



Confidential
Security C

Custom Analog Interoperability: Is It Possible?

Cliff Hou

Director, Product Services Division

TSMC

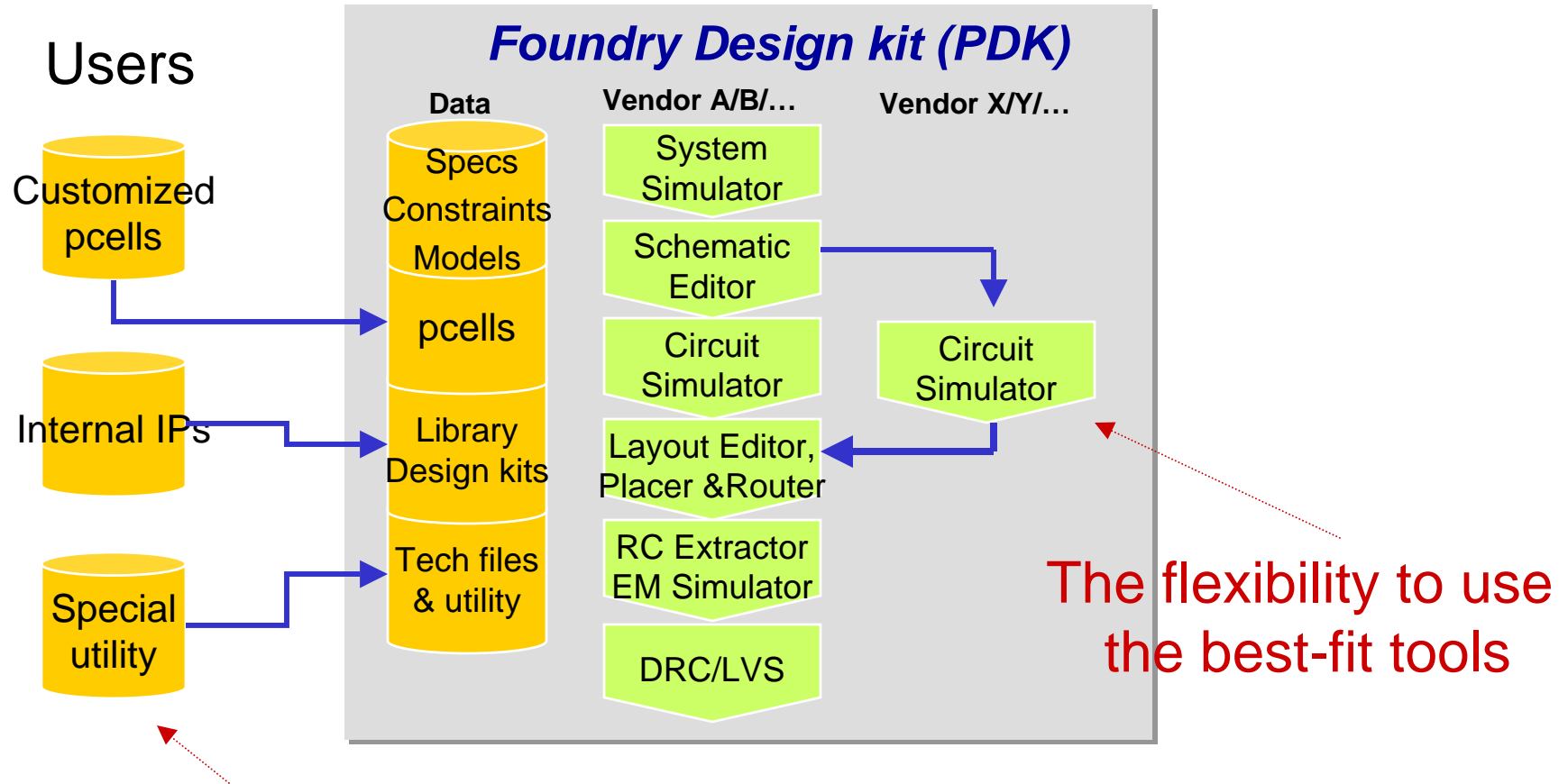
June 6 2007



Designer + EDA + Foundry Joint Collaboration to Resolve Design Challenges

- **With the fast increase of design complexity, challenges, and DFM issues, designer, EDA vendors, and foundry should work together to build an optimized and customized design env. for full custom designers**
 - **Foundry → design env. foundation and tools validation**
 - **EDA tools → design efficiency, accuracy, and new process features modeling**
 - **Designer → design features, applications, and cycle time reduction via optimized and customized design env.**

An Optimized and Customized Custom/Analog Design Env.



The flexibility to fit in customers' specialty and differentiation



Interoperability Issues

- **Proprietary design databases**
- **Non-portable/reusable PDKs**
- **Lack of a unified constraint format to facilitate top-down design methodology**
- **Multiple and incompatible tool interfaces**
- **The foundation of custom design is fragmented and incompatible**
 - **Stifle custom digital, analog, mixed-signal/RF innovation**
 - **Not too different from digital design decades ago**



The End Results

- **Poor productivity**
 - Both for the PDK developer as well as the designer
 - Limits reuse and portability
 - Delays new technology and tool adoption
- **Redundancy**
 - Difficult to maintain and update
 - High support cost
- **Poorer quality libraries and design**



Foundry Perspective on PDK

- **PDKs provide the foundation for quality design and facilitate a higher success rate on first silicon**
- **Interoperable PDKs facilitate foundry to update and add device and technology information into design kits promptly and efficiently**
- **Open-standard PDKs allow easy plug-in of different vendor's tools to enable foundry for more flexible flow support and improve quality**
- **High quality PDKs shorten the overall design cycle for fabless customers**

What MS/RF PDK Contains?

- Configurable passive & active components
- Scalable active devices
- RF non-linear device models
- Symbols, models, layout generators, DRC/LVS rules
- Temperature variations, corners

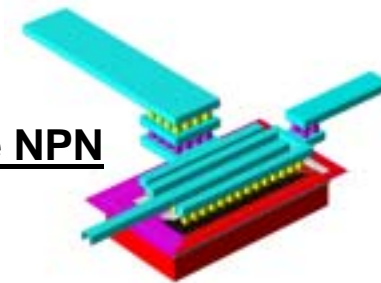
• Active devices

- NPN transistors for digital/high performance/high breakdown
- RF and standard N- & P-channel MOSFETS
- Junction varactor
- MOS varactor
- N+ P-well diode/P+ N-well diode

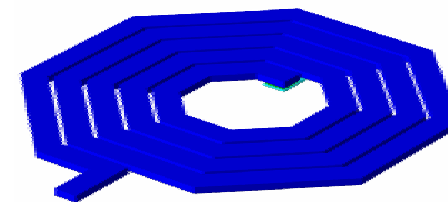
• Passive devices

- Metal-insulator-metal capacitor
- Scalable octagonal inductor
- N-well resistor with multiplicity for series/parallel connection
- Polysilicon resistors with multiplicity for series connection
- Diffusion resistors with multiplicity for series connection
- ...

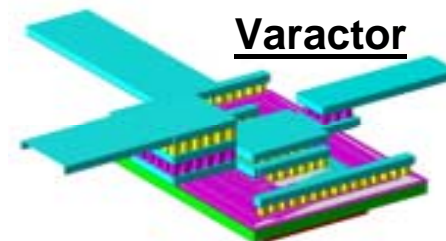
SiGe NPN



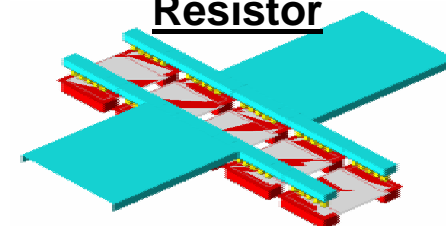
Inductor



Varactor



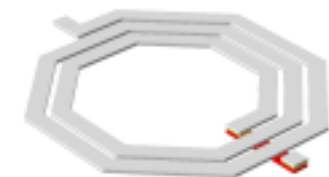
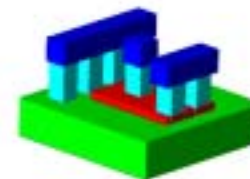
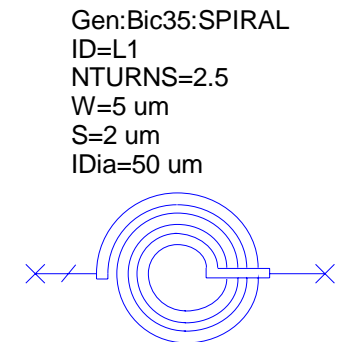
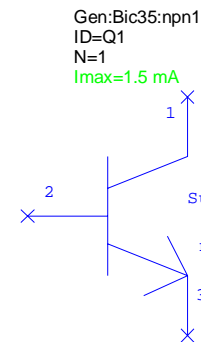
Resistor



PDK Interoperability Requirements

- **Design constraints**
 - **Electrical**
 - **Physical**
 - **Manufacturing**
- **Simulation Models**
 - HSPICE, Spectre, Eldo, others
- **Electrical**
 - Schematic symbol
 - **Naming conventions**
 - **Electrical/physical parameters**
- **Physical**
 - **Parameterized cells (Pcells)**
 - DRC/LVS/LPE/DFM rules
 - GDS layer map

Symmetry
 Distance between inductive elements
 Placement of dummy elements
 Placement of pass capacitors
 P&R of guard band
 Common centroid
 Placement of parallel Tr.s
 Zig zag placement of Rs
 Enclosed R, C and MOS
 Isolation of N-region, Additional substrate for MD





Is unified PDK possible?

- **Interoperable Pcells**
 - Possible with current OA capabilities and the true open collaboration of designers, EDA vendors, and foundries.
 - Potential boost in development productivity with better development environments (Python, Tcl, etc.)
- **Certain standards still required**
 - To specify and implement PDK parameters
 - To specify design constraints
- **OpenAccess is a good first step**
 - A unified data storage mechanism
 - Good physical data model & APIs
 - Need to be expanded to cover:
 - ◆ Electrical/physical parameters
 - ◆ Parameter relationships
 - ◆ Design constraints