



Persistent PCells

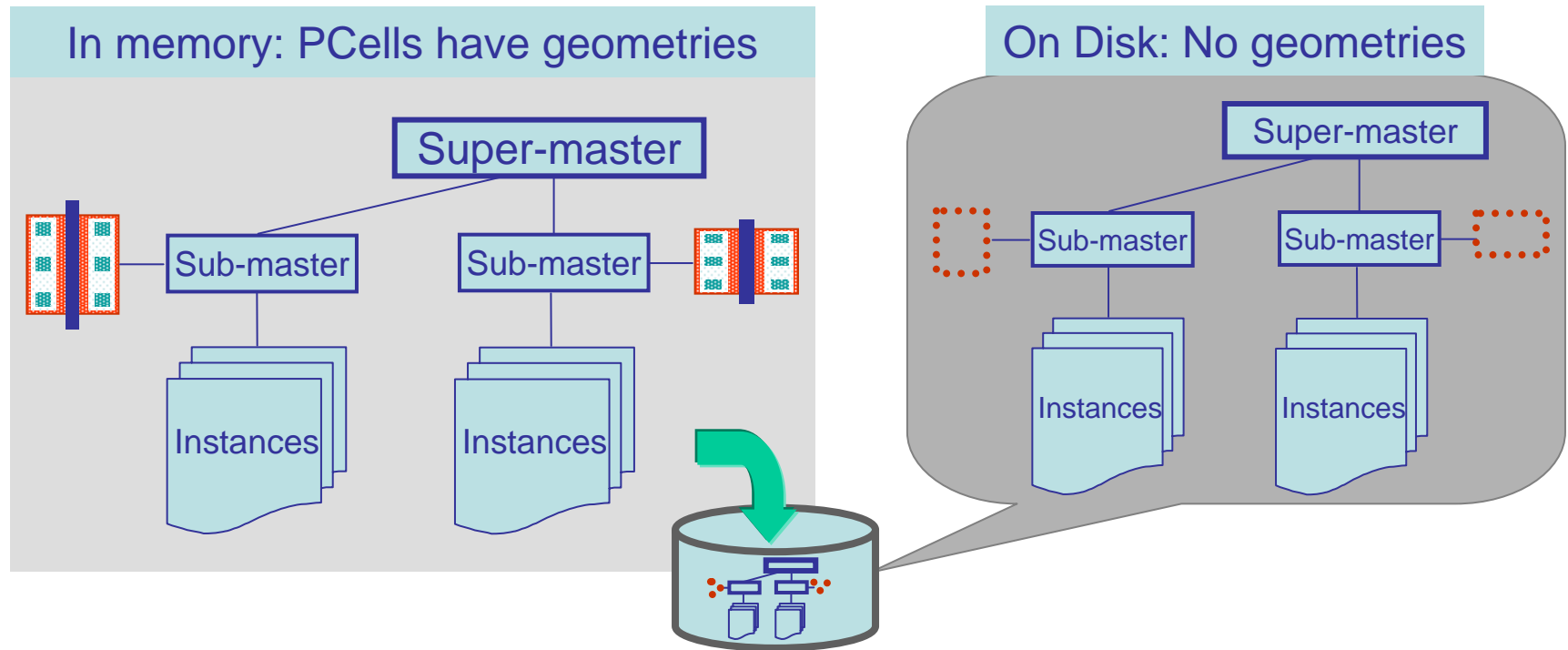
Ed Petrus
COO & VP of Engineering

November 2006

Customer Expectations

- ⊕ Speed up load time for designs with lots of PCells
- ⊕ “Freeze” sets of PCell instances during project life cycle
- ⊕ Make it possible to access designs containing PCells in tools from multiple vendors
 - OpenAccess based tools, of course

Why are PCells not Persistent?



- ❖ PCells in a design must be “re-evaluated” each time the design is read from disk...
- ❖ ...so geometries can be re-generated and attached to the sub-master structures
- ❖ That is how it has always been in Cadence’s CIC data models, hence in OpenAccess

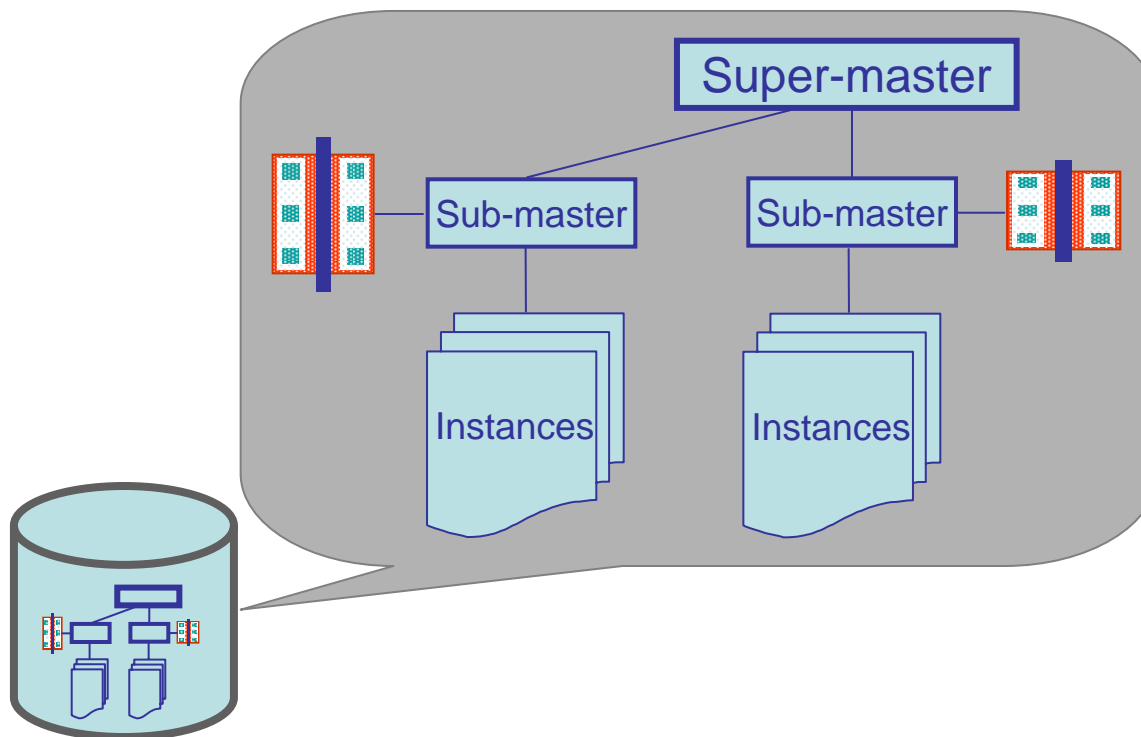
It's About the Sub-masters ...

⊕ Synonymous:

Persistent PCells ⇔ Persistent Sub-Masters

The Customer is asking for ..

- ⊕ The ability to *cache* PCell geometries, to disk that is
- ⊕ Make the sub-masters “persist” beyond a session in memory
- ⊕ Re-evaluate PCells on conditions specified by the user

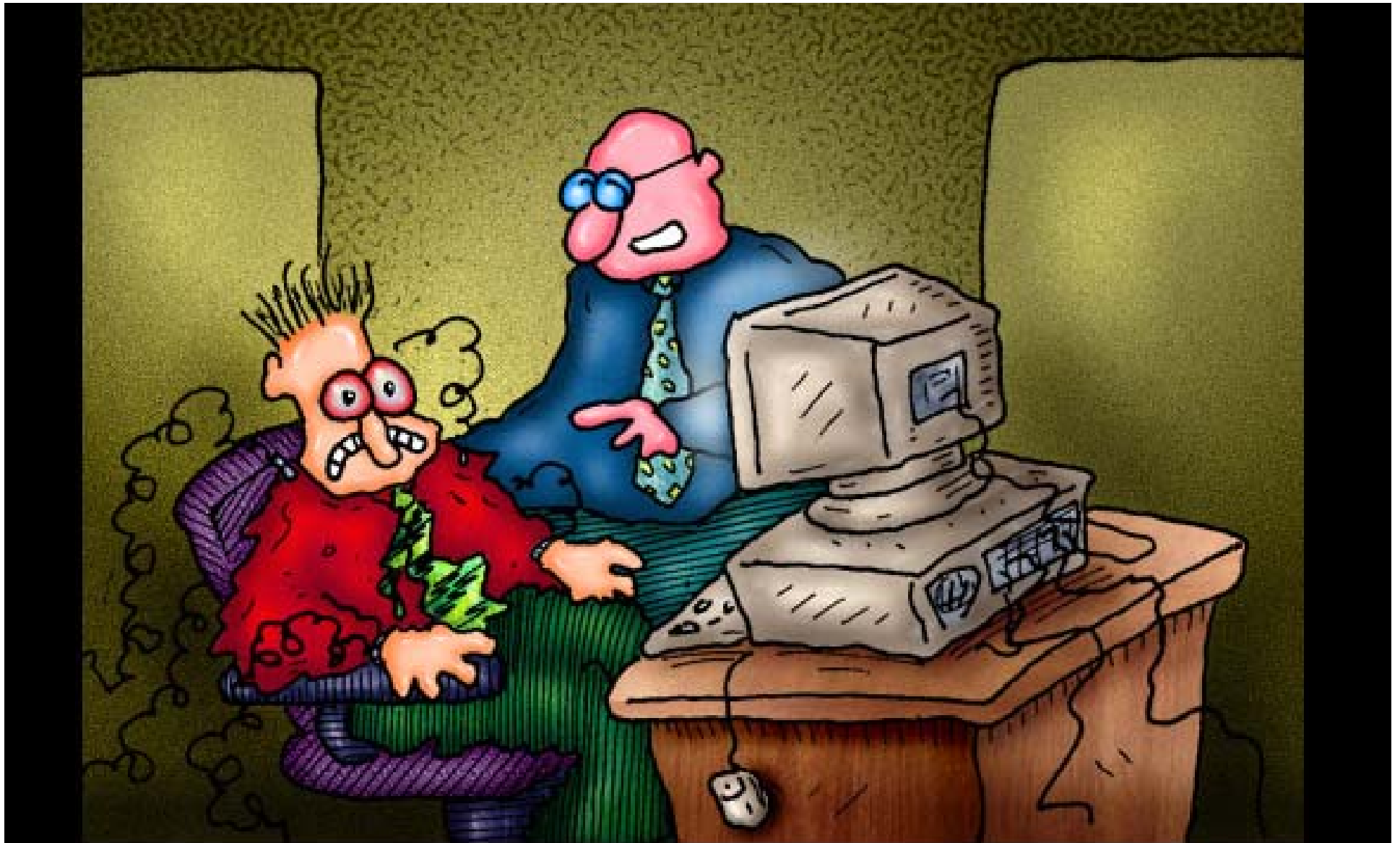


Status of Persistent PCells in OpenAccess

- ⊕ This issue has been an issue for > 2 years
- ⊕ SI2 study - “PCell Study Group” - published recommendations in Feb-2006
 - Participants: ARM, Freescale, LSI Logic, National, Philips, Tektronix, SI2
 - Top recommendation: “support Persistent PCells on OpenAccess”
 - See *Si2 PCell Study Group Report*, (Authors: English & DasGupta)
- ⊕ Requirements specification not available
- ⊕ As of late, the issue is being pushed into the realm of the products
 - That is, not likely to be addressed by the OpenAccess reference implementation
 - It is now a “business” issue

Motivation?

- ⊕ Cadence is not motivated to solve the problem
- ⊕ Other members of the OpenAccess coalition are motivated to solve the problem
- ⊕ Unfortunately, “Persistent PCells” in OpenAccess has become a politically charged issue



"...And it gives you a mild electric shock if you write anything about Persistent PCells!"

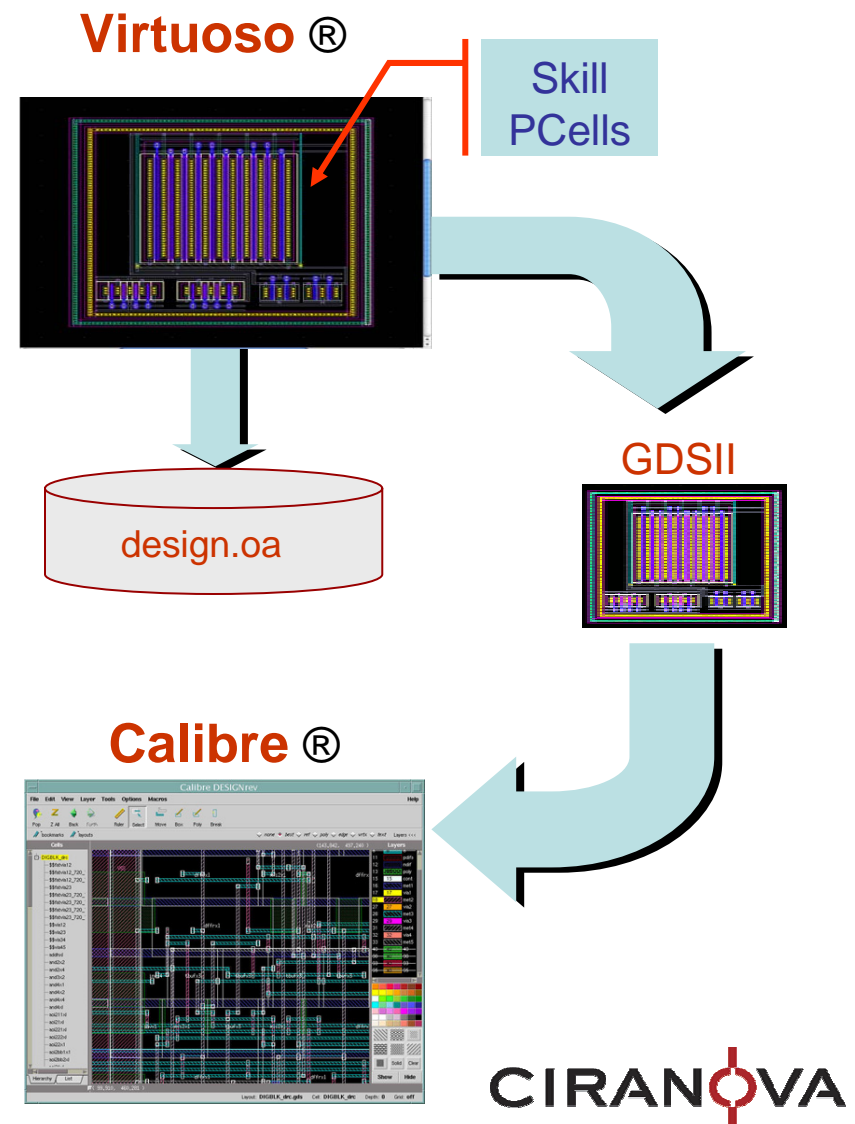
A Perspective on the Issue

- ⊕ The “Open” in OpenAccess
 - Is complete openness of design data inevitable?
 - Fair play: dominant EDA tools should not monopolize design data
- ⊕ Whose IP is in and around PCells?
 - Device geometries belong to the customer
 - The designs belong to the customer
- ⊕ A mechanism for caching PCells in OpenAccess should be available to all tools
 - Both to read AND to generate sub-master cache



Openness of Design Data

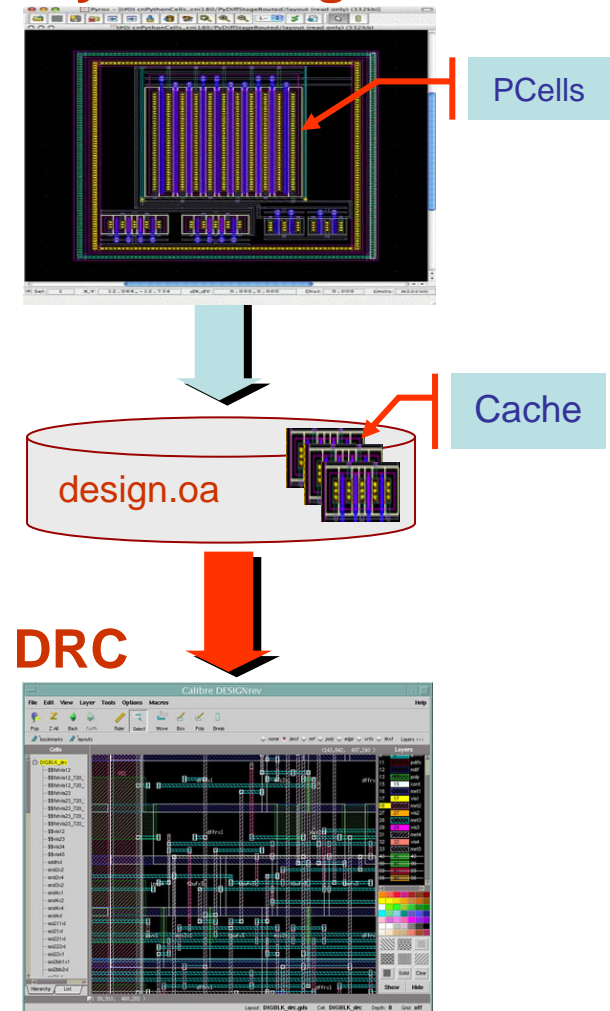
- ❖ CIC 6.1 is in Production
- ❖ Calibre on OA is in Beta
- ❖ If a design uses Skill PCells, then the design data must go through GDSII to complete this flow
- ❖ Calibre can't read the design through OpenAccess
 - Isn't able to regenerate the Sub-master geometries



The Requirements

- ⊕ Enable sub-master caching
 - Persistent on disk
- ⊕ The mechanism should work for all OpenAccess PCells
- ⊕ Allow for user policies to determine:
 - The location for the cache
 - The conditions for invalidating cache
 - Fine control over the cache
- ⊕ Allow design management systems to work with the cache
- ⊕ Allow tools other than the creator of the cache to access the cache, for example:
 - If a layout tool saves a design along with a cache of sub-masters ...
 - Then a DRC engine can read the design and access sub-masters from the cache
- ⊕ More?

Layout Design



The Ciranova Approach

- ⊕ The original goal was to cache Ciranova™ PyCells™ for speed
 - Interoperability of PyCells is a non-issue
 - PyCell evaluator plug-in is freely available
- ⊕ The mechanism to cache PyCells turned out to be more broadly applicable to all OpenAccess PCells
 - Regardless of their source: SKILL, PyCells, Tcl, C++
- ⊕ It also became apparent that the cached sub-masters can be created and read by multiples of EDA tools from different vendors
- ⊕ Works entirely through OpenAccess
 - No tool intervention necessary

What Can It Do – Speed up

- ⊕ 10X or more faster load time for designs with OpenAccess PCells
 - For example, a design containing 1500 unique sub-masters (Skill PCells) takes ~3 seconds to load with caching vs ~35 seconds without caching*
 - The more complex the PCells, the greater the number of sub-masters, the greater the speedup

* Linux/Intel, 2.0 GHz, 1GB RAM, 130nm PMOS PCells, 30MB cache file

What Can it Do - Interoperability

- ⊕ Cache SKILL PCells from Virtuoso[®] and read the designs into OpenAccess based tools
 - Calibre[®]
 - Laker[™]
 - RDE[™] (SiliconNavigator[™])
 - AWR[®] Analog Office[®]
 - More ..
- ⊕ Works as of the production release of CIC 6.1 along with beta releases of Calibre, Laker and RDE

Availability

- ⊕ Available now for beta testing
- ⊕ Formal release in December 2006 *
- ⊕ Available standalone as PCell Xtreme™ to all users of OpenAccess tools
- ⊕ OR as part of a support package for PyCell Studio™

* Not part of the free download

Summary

- ⊕ Ciranova first talked about Interoperable PCells in April 2005
- ⊕ Launched the freely downloadable *PyCell Studio* in January 2006
 - Introduced protocols for stretch handles and auto-abutment
 - PyCells tested in Virtuoso, Calibre, Laker, RDE, AWR[®] Analog Office[®], and more
 - 100s of downloads
- ⊕ Sub-master caching completes the story
 - Please help us validate & enhance the functionality
 - Let's make this a purely technical issue and work to put it to rest
- ⊕ **Please come by and see the demo later this evening**