

# *Post-Silicon Debug*

***DAC 2012***

**David S. Erikson**

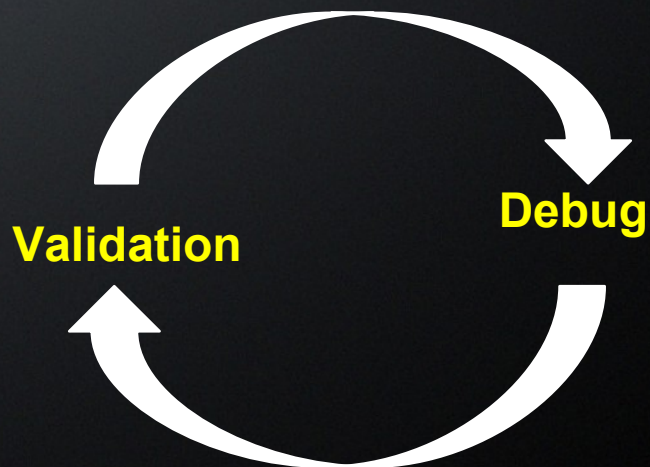
**June 7, 2012**

**Public**

**AMD** 

# AMD Debug and validation Infrastructure

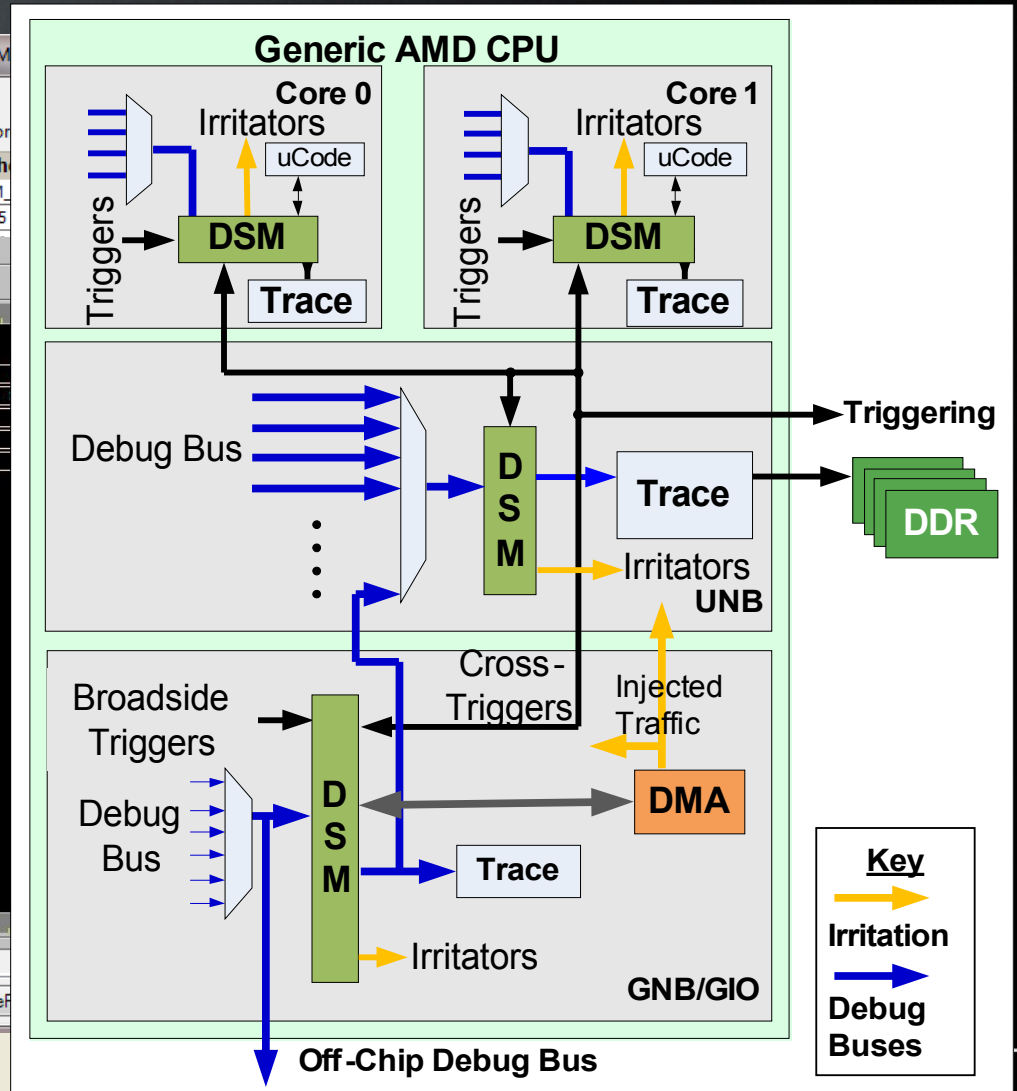
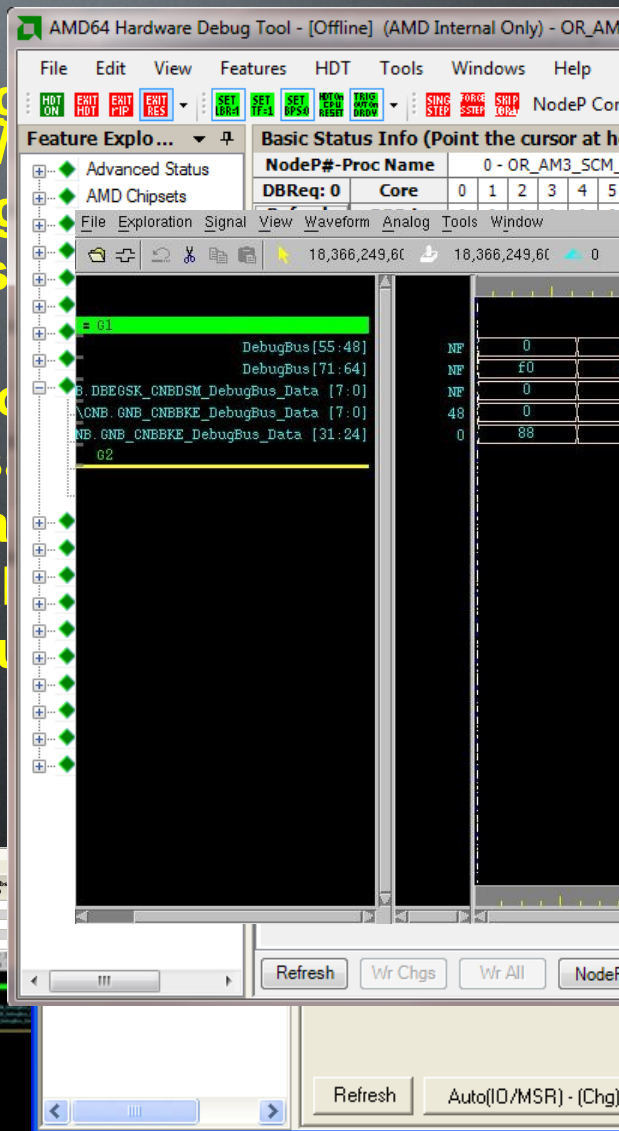
- AMD has capable debug, validation, and automation toolsets which have been successfully deployed through many silicon programs
  - On die observation, actions, and controls for quickly root causing myriad issues – lower Days To Solve metrics
  - Strong validation methods and suites to quickly find issues – lower Days To Find metrics
  - Debug and validation use the same hooks – observe, control, automate





# On Die Observation and Control Infrastructure

- Debug (DSM)
- Debug
- Cross
- Trace
- Irritators
- Trans
- Global
- Scan
- SW S



## Verification and validation – Left Shift

- Waiting for first-silicon is **NOT** the solution to verification bottlenecks
  - We need to shift as much as possible to the left
- While verification isn't perfect and scaling is an issue, post-silicon validation is not the only answer
- Other pre-silicon methods augment verification
  - **Simulators** provide significant coverage of software execution
  - **Emulation** provides significant coverage of the RTL and entire SW stack – including microcontroller firmware stacks
- **The largest schedule impact available is defining the right product with a simulator architecture**

SimNow™



## Current State

- **On die hooks are a given** – the question is how much die real estate to spend
  - Time to money, balanced with quality, is the metric driving most organizations
  - What features, focus, should the hooks provide?
- AMD's debug infrastructure has proven **sufficient or better** for debug in recent history
- The largest challenges for AMD debug hooks are **NOT IN DEBUG**
  - **Complexity** Management
  - Validation **Coverage**
  - **Power Management**
  - **Security**
  - Applying hooks to new IP, features, **architectural changes**





# Hooks Challenges

- **Complexity** is a challenge for users
  - How do I use these hooks?
- Validation **coverage metrics** are the largest problem
  - When are you done? Did you cover everything?
  - Same problem as verification, but without the visibility or control
- **Power management** with increasing complicated power schemes pose a current and future challenge for debug/validation hooks
  - 3 to 5 SOC power domains to 10, 20, more?
  - Debugging continuity across power events is challenging
- **Security** schemes with more stringent requirements pose a current and future challenge for debug/validation hooks
- **New architectures** and features
  - What is needed? HSA/GPGPU for example



# Complexity Challenges

- Hooks **complexity** leads to lower than desired usage by the debug community



We are here



- Mitigated by **abstraction** via simpler software layers
  - GUI and scripting based
  - Libraries of macros and base scripts
- Debug hooks have great similarity across the product line
  - Helps with tools, understanding, and usage
  - Aids in re-use across products



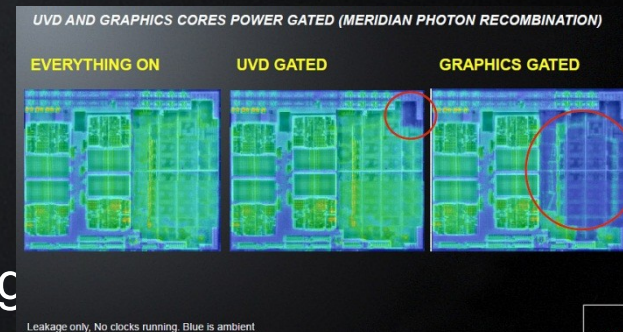
## *Coverage in Hardware and via DFD hooks*

- PMCs, counters, etc. are the norm. Some are architectural.
- How to get coverage metrics like our **pre-silicon** friends?
  - Hardware coverage metrics, self-checking and self-measuring tests, and programmable coverage metrics
- **Hardware assertions** – hardware and programmable using debug hooks
  - CPU, GPU, and SOC
- **Coverage points** – hardware and programmable using debug hooks
  - Events – toggle and functional
  - States and arcs
  - Microcode arcs
  - Performance metrics
- Internal block power, voltage, thermal, voltage margin analysis



# Power management Complexities

- Debugging across power domains
  - Debug features become part of normal save/restore routines
  - Able to “fake” power gating and keep rails up
  - Change or pause normal power up/down flows
- Many bugs lurk in the PM interactions
- Best practices
  - Try to make PM transparent for debug
  - Visibility of power state, power management state, and clock gating state
  - Control of power gating FETs, gating sequence timing and order

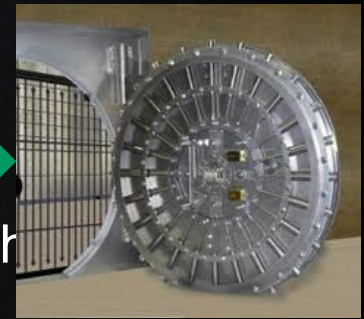


# Security challenges

- Secure Boot, Authenticated Software, DRM, Device, Key Management
- Security impacts the ability to debug a device
- Debug (visibility and control) is directly opposite of security
- Best practices:
  - Map debug features to security states ensuring proper alignment
  - Verify debug features crossed with security states
  - Validate debug feature presence/usage in each security state
  - Employ security processor for debug/validation



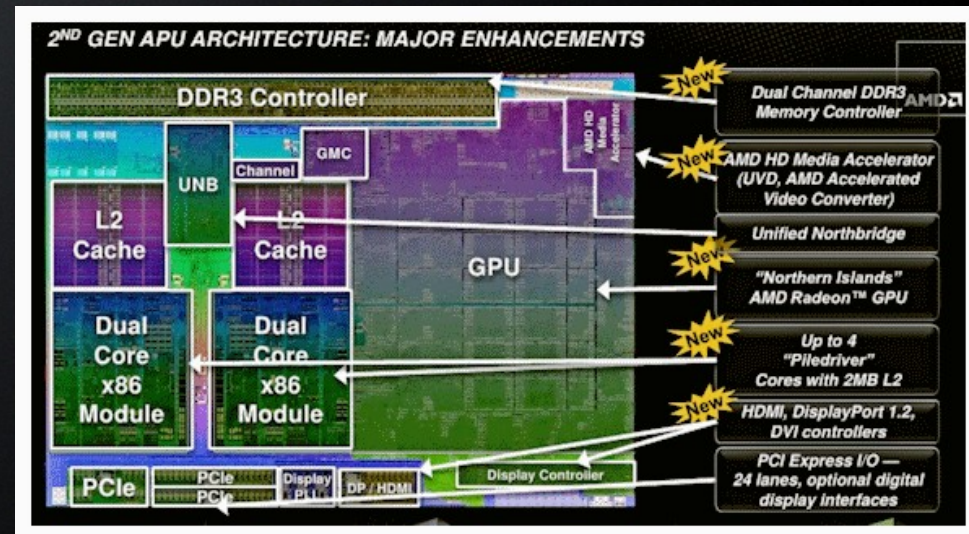
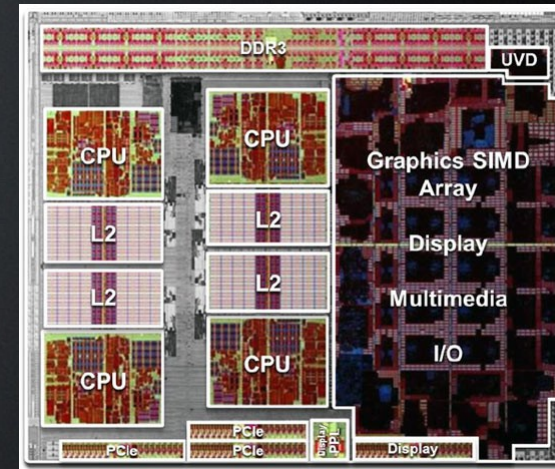
Auth





# New features and architectural elements

- HSA is a good example of a newer architectural element
- HSA is a GPGPU computing initiative
- Current AMD APUs use HSA
- Paradigm shift for GPU with HSA
  - Becomes a coherent agent
  - Executes code without driver layer intervention
- DSM integration in GPU supports quick hardware root cause
  - New hooks created



## *Hooks: Thoughts about next steps*

### ***Validation Agents & portable validation***

- Encapsulation of module validation inside the IP, Subsystem, and SOC
  - IP stimulus/response suites
    - Utilizing subsystem validation agents
    - Validation concepts modularized and portable with IP for re-use
  - Subsystem validation agents
    - Support for local transaction generation and receipt
    - Subsystem validation suite – supports subsystem re-use
  - SOC wide validation agents
    - Use subsystem validation agents for SOC concurrency injection
    - Use functional bus-masters for SOC concurrency injection





## *Hooks: Thoughts about next steps*

### *Verification Concepts in Post-Si*

- Pre-silicon verification-like post-silicon validation automation
  - Model concepts on SystemVerilog/OVM/UVM
  - Controlled Stimulus on die – constrained random support via agents
  - Measurable responses – transaction receiver and interface monitor
  - Coverage metrics and assertions
  - Not only CPU based, but SOC based supports
  - Coverage database supports via drivers, monitors
- Couple this with existing x86 randomization and SOC randomization
- Augment coverage points by observing normal functional execution
  - System Test and Compatibility
  - Directed and randomized diagnostics



## *Hooks: Thoughts about next steps*

### *Pre/Post silicon synergy*

- Driving post-si validation & debug hooks into use in verification
  - Improves the hooks, provides verification of the hooks
  - Provides the ability to correlate pre-si and post-si coverage
  - Streamlines tool usage pre- and post-silicon
  - Hooks must be sufficiently useful to gain traction in verification





Debug and Validation are two sides of the same coin



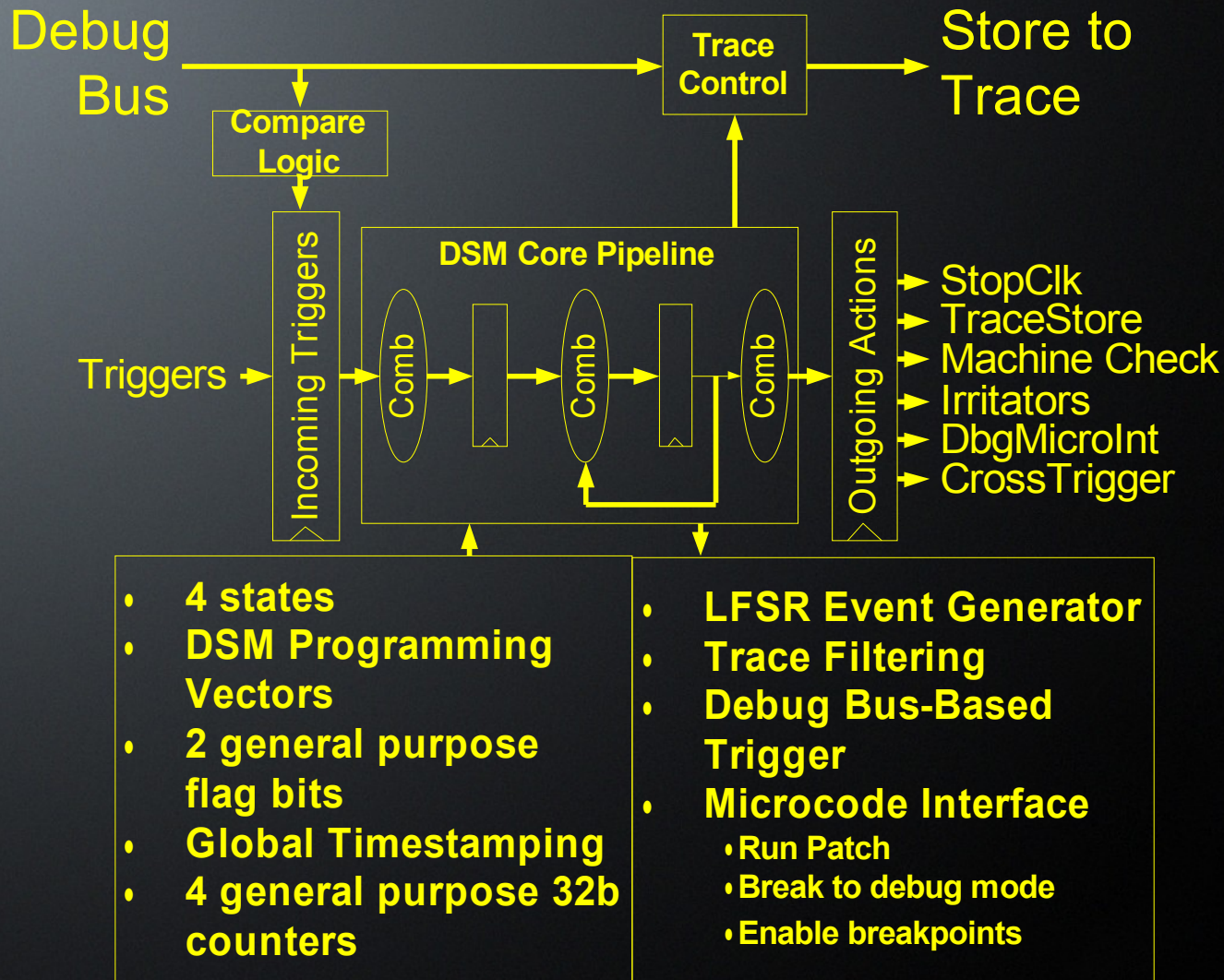
Focus on Validation Hooks will create innovations in Debug

***Backup***





# Debug State Machine



## ■ Disclaimer & Attribution

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. There is no obligation to update or otherwise correct or revise this information. However, we reserve the right to revise this information and to make changes from time to time to the content hereof without obligation to notify any person of such revisions or changes.

NO REPRESENTATIONS OR WARRANTIES ARE MADE WITH RESPECT TO THE CONTENTS HEREOF AND NO RESPONSIBILITY IS ASSUMED FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

ALL IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. IN NO EVENT WILL ANY LIABILITY TO ANY PERSON BE INCURRED FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD, AMD Radeon, the AMD arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc. All other names used in this presentation are for informational purposes only and may be trademarks of their respective owners.

©2012 Advanced Micro Devices, Inc. All rights reserved.

