Early Hardware and Firmware Co-Development Using Hardware Acceleration

Joel Artmann
Medtronic

Presented at CDNLive Boston
August 2013

Introduction

- Author
  - Joel Artmann – Design and Verification Engineer at Medtronic.
  - Acknowledgements – EME Team
- Medtronic
  - Medical Device Technology and Therapies
  - Implantable Pacemakers and Defibrillators
- EME
  - Evaluation Methods and Environments
  - Verification Environment Group
- Acceleration Use at Medtronic
  - Enable FW Development on RTL Design Model
  - Reduce need for abstract model
  - Hardware and Firmware integration early in development

Agenda

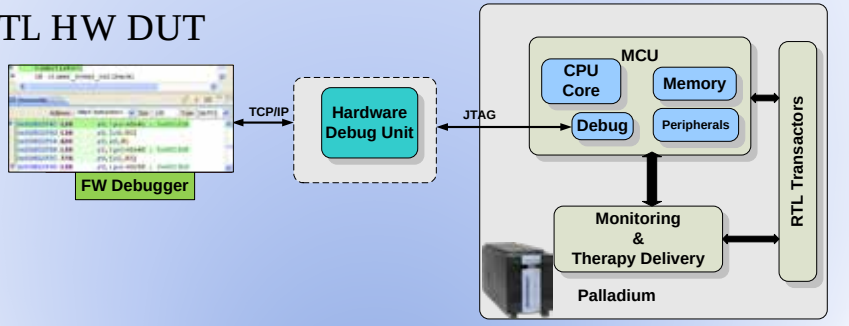
- Acceleration Use Cases
- Specific Challenges (Problem Statements)
- Results and Achievements
- Future Direction

Acceleration Use Cases

- FW Development
- Device Level Testing
- Co-Development & Deb

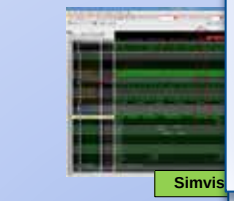
FW Development

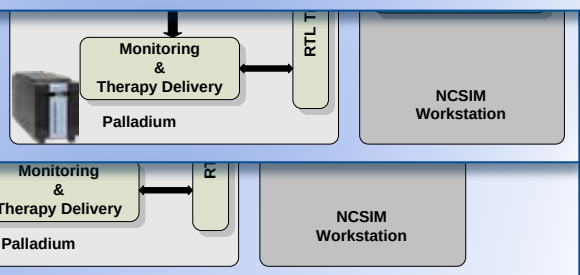
- FW Development Studio (IDE)
- FW Debugger
- RTL HW DUT



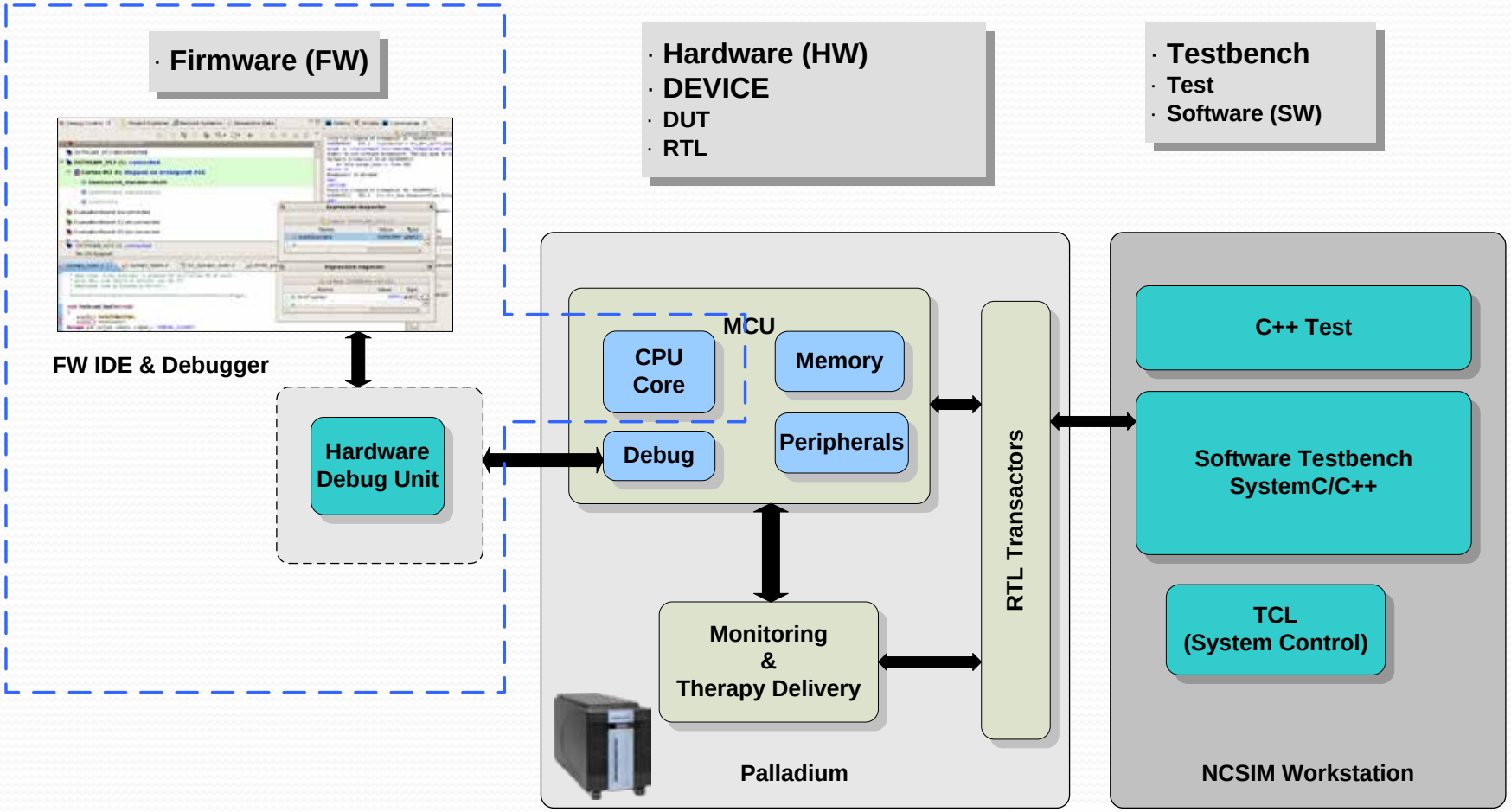
Co-Develop

- Legacy Test Support
- Legacy Test API
- FW Debugger
- Simvision HW Debugger
- RTL HW DUT



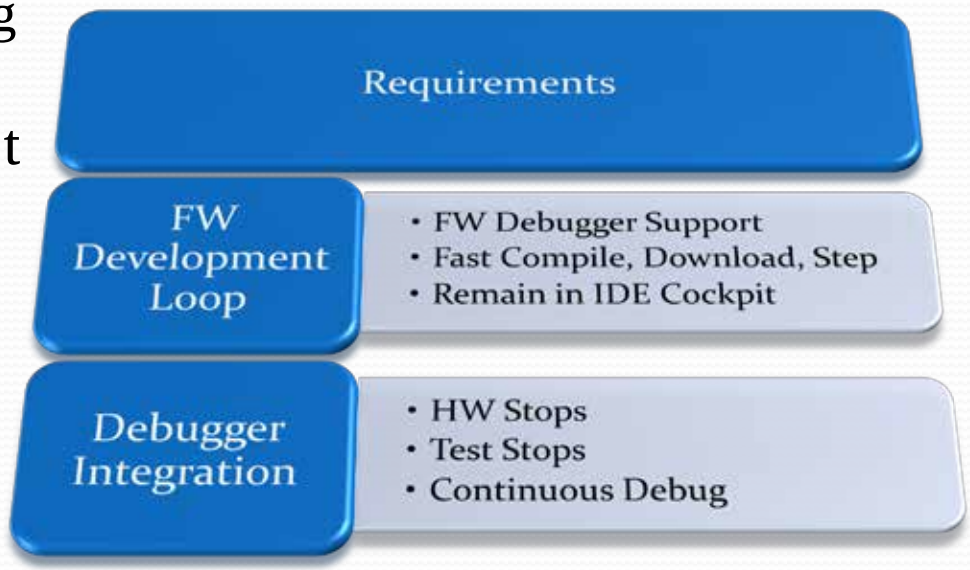


Terminology



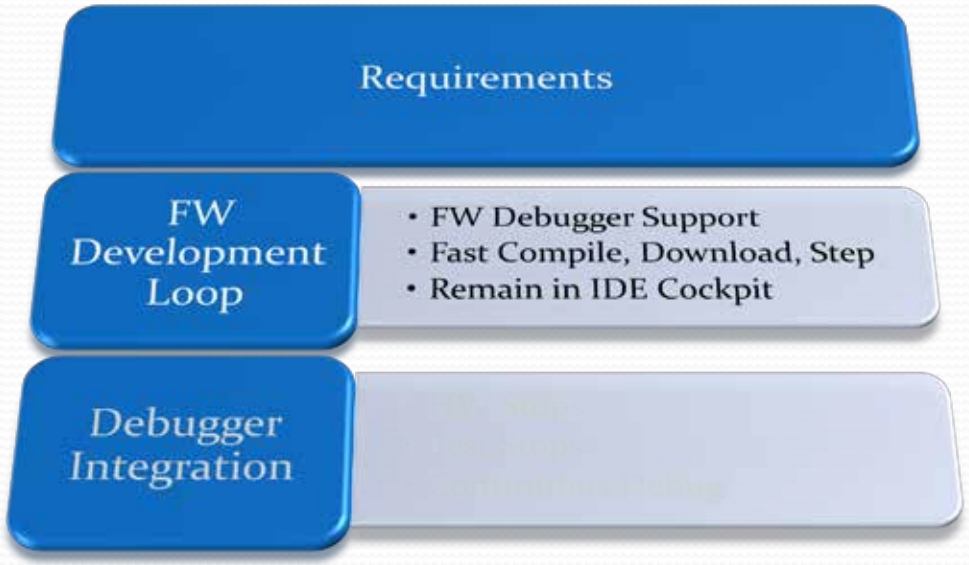
Challenges

- Device Level Testing
  - Ported environment
  - Architecture most challenging
  - Device Level Coverage
- Co-Development Environment
- FW Development
  - FW Debugger Support
  - Fast Compile
  - High Speed Download
  - Quick Step
  - Low Loop Time Latency
  - Hardware Stops
  - Test Stops at a FW Breakpoint
  - Batch-able Tests



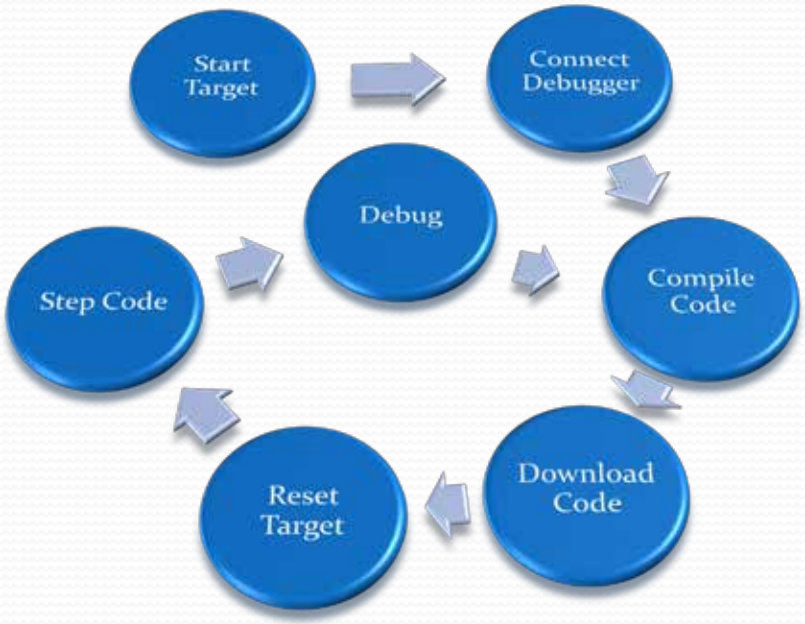
FW Development Loop

- Development & Debug Experience



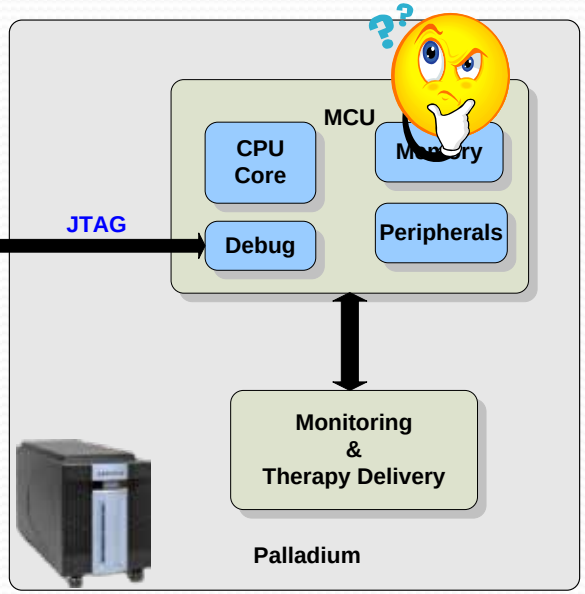
FW Development Loop Expectations

- Start Target
- Connect Debugger
- Download Code
- Restart Target
- Step Code
- Debug
- Repeat



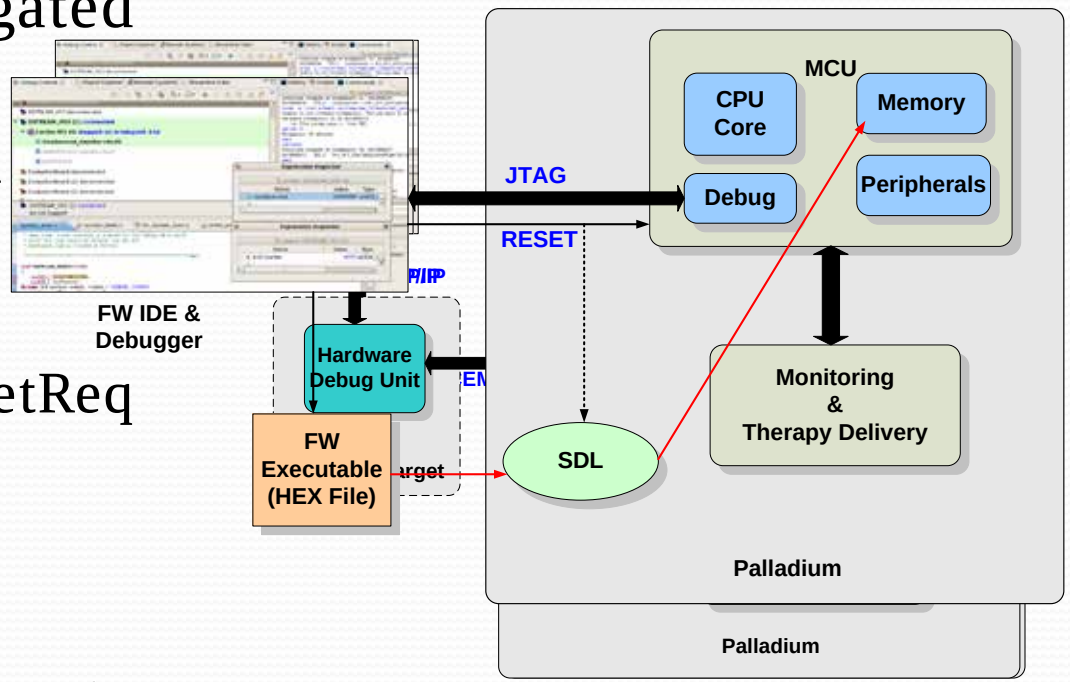
FW Development Loop – Hard?

- 1. Connect a FW debugger?
  - COTS Debugger
  - Virtual or Physical connection?
- 2. Load new FW?
  - No FLASH loader.
  - JTAG download is slow
- 3. Remain in IDE
  - Automatic Invocation with IDE.



FW Development Loop - Solution

- 1. Software JTAG Investigated
  - Not optimal
  - Physical Debugger Unit
    - Dynamic Target Mode
- 2. “Soft” Load Code
  - SDL detects SystemResetReq
  - Automaticaly soft load memories
- 3. IDE Cockpit
  - Single “Button” starts target.
  - Builds are referenced.



Debugger Integration

- Hardware Stops at a FW Breakpoint
- Test Stops at a FW Breakpoint

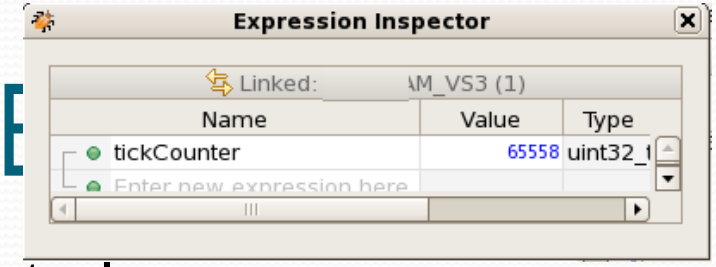
- FW Breakpoints

FW Breakpoint E

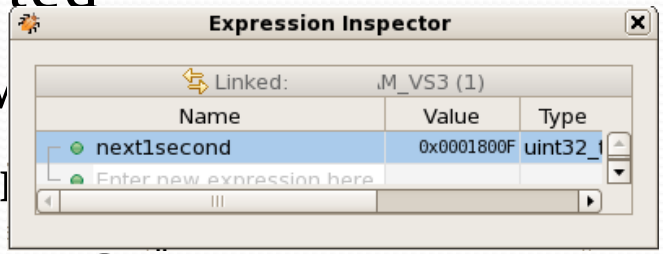
```
void OneSecond_Handler(void)
{
    uint32_t ticksToNext1Sec;
    uint32_t tickCounter;
#pragma arm section rwdata, zidata = "GENERAL_CLEARED"
    static uint32_t next1second = 0;

    tickCounter = Drv_Hrt_GetTickCounter();
    next1second = next1second + NUM_OF_TICKS_IN_1_SEC;
    ticksToNext1Sec = NUM_OF_TICKS_IN_1_SEC - (tickCounter - next1second);
    if ( 2 * NUM_OF_TICKS_IN_1_SEC < ticksToNext1Sec)
    { // The two clocks are out of sync
        ticksToNext1Sec = NUM_OF_TICKS_IN_1_SEC;
        next1second = tickCounter + NUM_OF_TICKS_IN_1_SEC - 1;
    }
    Drv_Hrt_StartDedicatedTimerInTicks( HRT_TMR_8, ticksToNext1Sec, OneSecond_Handler );

    OneSecTsk();
}
```



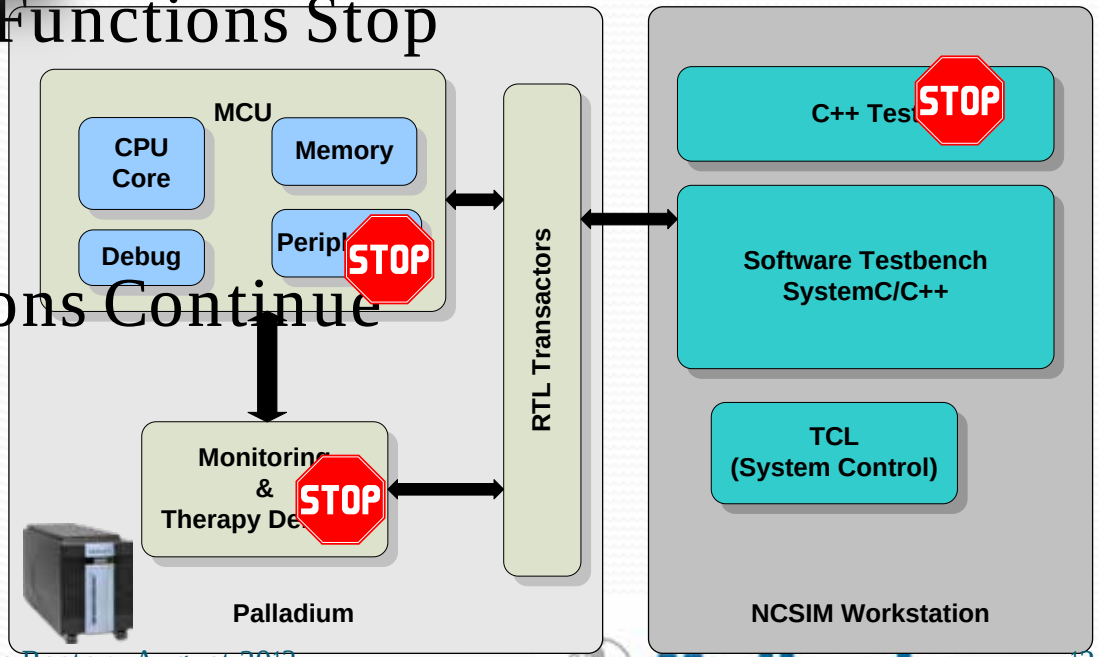
upported



orted v

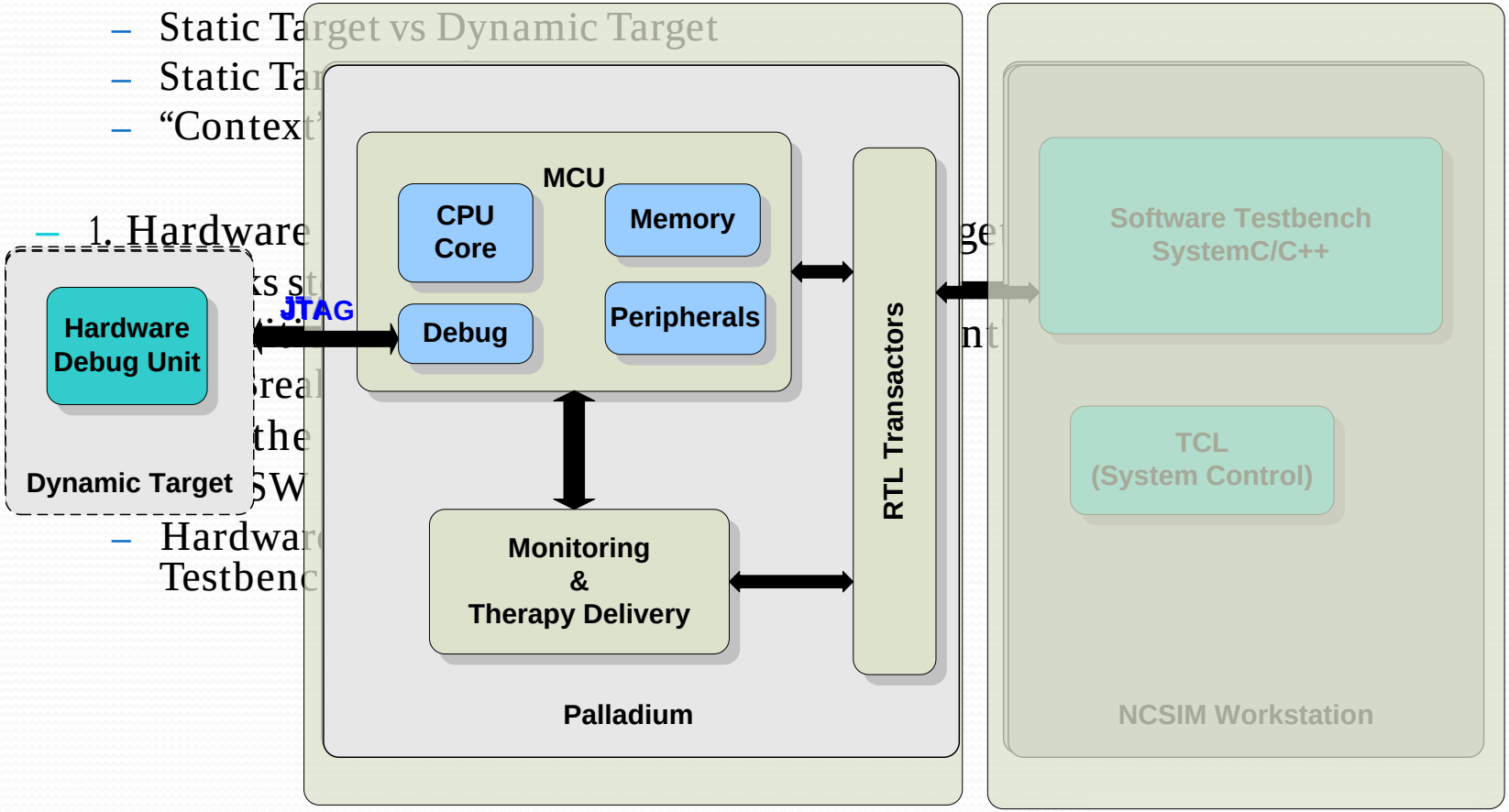
tion Ti

- Critical Hardware Functions Stop
  - Timers
  - Device Features
- Debugger Operations Continue



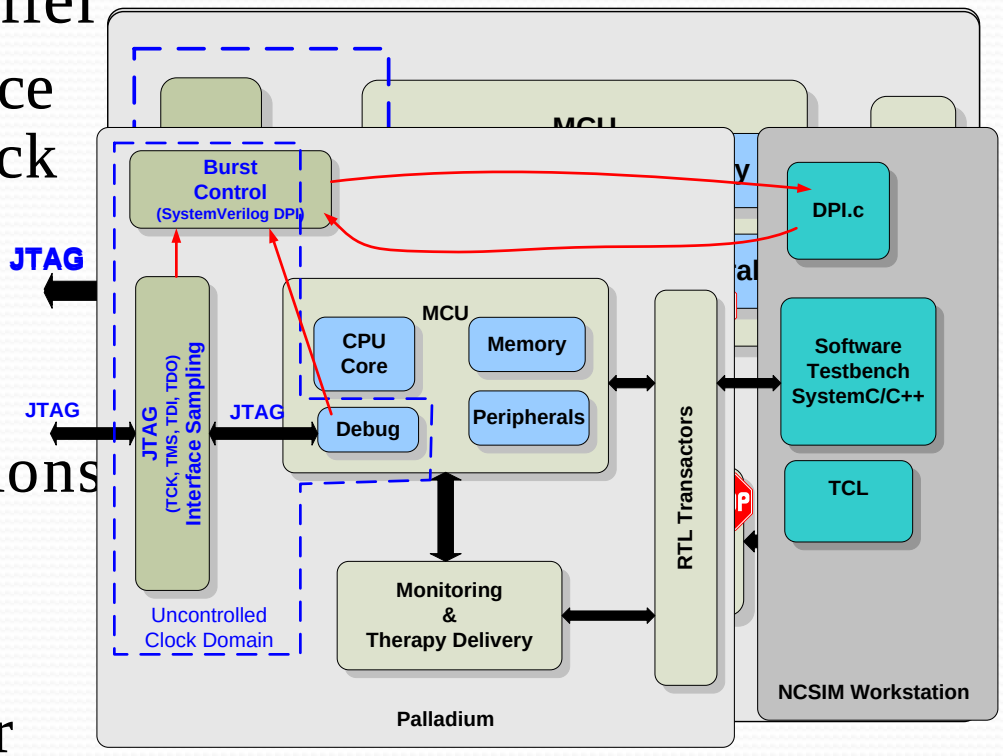
FW Breakpoints – Why Hard?

- Verification/Test Environment uses "TBrun" Static
  - Static Target vs Dynamic Target
  - Static Ta
  - "Context"
- 1. Hardware
  - Hardwa
    Testbenc



FW Breakpoint - Solution

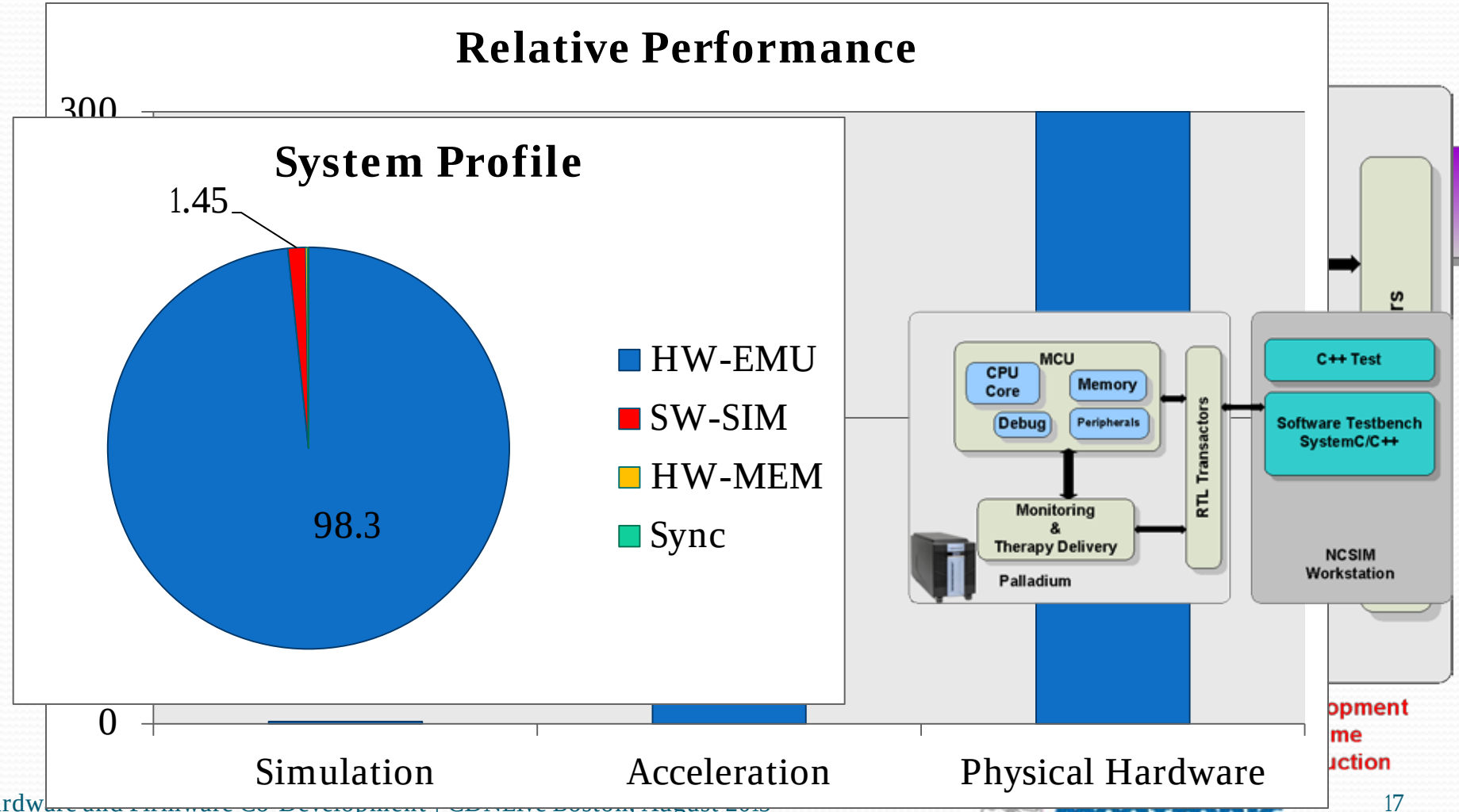
- 1. Uncontrolled Clock Buffer
  - Hardware Target Interface to buffer Controlled Clock Stop
- 2. HALT Distribution
  - Stops GPTimers and Critical Hardware functions
- 3. SystemVerilog DPI
  - In-Circuit-Acceleration
  - While(1) in C waiting for hardware to give OkToRun.



Scorecard

Requirement	Result	Solution
FW Debug Support	✓	Physical Debug Unit
Fast Download, Fast Loop	✓	SDL “Soft” Load of Code
Remain in IDE	✓	Single “Click” button
HW Stop at Breakpoint	✓	SDL Detects HALT Signal
Test Stops at Breakpoint	✓	ICA + SystemVerilog
Continuous Debug	✓	Uncontrolled Clock Domain Buffer

EME Accelerator Achievements



Future

- Low Power Verification and Profiling
- Power Shut-off Verification with CPF
- Test Execution Management
  - Batch Run
  - Coverage Collection
  - Requirement Status
- Project Adoption

Summary

- Configurations Available
  - FW Development
  - Device Level Testing
  - Co-Development & Debug

- Capabilities Utilized
  - **TBA** – Testbench Integration with SCEMI
  - **SDL** – Soft Load, HALT Distribution
  - **ICA** – Dynamic Target with TBA
  - **Uncontrolled Clock** - Dynamic Target ICE Interface