Lecture 6. Tutorial on Cadence Virtuoso Schematic Editor

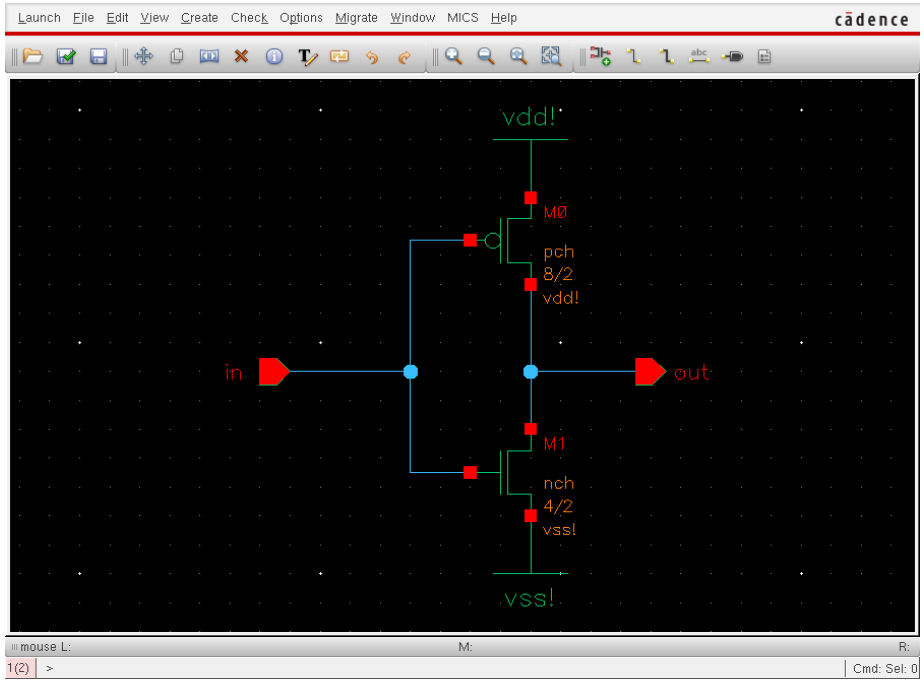
Jaeha Kim
Mixed-Signal IC and System Group (MICS)
Seoul National University
jaeha@ieee.org

Schematic Editor

- Schematic editor (e.g. Cadence Virtuoso) lets you express the circuits graphically
  - More intuitive and less error prone than typing the text-based SPICE netlists

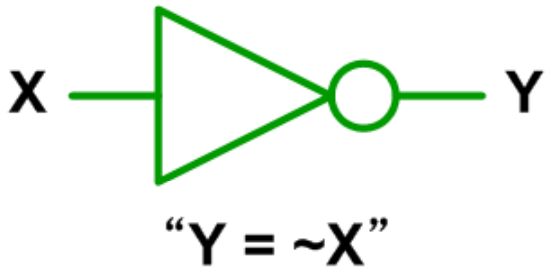
- The SPICE netlists can be generated which is then included in the simulation deck

- Refer to the Virtuoso Tutorial on the website

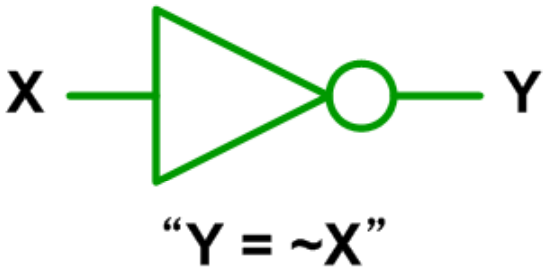


Motivation for Portable Design

- A fabless semiconductor IP company needs to deliver its circuit designs in multiple IC technologies
- For digital designs, it is straightforward
  - Designs are represented in process-independent HDL codes
  - Reusing designs across different technologies is common



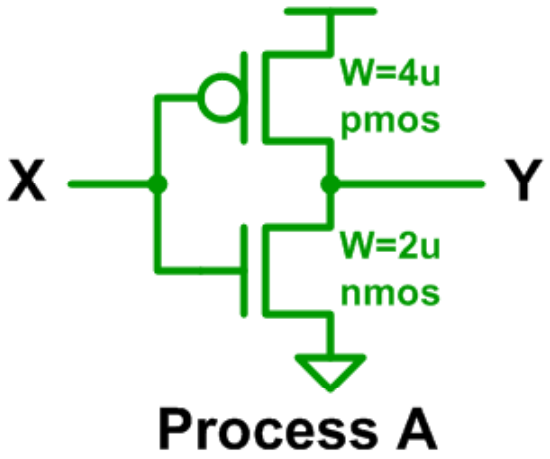
Process A

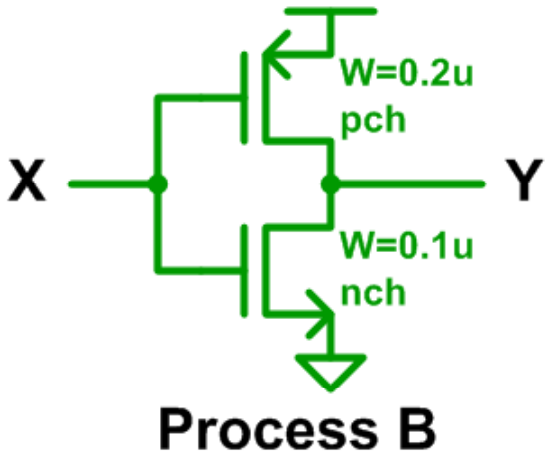


Process B

Motivation (2)

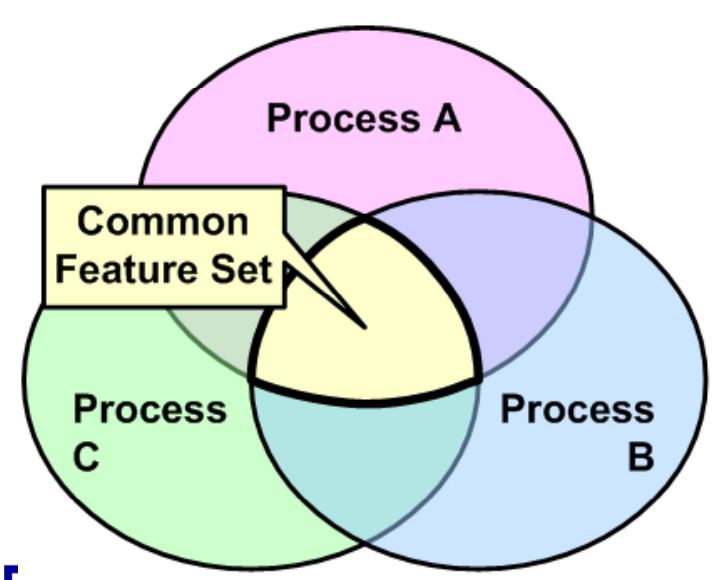
- But for analog designs, process difference is a barrier
  - IC foundries call their transistors with their own names and own parameter sets – even if they are the same NMOS!
  - This difference often leads to incompatible IC design flows
- Analog design reuse still relies on manual effort
  - Even when porting circuit schematics across technologies

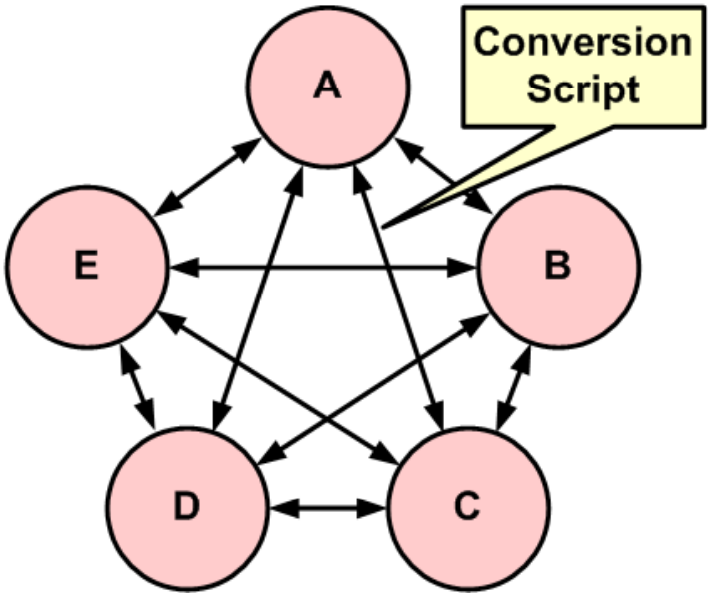




Existing Approaches for Porting Schematics

- Use a standardized symbol library upon agreement
  - The library must be limited to the common feature set
  - No one is happy; can't express fully in a given process
- Use conversion scripts to port between processes
  - CAD managers' nightmare: # scripts required = N·(N-1)

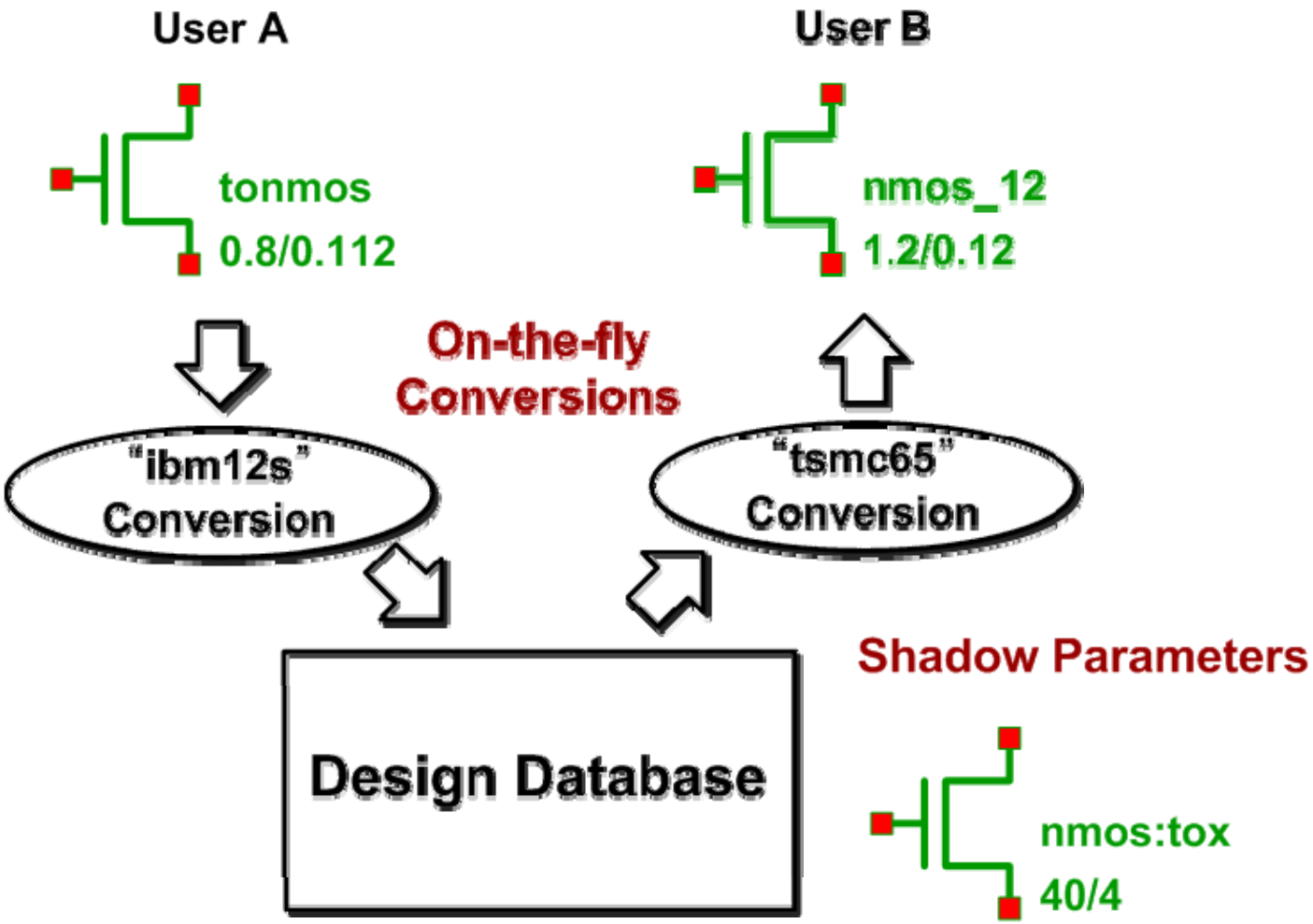




Our Solution: P.I. SCHEMA

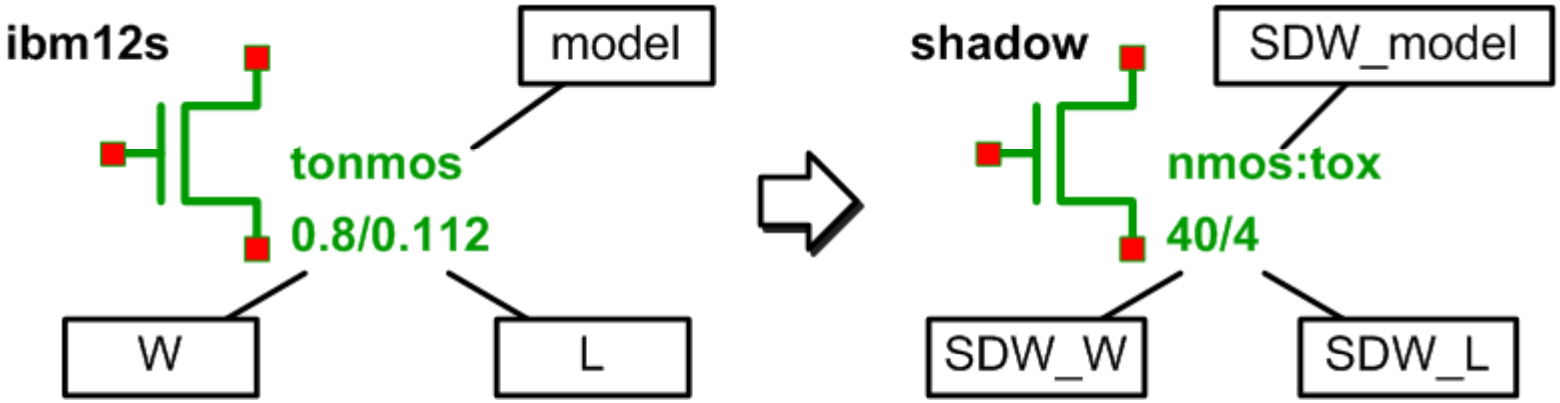
- Process-Independent Schematic Symbol Library
- A symbol library that aims to represent circuit schematics in *all* IC process technologies
  - Schematics can fully express the features of each technology with its tailored parameters and value conventions
  - The same look-and-feel as the dedicated symbol library
- Schematics can be exchanged across technologies *as-is* without explicit conversion steps
  - Copy design entries between different design database
  - Share common circuit blocks from a central library
  - Copy-and-paste part of the circuits from one schematic to another

How P.I. SCHEMA Works



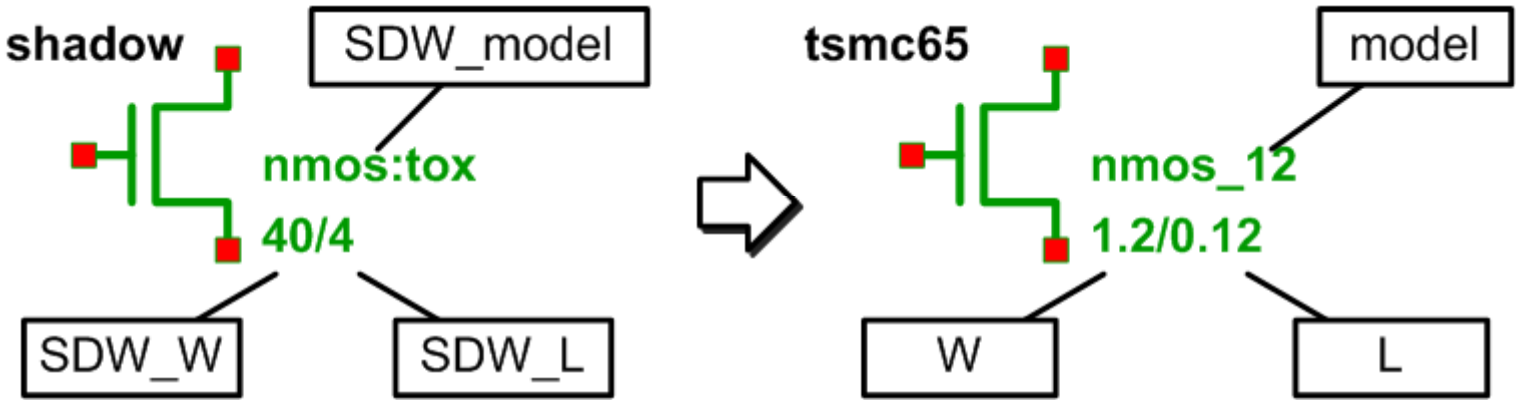
Shadow Parameters

- Store the design intents in process-independent way
  - So they can be mapped to any process technology
  - Hidden from the users – hence called "*shadow parameters*"
- Any design entry gets translated into a process-independent form before being stored in the database
  - The design database is not attached to any particular process



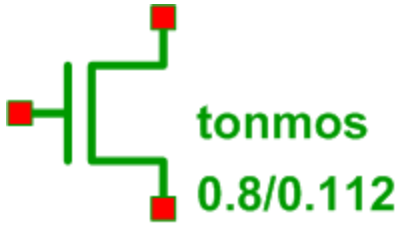
On-the-Fly Conversions

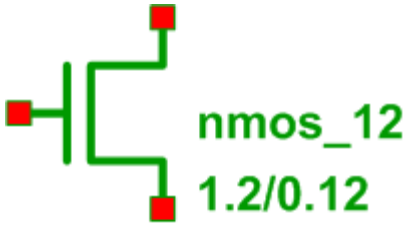
- Design entries in the database *display differently* depending on the choice of the IC technology
  - Converts from the shadow parameter values “on the fly”
  - Users only see the parameters being displayed and *think* that the designs have been converted for the technology
  - But the designs themselves don’t change



Implementation Details

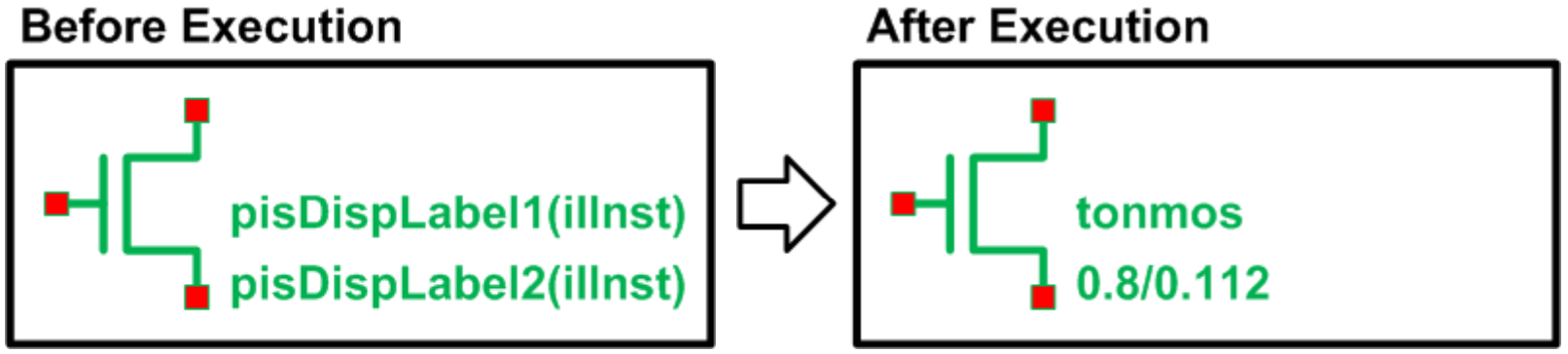
- With shadow parameters and on-the-fly conversions, we can create an *illusion* that each user is using a customized symbol library for their IC technology
  - With a single symbol library (P.I. SCHEMA)
  - With the same design database
- Key is to note that such illusion can be realized with:
  - Symbol labels tailored to the process
  - Parameter editing interface tailored to the process



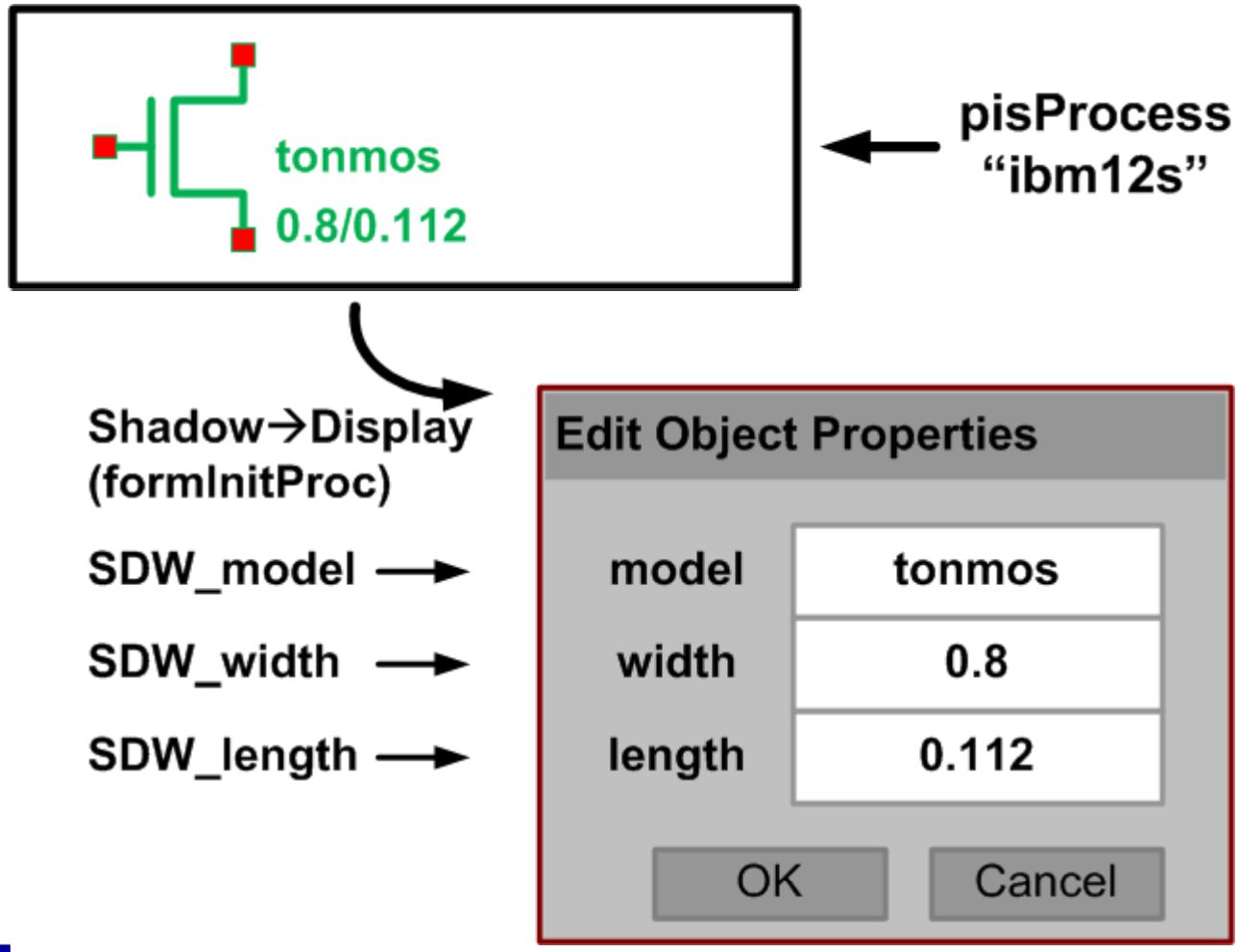


Tailoring Symbol Labels to the Process

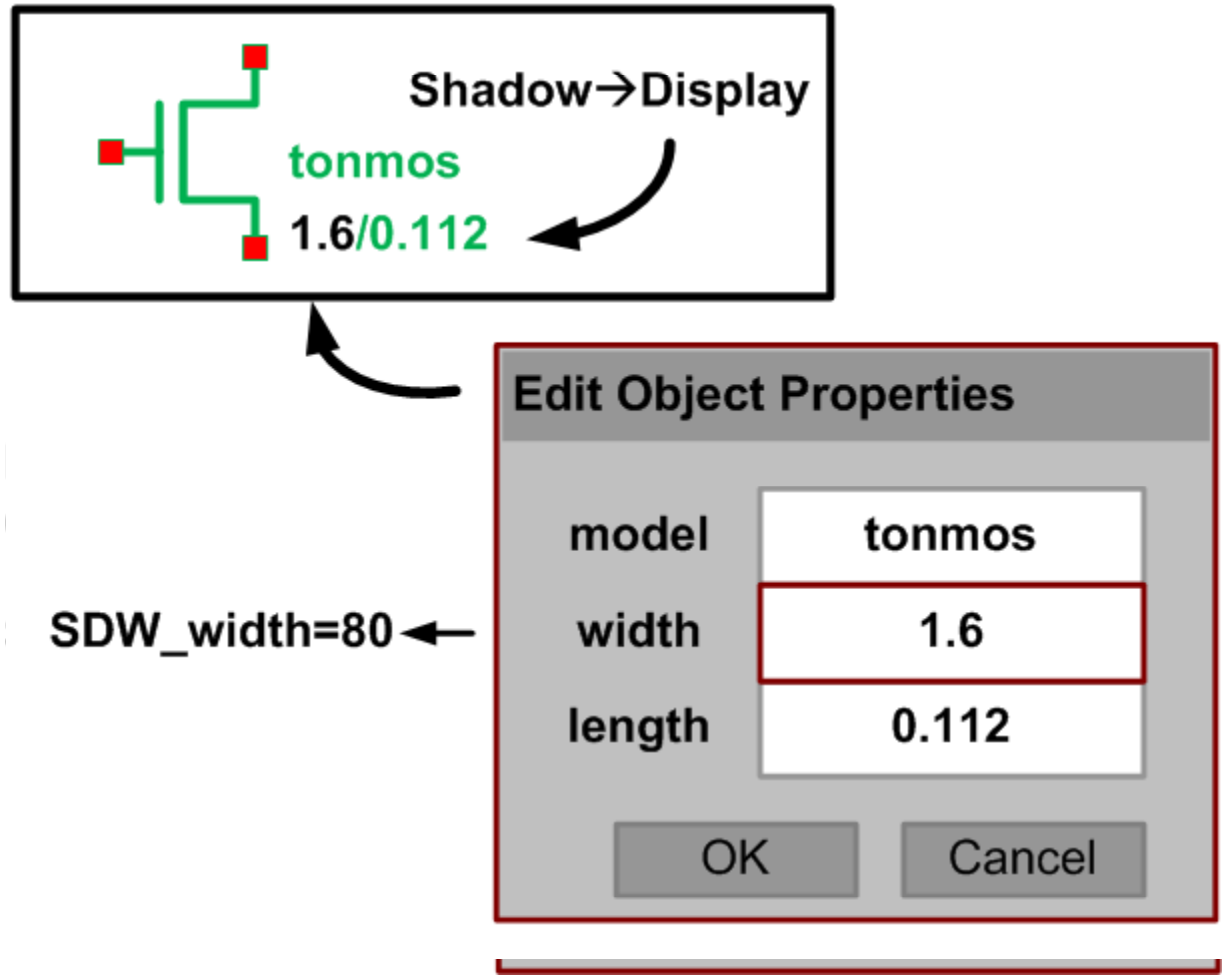
- Use executable labels
  - In Cadence Virtuoso, they are SKILL labels (ilLabel)
  - The “pisDispLabel” functions return process-specific values by translating the shadow parameters based on the process
  - The process technology is specified by a library property named “pisProcess”



Tailoring Parameter Editing Interface



Tailoring Parameter Editing Interface (2)



Technology Configuration File

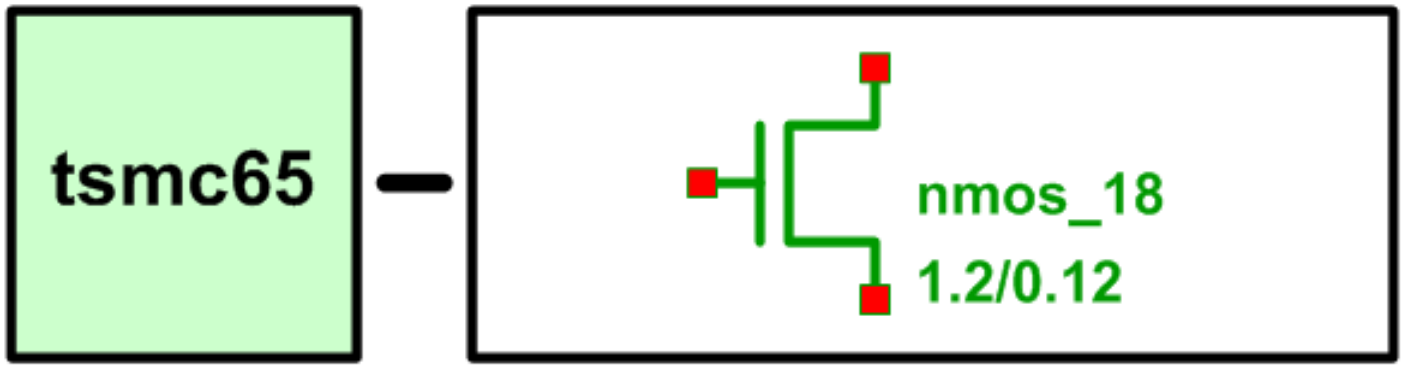
- A configuration file for each process technology defines:
  - Conversion from shadow to display parameters
  - Conversion from display to shadow parameters
  - Symbol labels (pisDispLabel)
  - Parameter editing interface
  - Netlisting format

- These are defined as “virtual functions” that are selected based on the technology choice (pisProcess property)
  - In Cadence, object-oriented programming using SKILL++

Demo 1: Porting Schematics

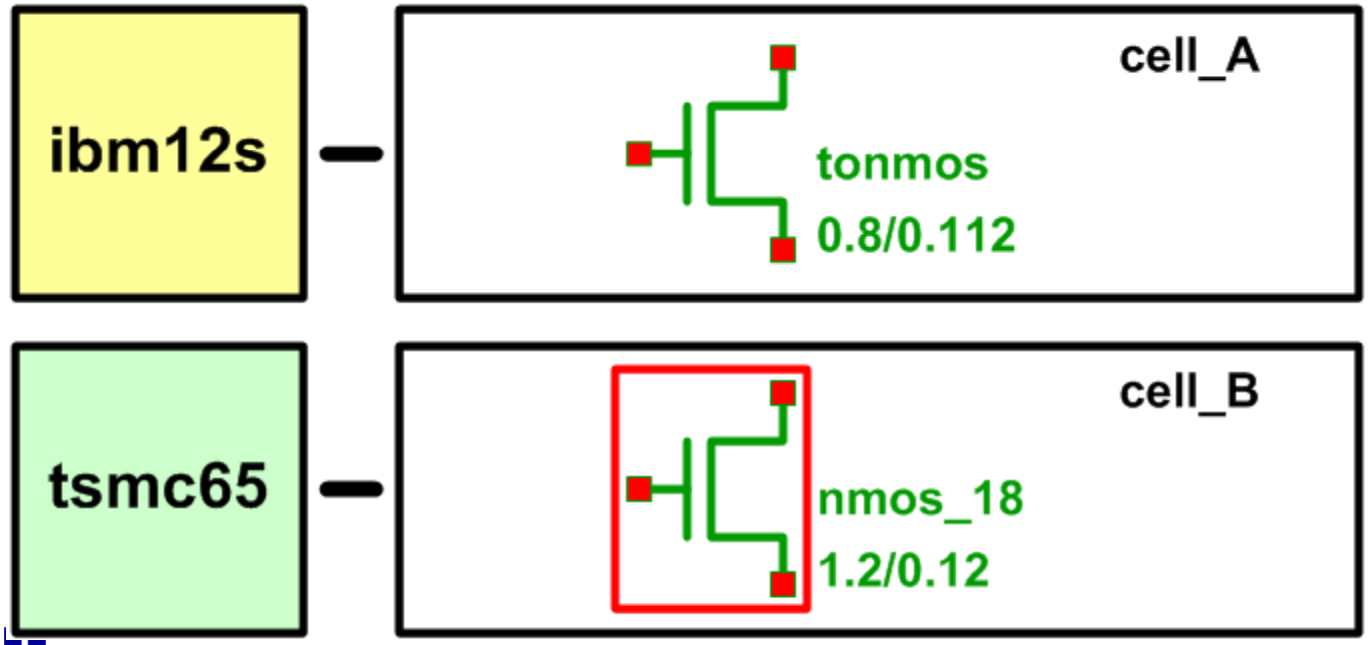
- Simply switch the conversion layer to convert the way the design is displayed
- For the users, it means just updating the “pisProcess” from one process (ibm12s) to another (tsmc65)

pisProcess



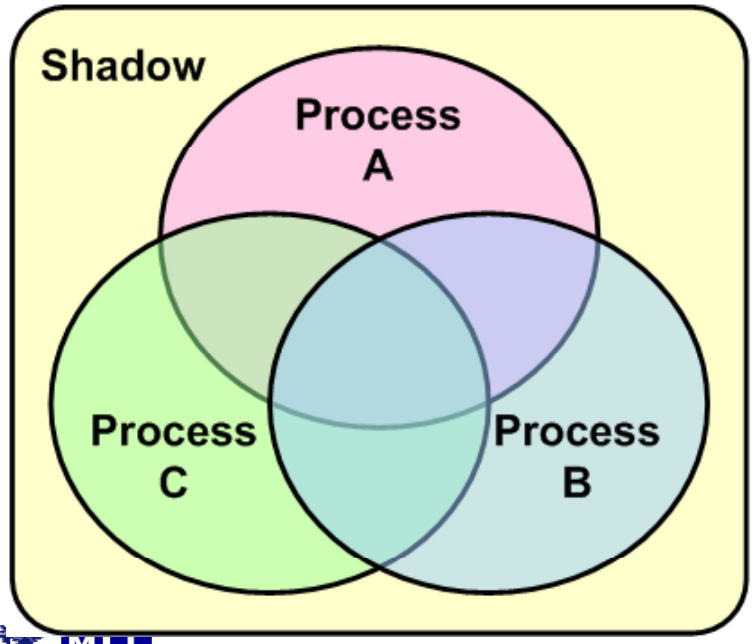
Demo 2: Copy-and-Pasting

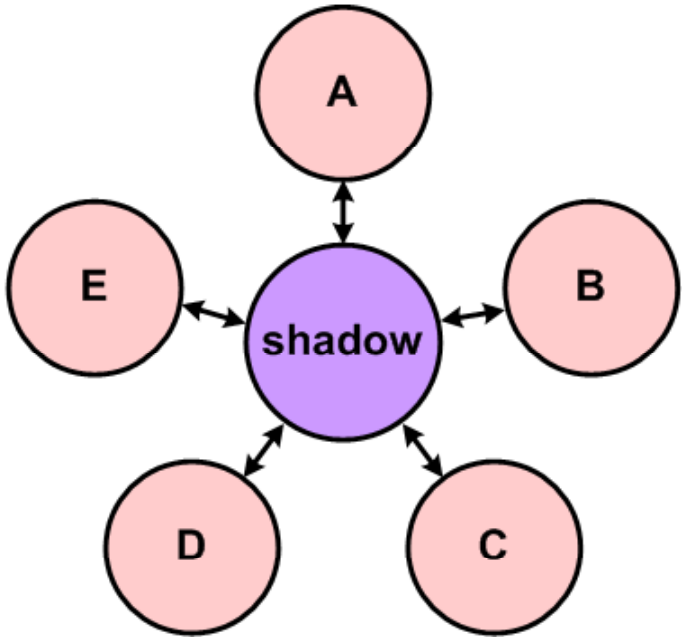
- All design databases use a single, shared symbol library regardless of their IC technologies
- You can freely copy-and-paste part of the circuits between the schematics based on different processes



Comparison with Existing Approaches

- vs. using a standardized symbol library
  - P.I.SCHEMA can fully express all features of any technology
- vs. using conversion scripts
  - CAD manager manages N configurations instead of N·(N-1)





Conclusions

- P.I.SCHEMA provides an abstract representation of analog circuits without sacrificing ease of use
  - Schematics stored in the database are process-independent
  - Schematics displayed to each user are process-specific

- P.I.SCHEMA facilitates share and reuse of analog circuits among diverse design communities
  - Circuit schematics can be freely exchanged regardless of process technology differences
  - IC design flows dependent on symbol library can now converge