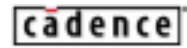


how big can you dream?™



The Future of High-Level Modeling and System-Level Design

Some Scenarios

Grant Martin

Fellow, Cadence Labs

Electronic Design Processes Workshop, Monterey, April 21-23, 2002

1 CADENCE DESIGN SYSTEMS, INC.

Outline



- Six Scenarios for System Level Design
- What is the likely future?

2

The Cult Scenario



cadence



- High Priests, Gurus, and their Acolytes
- Obscure Mathematics and Jargon
- Impenetrable to ordinary designers
- No flow
 - Models have no reuse at lower levels of abstraction
- There is no mainstream

3

The Niche Scenario



- Best practices today
- Dataflow: from algorithm to implementation
- Finite State Machine capture, simulation, and generation of synthesisable code
- Most of the well-understood niches have been explored

4

The Platform-Based Design Scenario

cadence

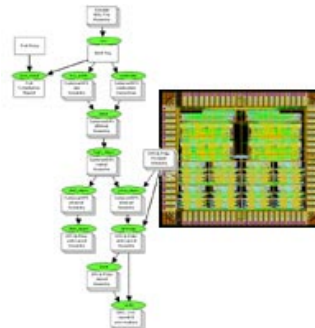
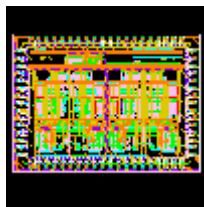


- Co-ordinated family of HW-SW architectures promoting high levels of reuse of HW and SW virtual components
- If Platform-based design *succeeds*:
 - Most designs are straight-forward *derivatives*
 - Design = configuration, selection, “soft” programming
 - Platforms can be designed with blood, sweat and tears
 - System design helps, but only a small community

5

The Hardware Scenario

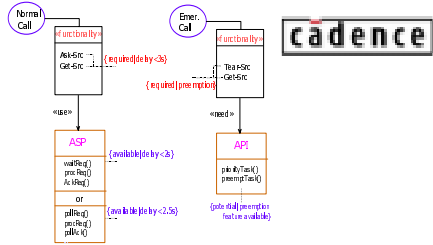
cadence



- Bob Brodersen/BWRC SSHAFT group are key proponents for this approach
- Since HW = 100X better than SW in power and area, and 10X+ in performance, why do anything in SW?
- Direct mapping from algorithms in high level models to HW implementation via automated flows
- BUT: is this likely? For most products in DSM technologies?
 - Requires Masks, \$1M NRE, a Month, risk Management
 - Also requires generalised behavioural synthesis, beyond the known niches

6

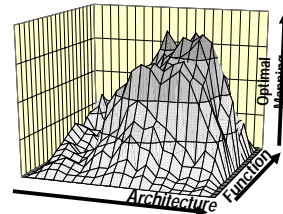
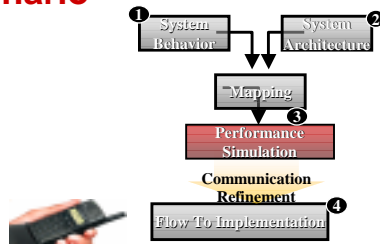
The Software Scenario



- *Au contraire*, most products will be done in software only
- Assumes triumph of the Platform-based approach
 - Derivatives will be done soft, even reconfigurable HW done “soft”
- SW people will not take kindly to HW-based concepts of ‘system-level design’ or ‘high-level modeling’
- Much more likely to:
 - Hack C code
 - Use UML/SDL and the like and base flows on these
 - In this scenario, system level design = “software-software codesign”

7

The Optimistic System-Level Design Scenario



- Function-architecture codesign is ***needed***
- Platforms imply implementation ***choice***
- ***Design space exploration*** is key to optimal product design
- Models of computation are ***important!*** Dataflow \neq control!

8

What are the pre-requisites for the optimistic scenario?

cadence

Lingua Franca



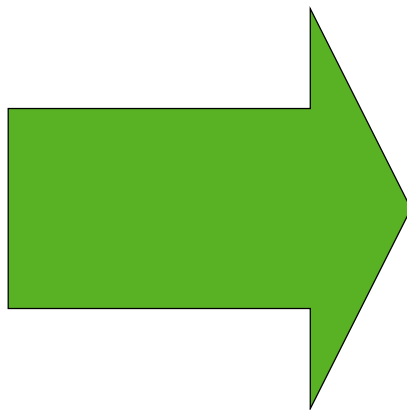
**Consensus on
methodologie(s) and
Model abstraction(s)**



Flow



9



10

What is the likely future?



- I am an optimist
- System Level Design will have a Radiant Future
- We've got to work hard to make it happen
- We've got to keep educating people on the important issues.