

NINETEENTH ANNUAL

BiTS

TM

Burn-in & Test Strategies Workshop

March 4 - 7, 2018

**Hilton Phoenix / Mesa Hotel
Mesa, Arizona**

Archive

COPYRIGHT NOTICE

The presentation(s)/poster(s) in this publication comprise the Proceedings of the 2018 BiTS Workshop. The content reflects the opinion of the authors and their respective companies. They are reproduced here as they were presented at the 2018 BiTS Workshop. This version of the presentation or poster may differ from the version that was distributed in hardcopy & softcopy form at the 2018 BiTS Workshop. The inclusion of the presentations/posters in this publication does not constitute an endorsement by BiTS Workshop or the workshop's sponsors.

There is NO copyright protection claimed on the presentation/poster content by BiTS Workshop. However, each presentation/poster is the work of the authors and their respective companies: as such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

The BiTS logo and 'Burn-in & Test Strategies Workshop' are trademarks of BiTS Workshop. All rights reserved.

www.bitsworkshop.org

SOC Low Power Debug Techniques

Krishna Dandamaraju
Intel Corporation



BiTS Workshop
March 4 - 7, 2018



Agenda

- Introduction
- Low Power states in Intel SOCs
- Low Power Debug Challenges
- Low Power Debug Methods
- Typical example

What is Power management?

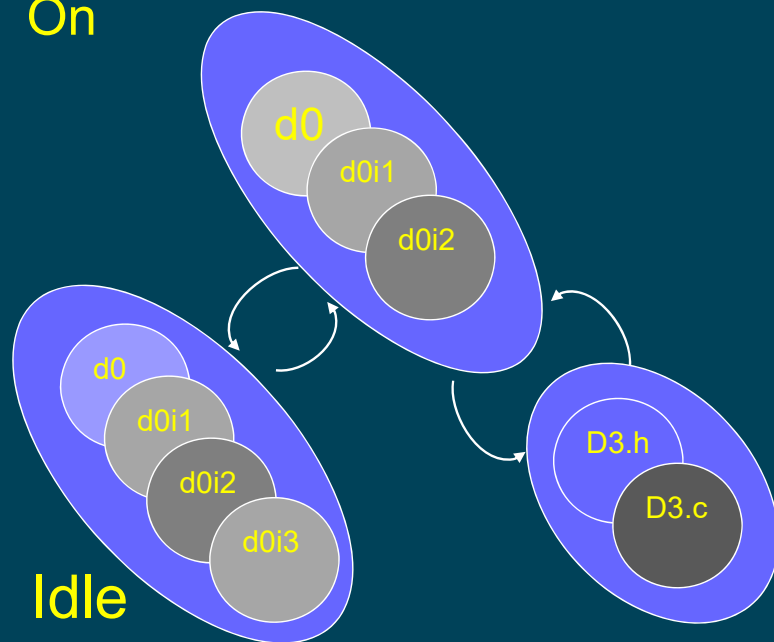
- Techniques to improve battery life without performance impact.
 - Phones, tablets, laptops, servers.
- Idle State Power management
 - Doing nothing efficiently.
 - When Idle, save power, but available instantly when needed.
 - Key to achieve long battery life. Pay attention to latencies.
- Active State Power management
 - Finish the task fast enough, to satisfy user needs, with least amount of power.
 - Operate at optimal operating point (voltage, freq) ,
 - Obey the power , thermal constraints to finish the task under given constraints.

Introduction

- Power management complexity is increasing
 - Active (Perf –P states) / Idle states (Sleep -S states), concurrency , Many power / clock domains.
 - Save state ; Turn off clock and power rails; Turn on ; Restore state
- Aggressive low power states to achieve battery life
 - Device (D states), Compute Engine (Cstates), System states (Sleep-Sx, Standby-S0ix)
 - Move to efficient active state or a low power state often and quickly
 - Many moving parts → synchronization issues with reset , clock
- Debug resources , tools managed through power states
 - Limited observability around deep power state transitions.
 - Debug tools themselves use power rails, clocks, fabrics that are shut down.
- Complexity/ limited observability →challenging debug

Low Power States in Intel SOCs

On



Idle

Device States

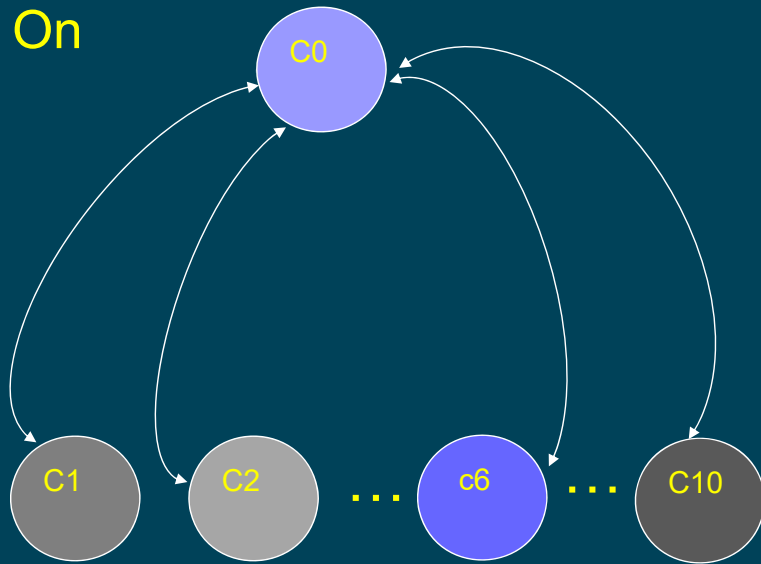
(USB, sensor hub, audio, pcie ...)

Off

Clocks, Power Rails turned off
Loss of state

Low Power States in Intel SOCs

On



Compute Engine States
(Cores, GFX, Imaging, ...)

Idle

Off



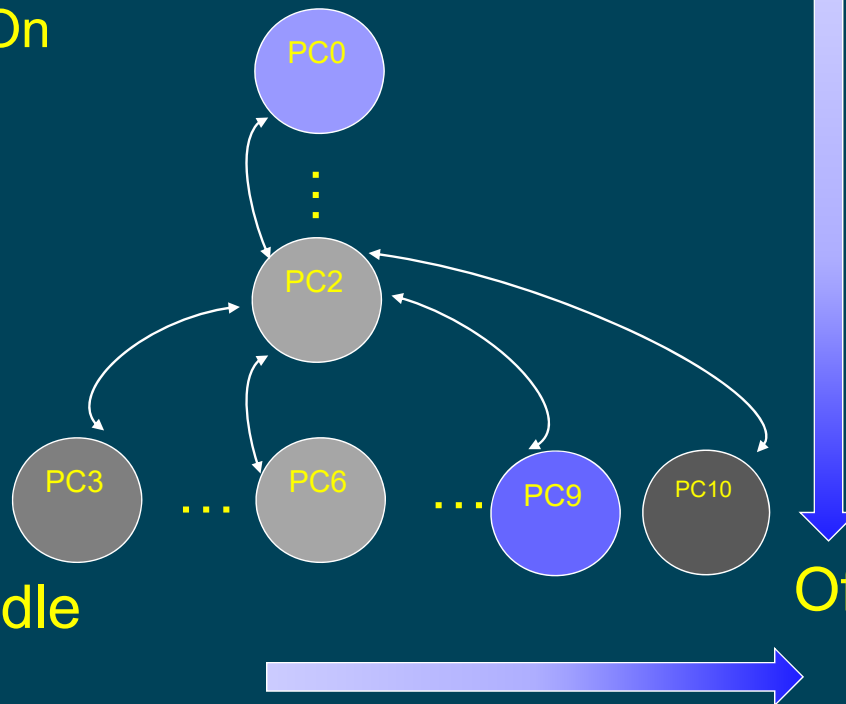
Low Power States in Intel SOCs

On

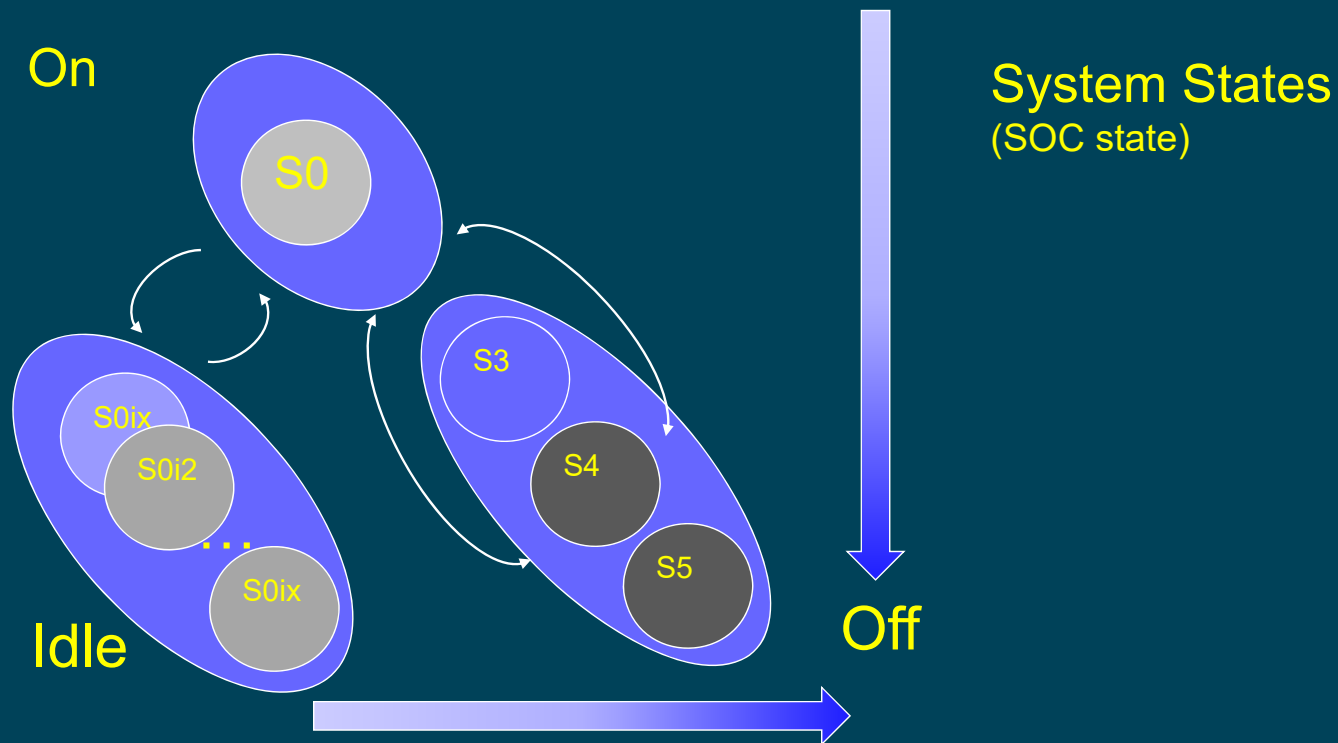
Idle

Package States
(Computing sub system)

Off



Low Power States in Intel SOCs



System States
(SOC state)

Off

Low Power Debug Challenges

- Accessibility to state during low power states
- Debug resources themselves go to low power states
 - Additional complexity for DFX tools to survive and functional through power states
 - DFX is the last one to shutdown and first to bring up. Debug data saved in deep power wells when dfx is down.
- Marginality issues → Large Mean Time Between Failure (MTBF)
→ hard to reproduce
- State corruption may not result in visible failure immediately.
- Symptoms of failure vary vastly
 - Can't power up / wake up , devices doesn't work , memory corruption

Low Power Debug Techniques

Debug tool	Description
State Dump	Freeze clocks and dump state of every Flop in the system -
Machine Check Arch.	Errors/ hangs detected by HW, FW.
HW Trace	Trace HW signals, to memory or Logic Analyzer. State machines, busses, control signals ..
FW Trace	Trace FW debug messages
Crash log	Store critical failure information in RTC well , retrieve after reset.
FW debug hooks	Run Control of power management uCs
Telemetry	Power state / Perf information accessible to OS on real time basis
Delayed Authentication	Unlock machine after failure to extract debug information

Low Power Debug Techniques usage

Debug tool	Usage Scenarios
State Dump	Debug speed paths, state corruption, deadlocks , hangs
Machine Check Arch.	Stop on a error, Trigger
HW Trace	History building up to a hang/ deadlock
FW Trace	Power flow sequences, history of flow leading to a failure
FW debug hooks	Stop at critical phases, check or change state, explore work arounds,
Crash log	Reason for last crash. Volume / customer environment. Debugger not hooked.
Telemetry	Residency issues, overall picture for battery life or perf issues.
Delayed Authentication	Volume validation / customer env. Hook debugger after failure.

Debug example – Initial triage

- **Symptom**: Volume validation environment . USB harasser. system reboots after week of stress testing.
- **Triage** : **Windows event trace** : un-expected reboot. **Crash log** shows FW **Machine check** while resuming from **S0ix** flow, while restoring GFX.
- **Further debug** : Halt PM micro controllers (**Run control**) on a machine check error. Analyze last few **FW Trace** messages for context.
messages around s0ix transition extracted from RAMs in PM micro controller deep power well.

Debug example – Detailed debug

- **Immediate reason**: State dump leaves several bread crumbs of last few events. Shows Read from PM uController gets UR (unsupported Response) from GFX.
- **Root cause** : Compare state dumps normal s0ix exit Vs failed exit. Shows Reset values of some registers corrupted. Doesn't allow reads.
Produce theory (clock synchronization issue) , verify in simulation.
- **Workarounds**: Alter the timings of reset, clock using FW control, gating to avoid the issue

Call for Action

- Design for Debug focus in Power Management Architecture.
- Standardized , Power state resilient debug features.
- Well thought out debug plan for Power Management scenarios.

Disclaimer

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at www.intel.com.

Intel, the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.