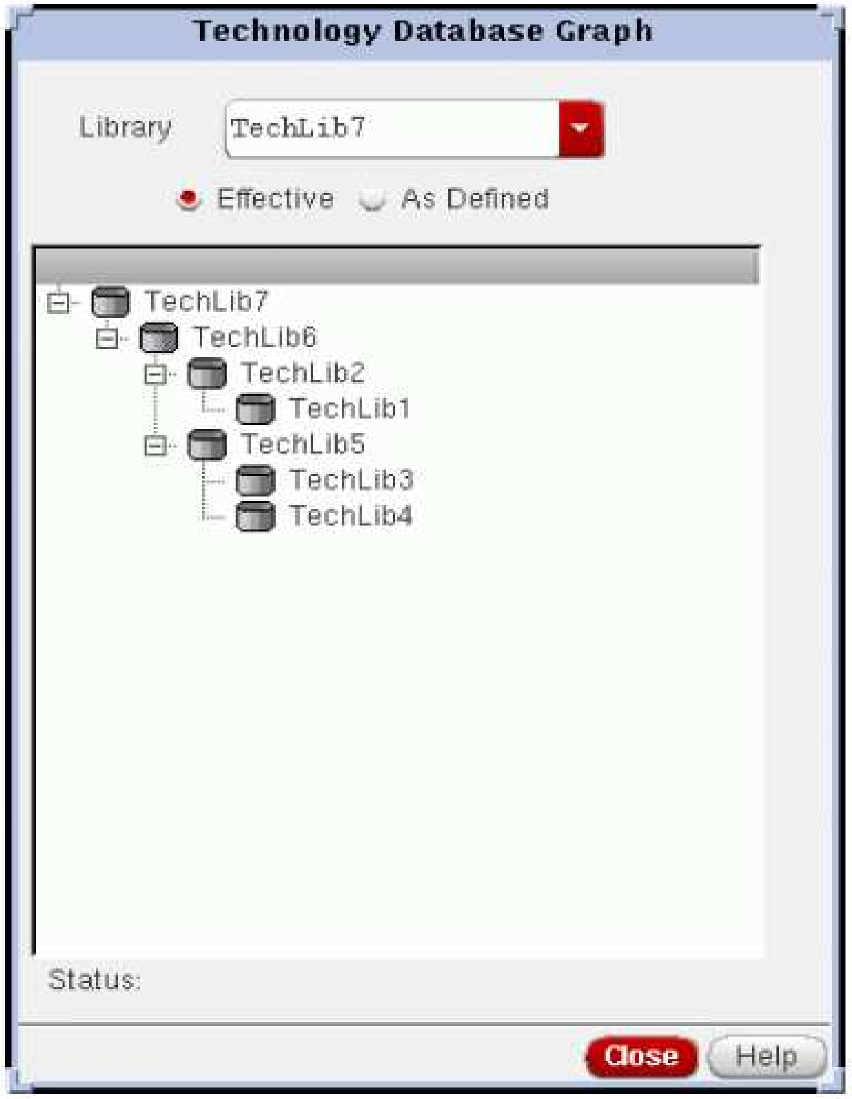
Loading Other Rules

If you want to reference other libraries *and* add local rules:

- Compile a local techfile
  - Add the appropriate reference libraries in the controls section
  - Add your layers, constraints, devices, and other constructs in the appropriate section

```
;*********************************

; CONTROLS


controls(

 refTechLibs(

; techLibName

; -----------

   "gpdk"

 ) ;refTechLibs

) ;controls



…

constraintGroups(

  ("myGroup" nil

   spacings(

     (minWidth "Oxide" 0.7)

   )   ))
```

Displaying the Technology Graph


Technology Database Graph

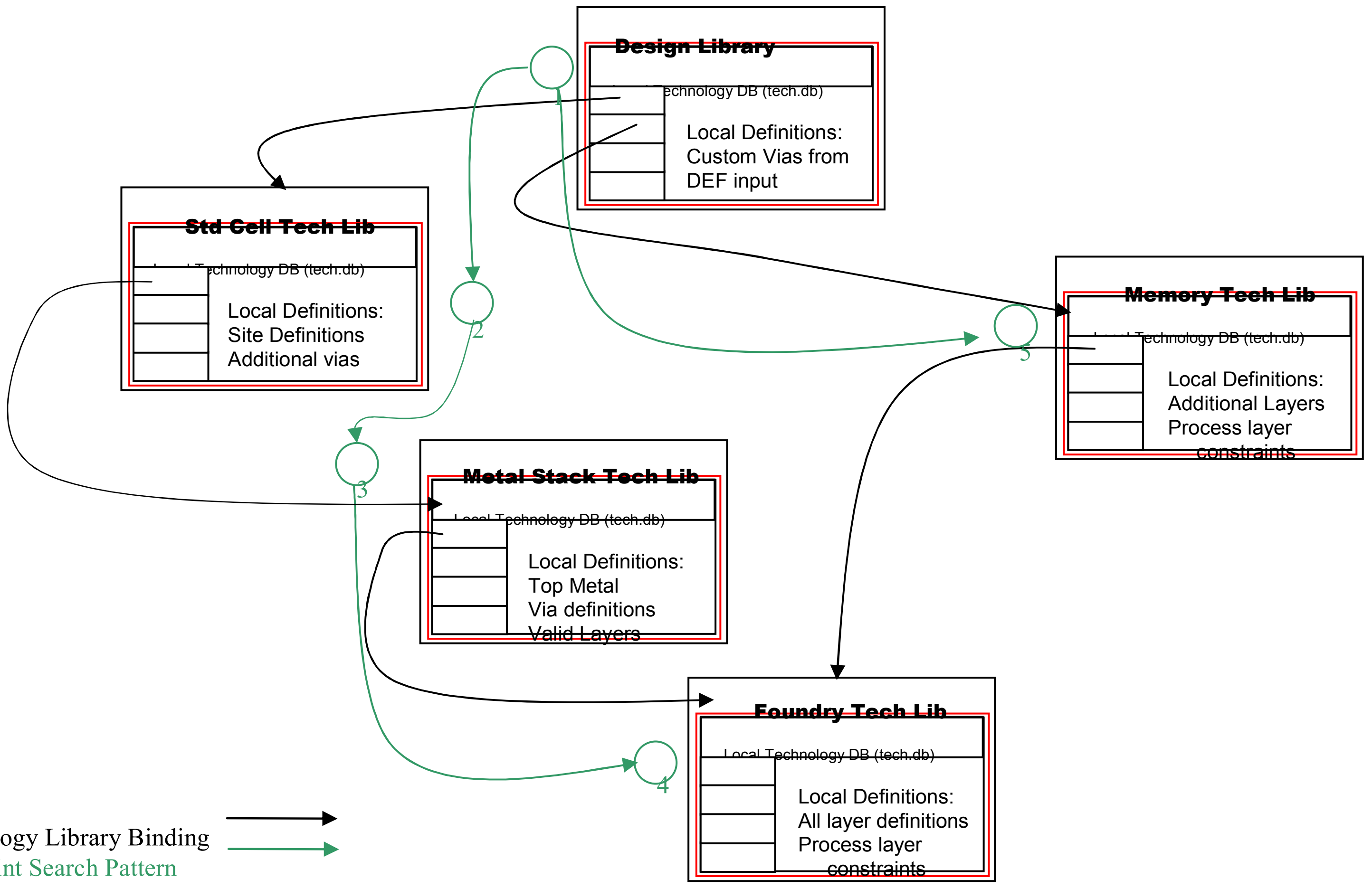
Library: TechLib7

[✓ Effective] As Defined

```
TechLib7
  TechLib6
    TechLib2
      TechLib1
    TechLib5
      TechLib3
      TechLib4
```

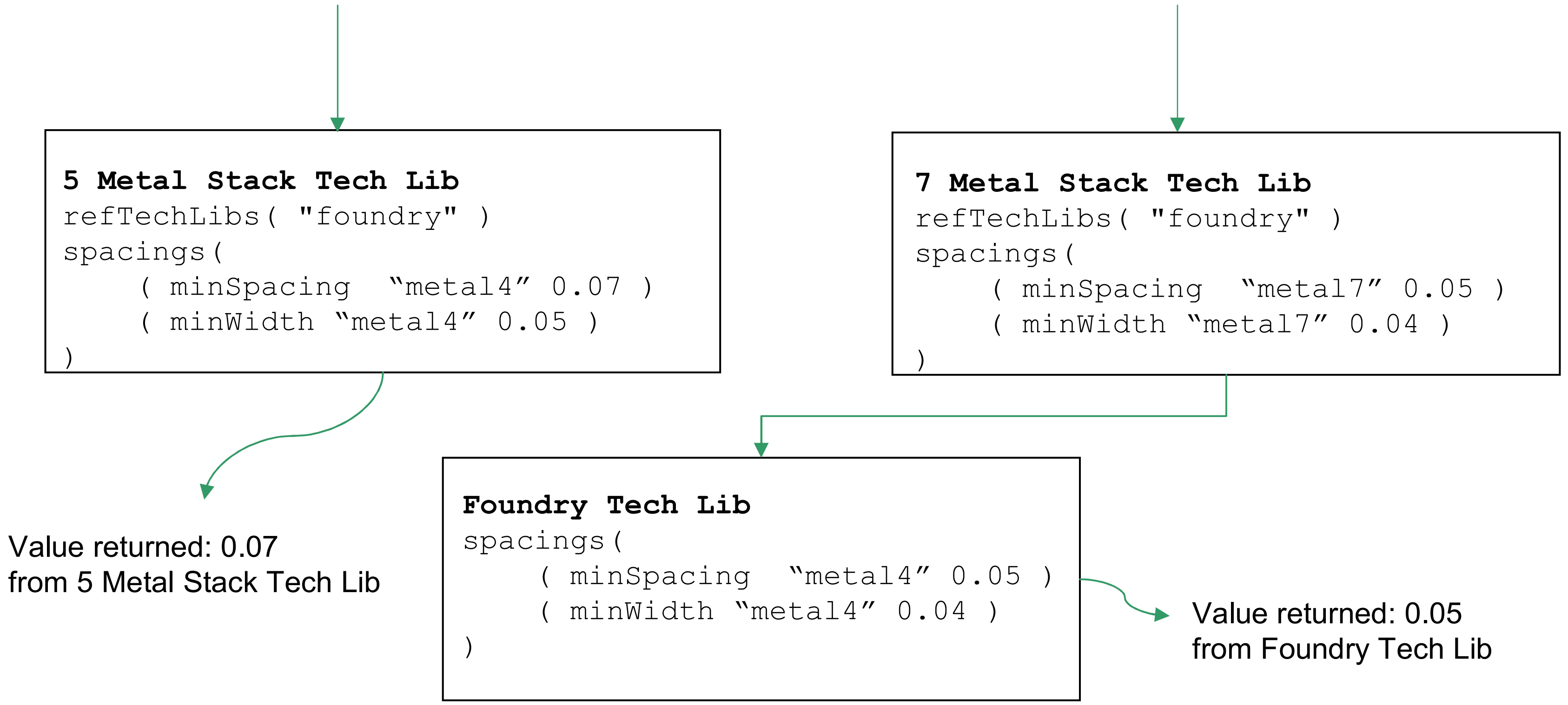
Status:

ITDB in Action: Depth-First Search



Technology Library Binding →
Constraint Search Pattern →

ITDB in Action: Constraint Search



```
5 Metal Stack Tech Lib
refTechLibs( "foundry" )
spacings(
     ( minSpacing  "metal4" 0.07 )
     ( minWidth "metal4" 0.05 )
)
```

```
7 Metal Stack Tech Lib
refTechLibs( "foundry" )
spacings(
     ( minSpacing  "metal7" 0.05 )
     ( minWidth "metal7" 0.04 )
)
```

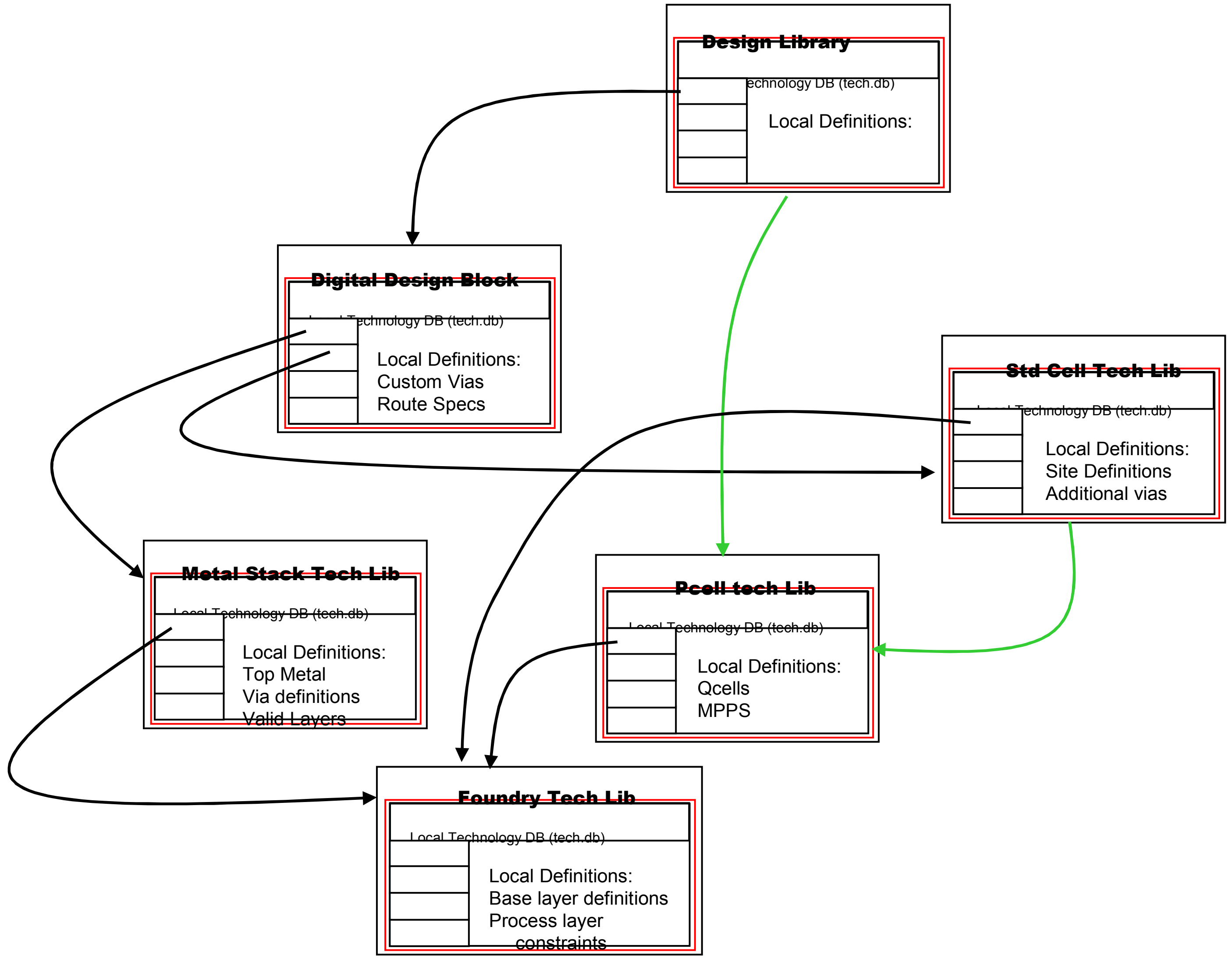
```
Foundry Tech Lib
spacings(
     ( minSpacing  "metal4" 0.05 )
     ( minWidth "metal4" 0.04 )
)
```

Value returned: 0.07
from 5 Metal Stack Tech Lib

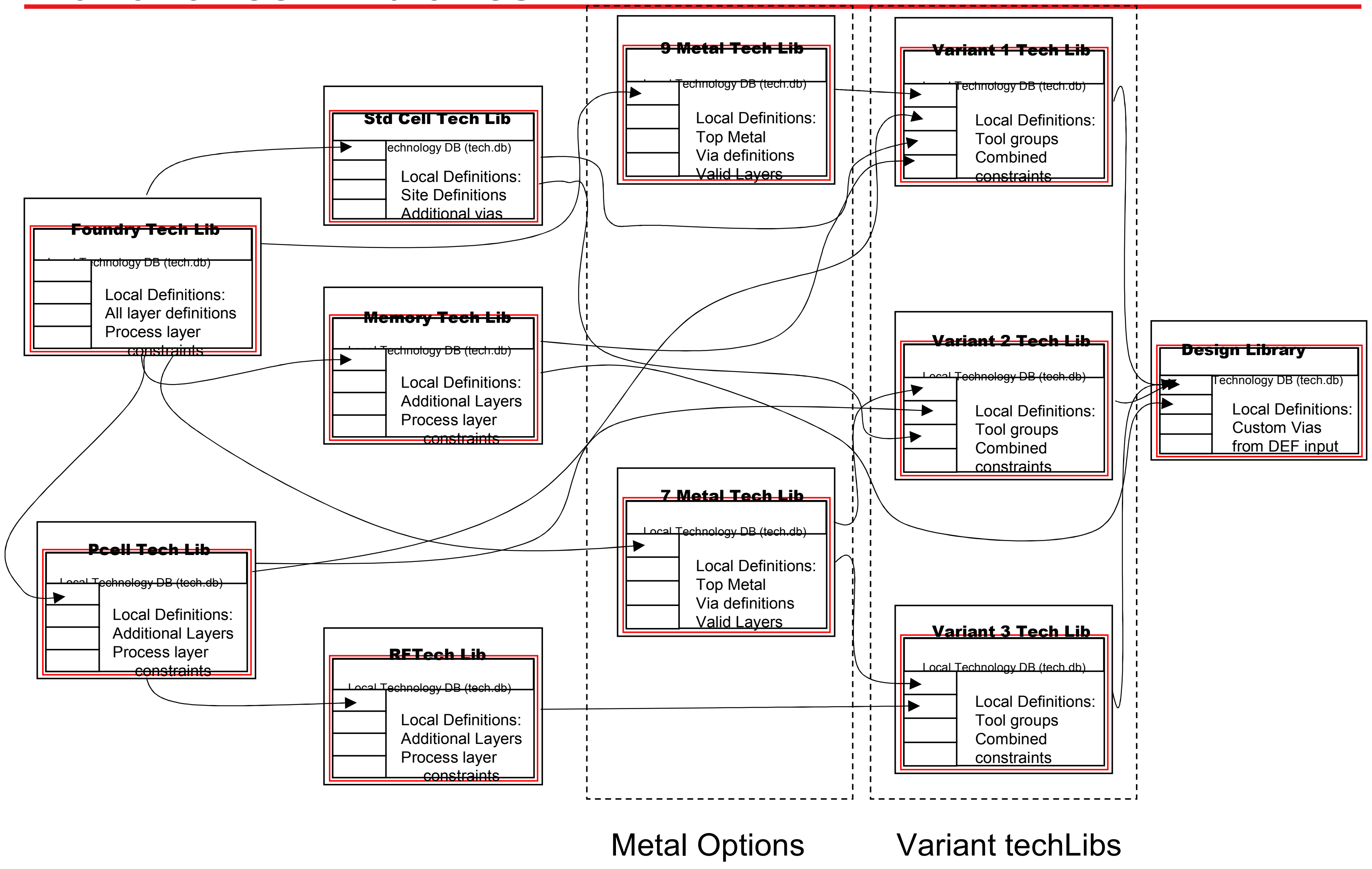
Value returned: 0.05
from Foundry Tech Lib

Search for minSpacing constraint for metal4

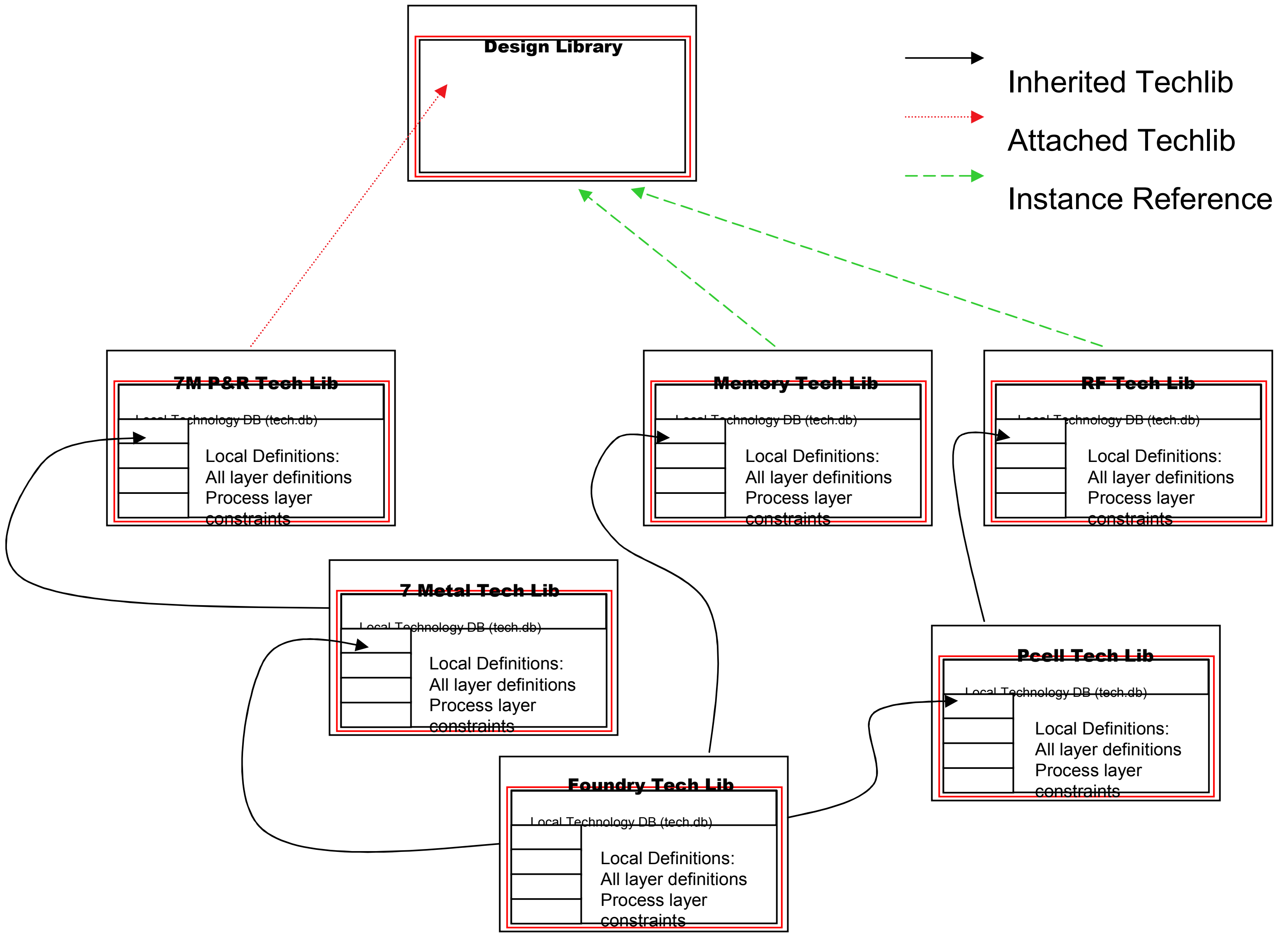
What goes where?



ITDB Use Model Example 1 – Variant Tech Libraries



ITDB Use Model Example 2 – Reference and Attachment



Conflicts

Most objects can only be defined once

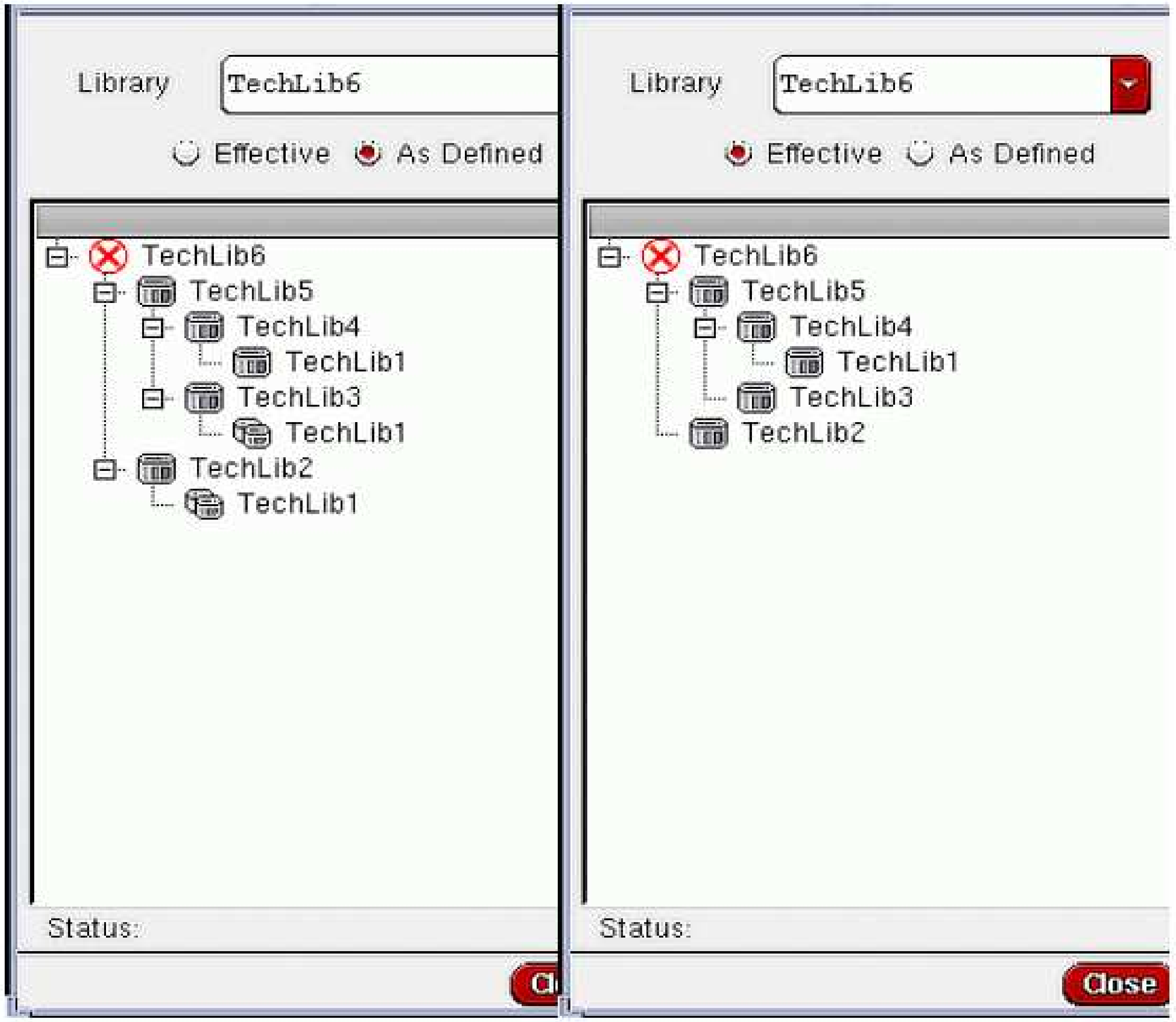
Conflict management

- General rule: “If it has a name, it cannot be redefined”
  - ViaDefs, layers, purposes, devices cannot be redefined
  - Constraint Groups cannot, but foundry constraints can
- Each type of object has a predefined set of conflict detection rules associated with it. For example:
  - Layers cannot be redefined in a techDB (same name or #)
  - VIADEF name must be unique in a techDB

DFII provides support from SKILL

```
techId~>refLibs (set/get)
techId~>refLibNames (get, when the graph is bound)
techId~>hasConflict
```

Displaying Technology Graphs with Conflicts



Library TechLib6

Effective [✓ As Defined]

```
TechLib6
  TechLib5
    TechLib4
      TechLib1
    TechLib3
      TechLib1
  TechLib2
    TechLib1
```

Status:

Library TechLib6

[✓ Effective] As Defined

```
TechLib6
  TechLib5
    TechLib4
      TechLib1
    TechLib3
  TechLib2
```

Status:

Setup

“What if I don’t want to use ITDB?”

- No Setup/PDK modification is needed if one does not want take advantage of ITDB
- Just keep using techDBs the same way we have in the past
  - Other design groups can still reference your technology using ITDB even if you don’t use it in any of your personal techDBs
  - Only the techDBs that use ITDB have the refTechLibs construct
  - A techDB is inherently non-ITDB until it references another techDB!

Advantages of ITDB

A flexible modular solution

Minimize duplication of data

- Single point of change

Modularize technology data

- Maintain variants by modularizing definitions
    - Metal stacks are only changes in the metallization libraries
- Shared common data
    - Base level layers
    - Foundry constraints
- Store data with consuming library
    - Sites with Standard Cell library
    - Device specific rules with device (pcell) libraries

Advantages of ITDB

Promotes appropriate ownership of tech data

- Relevant team manages their specific section
  - Device generators
  - Standard cell libs
  - Memories

Control technology requirements for design

- Custom vias for design
- Forbidden layers

Tool setup separate

- lefDefaultRouteSpec

Documentation

Virtuoso Technology Data User Guide

- This is the best documentation on ITDB with examples
- Chapter 1 – Usage and example
- Chapter 4 – Referencing and creating
- Appendix B – Duplicate groups and conflict avoidance
- Appendix C – Good ASCII example

Virtuoso Technology Data: ASCII Files Reference Manual

Design Data Translator's Reference

- LEF/DEF Translators and ITDB

Source Link Best Practices paper:

- Incremental Technology Database Usage and Practices