how big can you dream?™

cadence

Speac/Fleur Workshop, Grenoble:

VCC and its value for SpeAC

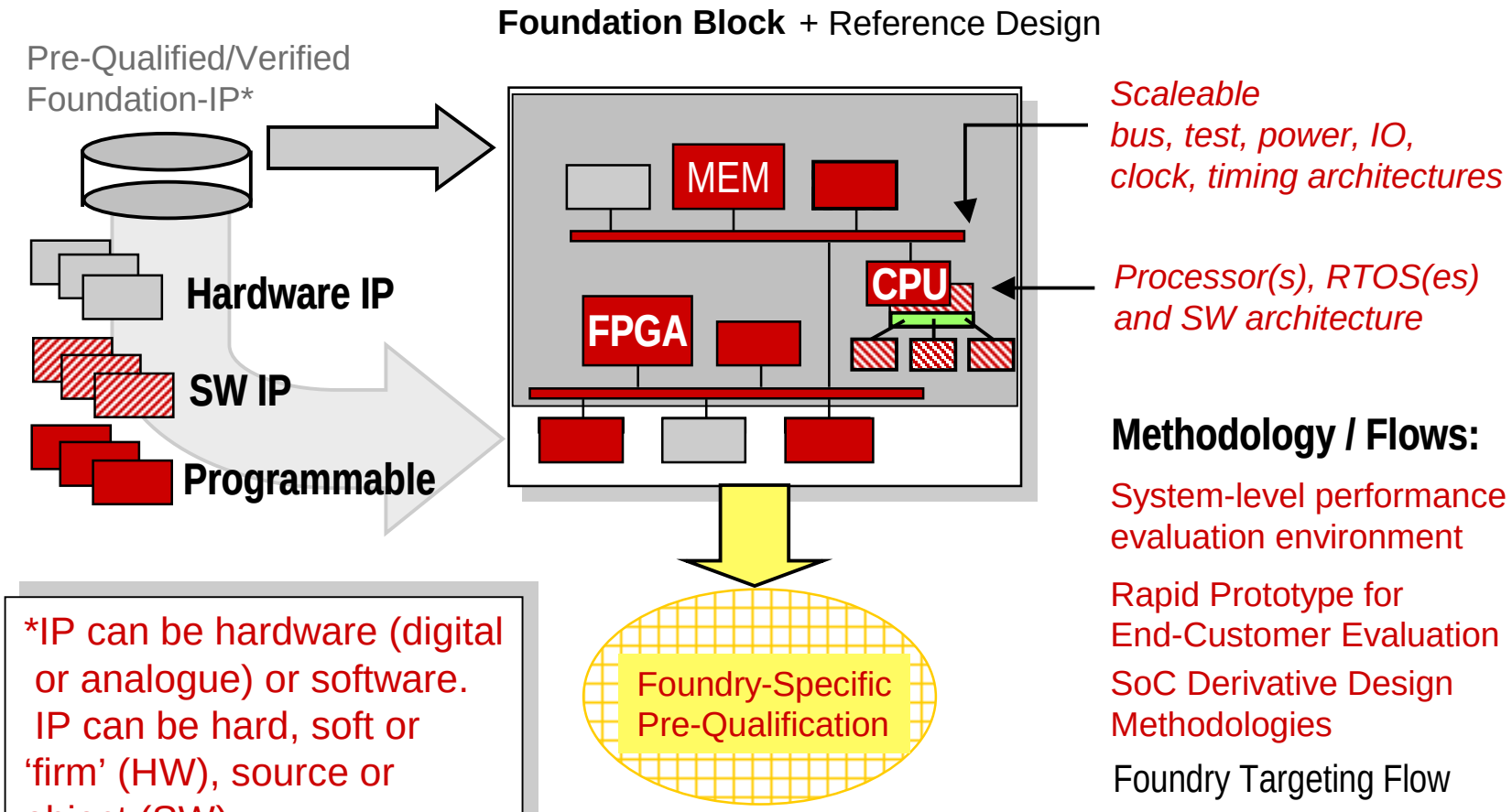
Grant Martin and Fred Husmann

Grenoble, January 24, 2002, 14:45-15:15

Outline

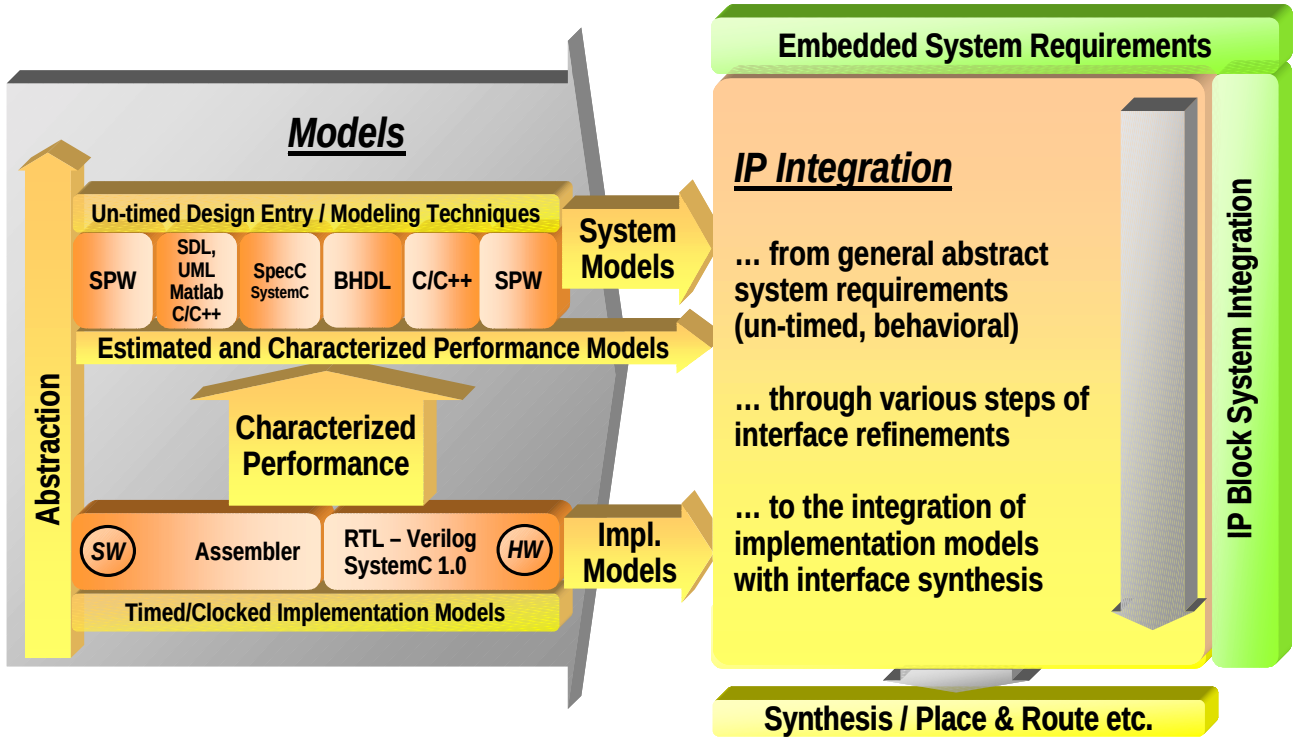
- VCC and Speac
  - The Plan
  - Results So Far
- Future Plans for VCC

Cadence planned contribution to SpeAC
The Platform-based Design Concept



2. System Specification: Plans
2.1 Multi-domain system specification

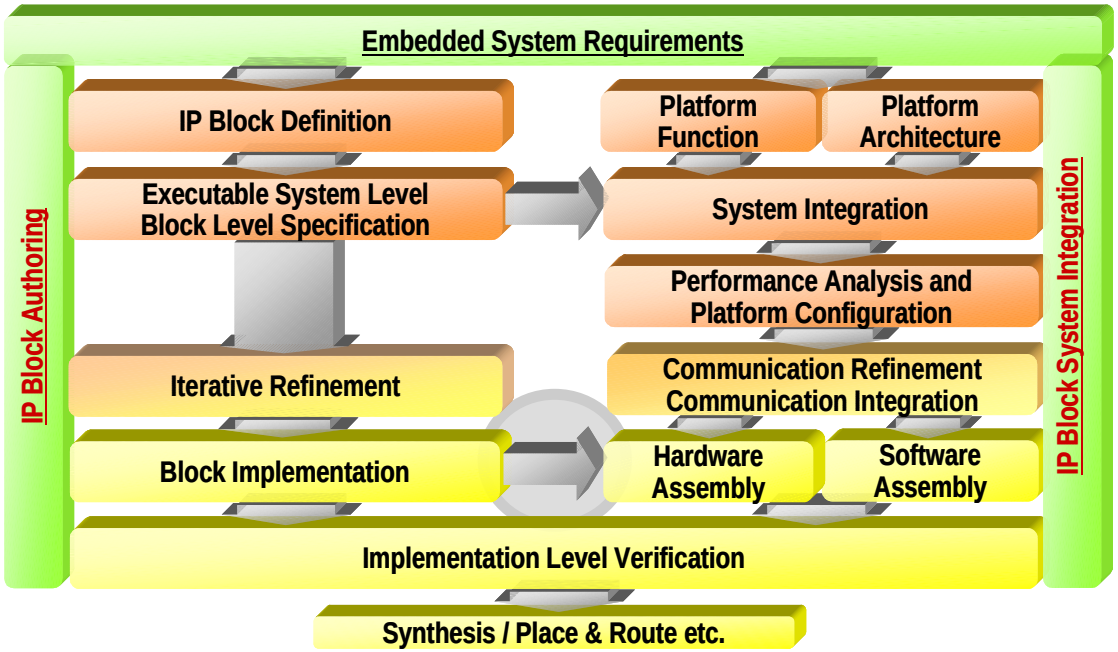
- Integration of functional modeling algorithms in a functional full system verification



3. Algorithm/Architecture Codesign: Plans

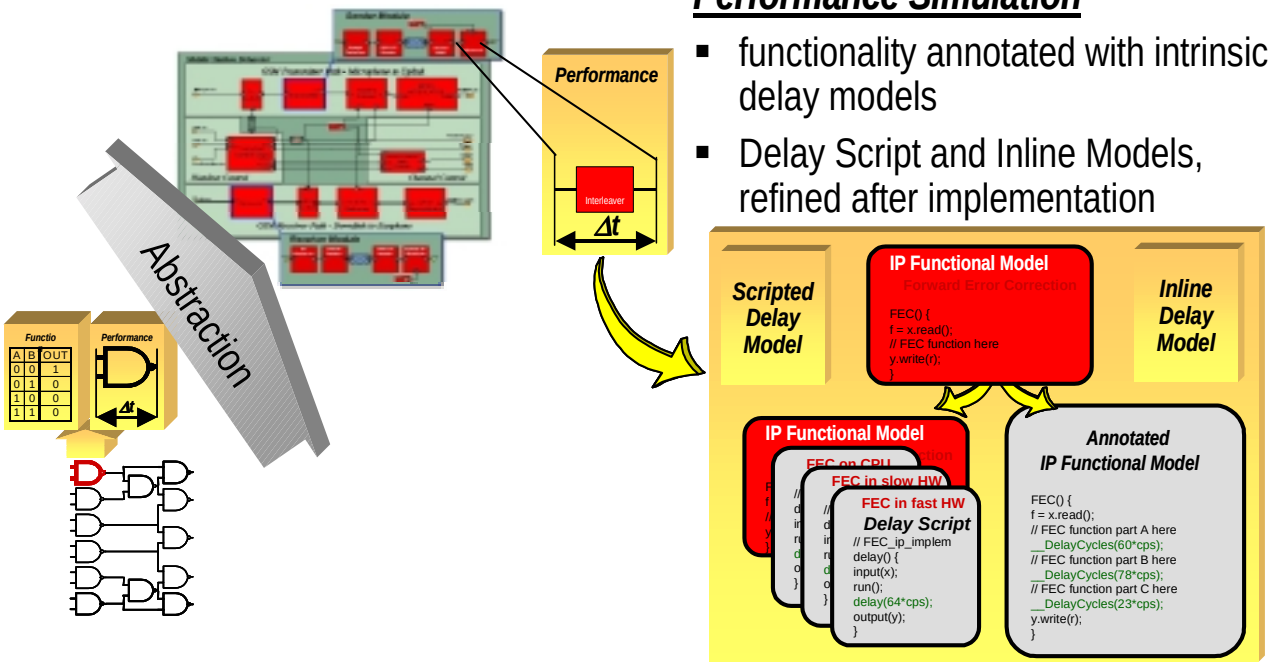
3.1 Mapping functional spec to a macro architecture

- Support of Platform based design flows according to the Y-chart model



3. Algorithm/Architecture Codesign: Plans
3.2 Architecture Evaluation

- Investigation on estimation technologies for functional descriptions based on architectural models



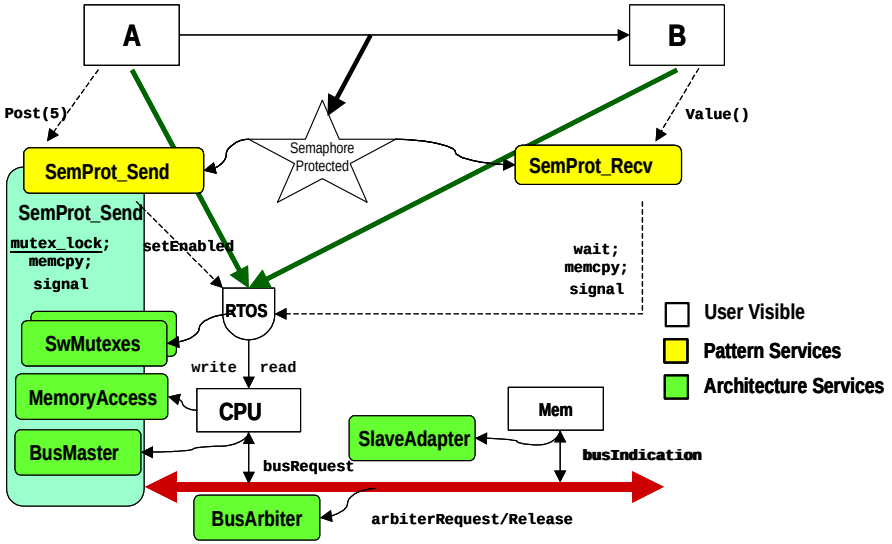
Performance Simulation

- functionality annotated with intrinsic delay models
- Delay Script and Inline Models, refined after implementation

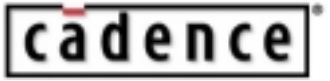
4. HW/SW-Codesign and Communication Layers: Plans

4.1 Communication & macro architecture refinement

- Co-Design methodology support for target architectures and prototypes
  - Research needs to be conducted on
    - What information optimization algorithms will need
    - Identify/Refine an API to give access to analysis results and control parameters for new simulation runs



2.1 Multi-domain system specification: Results to Date



- Investigation of C-based languages (Cadence):
  - Cadence has monitored or been an active participant in various system level languages:
    - SystemC – very active especially regarding 2.0 and beyond
    - Accellera: Semantics, C/C++, Rosetta – active/monitoring
    - SpecC –monitoring
    - UML – analysis of requirements for embedded systems (Embedded UML and UML Platform) and auditing OMG developments
  - We have chosen to focus most of our efforts on SystemC as we believe it is the natural C/C++ choice for standardisation
- Feasibility studies, which is best suited for VCC (Cadence)
  - SystemC is important for VCC Futures
  - Details will be available over the next several months

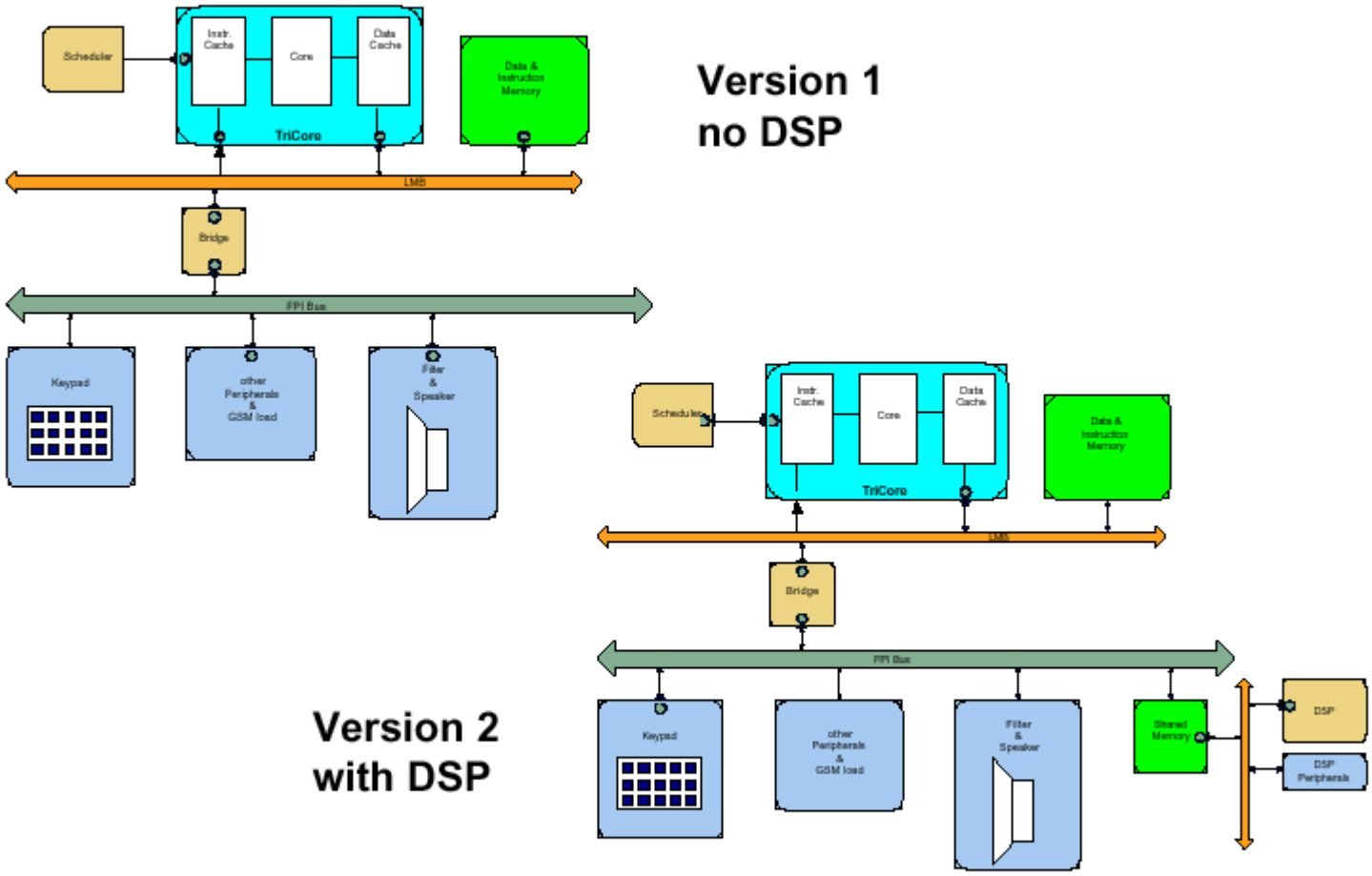
Work Package 3 Results to Date

3.1 Mapping functional spec to a macro architecture

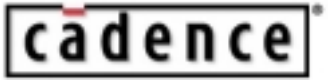
3.2 Architecture evaluation

- **3.1b Estimation of DSP Resources**
  - Analysis of possible VCC VPM enhancements for DSP estimation improvement carried out
  - Concepts:
    1. Statistical Method (as described in Date 2001 paper)
    2. Statically profile target SW code using target compiler/processor and back-annotate delay equations into VCC database manually or automatically (via the VCC Database API – 4.1)
    3. Integration of DSP Instruction Set Simulators
  - Methods 1, 2 and 3 all possible and in use today (method 3 proved in concept with an ARM ISS integration which will be released in VCC 2.2 Spring 2002)
  - Automating method 2 requires use of new API
  - Method 3 requires acquisition of appropriate cycle-accurate DSP ISS, and integration into VCC
- **3.2: VCC Workshop held Grenoble January 24, 2002**
- **Infineon:**
  - "Investigations of a virtual platform for architecture description ongoing
  - Virtual Component Codesign setup aimed for verification of functional as well as for performance constraints estabilished.
  - Reference Example overleaf (VCC model of Tricore subsystem)

VCC model of a TriCore subsystem

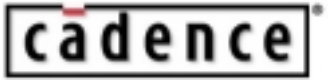


4.1 Communication and macro architecture refinement: Results to Date



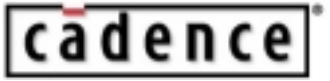
- An API has been identified and developed for VCC (the VCC Database API) and will be provided to FZI with documentation
- VCC Database API:
  - Programming interface to VCC environment
  - Consisten object-oriented functional access to all design objects, both functional and architectural
  - Java and Tcl implementations available
  - Allows one to develop customised VCC capabilities via programs and scripts: e.g.,
    - Creating VCC data from an external database
    - Extracting data from the VCC database for supporting external tools
    - Automated simulation runs
- Speac requirements were incorporated in the definition of the API

4.1 Example excerpt from API documentation



- **Class VccCommLinkEnum**
- java.lang.Object | +--COM.cadence.vcc.editor.diagram.VccCommLinkEnum
- public class VccCommLinkEnum
- extends java.lang.Object
- This object lists objects contained in a diagram. The sequence of the returned objects is not guaranteed to be the same from one complete traversal to the next.
- See Also:
  - VccMapping.EnumCommLinks
- Field Summaryprotected longnativeHandle_
  Protected data managed by this class.
- Constructor Summaryprotected VccCommLinkEnum()
  Object cannot be directly created.
- protected VccCommLinkEnum(long nativeHandle)
  Object cannot be directly created.
- Method Summaryprotected voidfinalize()
- VccCommLinkFindByName(java.lang.String name)
  Find the communication link with the specified name.
- VccCommLinkNext()
  Returns the next communication link.
- voidReset()
  Rewinds traversal to the beginning of the list.
- .....

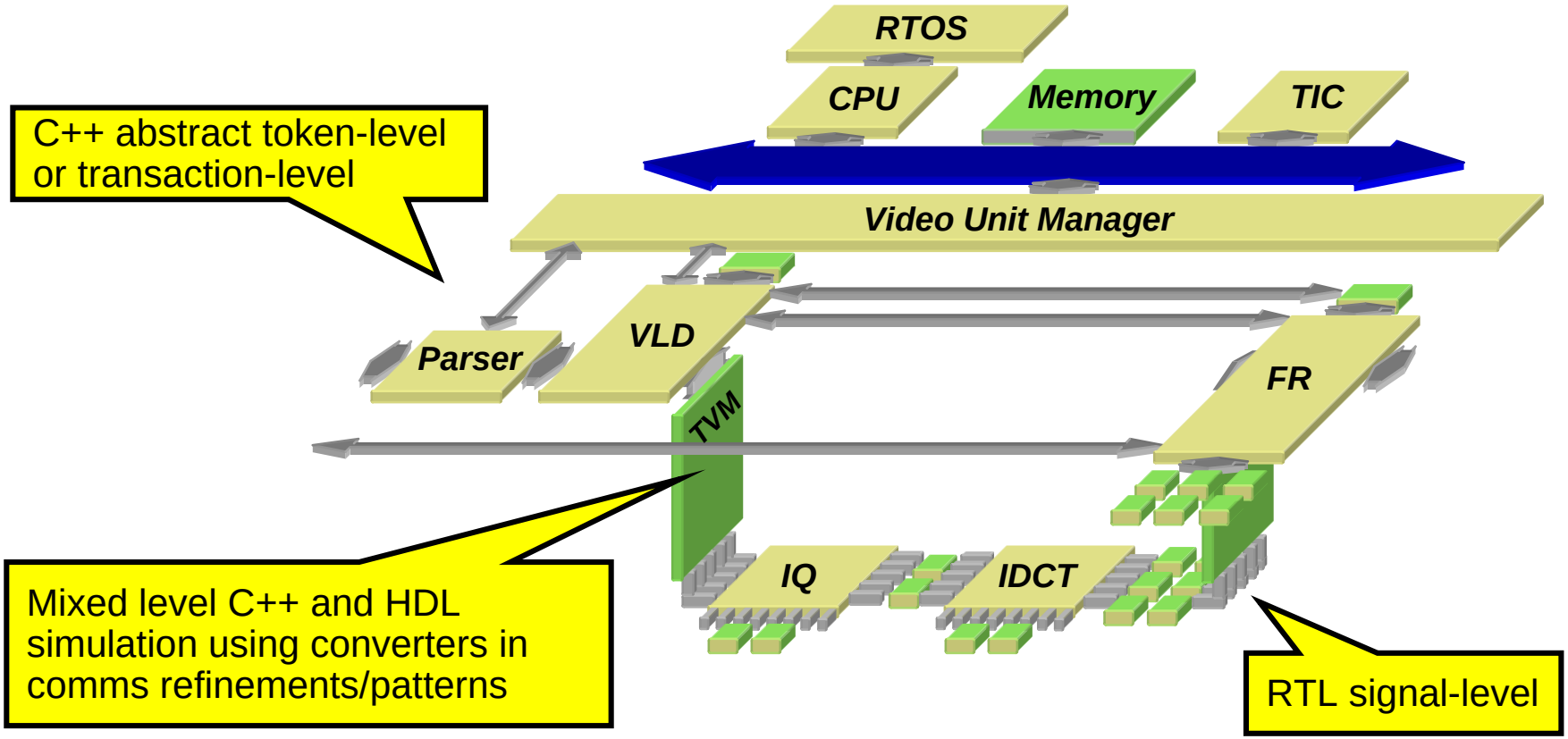
Future Plans: VCC



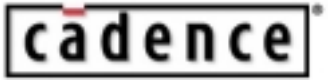
- VCC and SystemC:
  - "SystemC is important for Cadence's System Design and Verification Tools and their roadmap. We are continuing to plan the evolution of our SDV and SFV tools, including VCC, to incorporate SystemC as appropriate. Details are not yet available"
- VCC recent releases:
  - 2.2: Focus on Verification

VCC 2.2
… focus on verification

Target: HW Design and HW Verification Engineers
What is it: C++ based functional simulation at the channel and token level supporting mixed abstraction level simulation (token, transaction and signals) with NC-Sim



Future Plans: Longer-term VCC



- Evolving SystemC support
- completion of the various DSP SW estimation techniques
- completion of ISS integration kit
  - Allows users to incorporate ISSs in VCC (rather than Cadence R&D or services)
  - Planned release in VCC 3.0 timeframe (estimate: late 2002/early 2003)