HETEROGENEOUS COMPUTE IN A QUAD CORE CPU

Cyril Jean
Director Embedded Systems Solutions
Microsemi, a Microchip Company

https://tmt.knect365.com/risc-v-summit
Real-time Linux?

- **Wide spread Linux adoption**
  - Rich OS with thousands of applications to choose from

- **Requirements still exist for real-time while running Linux**
  - **Safety critical**
    - The ability to deterministically monitor the execution environment.
  - **Real-time system control**
    - Completing tasks deterministically, on time every time.
  - **Securing the IoT**
    - Execute a trusted execution environment deterministically for consistent results.

- **Working with our partner**
  - We have been able to architect a complex SoC FPGA that provides
    - Determinism and a rich OS within the same multi-core CPU cluster
What is Real Time?

- Subjective concept
  - Perception of the system reacting immediately to user inputs
  - System reacts within x milliseconds to an external input:
    - Usually
    - Most of the time
    - All the time, otherwise
      - The system fails
      - The system can become damaged
      - Somebody might get hurt

- System able to control a physical process at a speed suitable to the process under control

- What we usually mean by Real Time is Determinism
Determinism

- Periodic Interrupts
  - \( T_0 = T_1 \)
- Consistent Execution Times
  - \( E_0 = E_1 = E_2 \)
Standard Application Processor

- Memory Hierarchy
  - L1 cache
  - L2 cache
  - DDR memory
- Micro-architecture performance enhancement features

<table>
<thead>
<tr>
<th>Core 1</th>
<th>Core 2</th>
<th>Core 3</th>
<th>Core 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 cache</td>
<td>L1 cache</td>
<td>L1 cache</td>
<td>L1 cache</td>
</tr>
</tbody>
</table>

L2 cache

DDR4
Memory Hierarchy and Determinism

- Cache misses affect determinism
  - Retrieving data from DDR is non-deterministic
  - Accessing to L2 cache is non-deterministic
Measured ISR execution time in a quad core CPU

- **Periodic Interrupts**
  - $T_0 = T_1$
- **Inconsistent Execution Times**
  - $E_0 \neq E_1 = E_2$

![Execution Time Variability Graph]

- SMP 1
  - L1 cache
- SMP 2
  - L1 cache
- SMP 3
  - L1 cache
- SMP 4
  - L1 cache

- L2 cache
- DDR
- Classic real time system
  - Infinite background loop executes main application code
  - Time critical code is executed as a result of an interrupt
PolarFire SoC Flexible Memory Subsystem

- Configurable L1 memory subsystem
  - As Cache
  - As a Tightly Integrated Memory

- Configurable L2 memory subsystem
  - As a Cache
  - As a Scratchpad Memory
  - As a Loosely Integrated Memory (LIM)
    - Direct addressing of memory
Flexible Memory Subsystem Provides ISR Determinism

- Periodic Interrupts
  - $T_0 = T_1$
- More Consistent Execution Times
  - $E_0 \approx E_1 \approx E_2$

**Execution Time Variability**
Micro-Architecture Also Impacts Determinism

- SMP 1
  - L1 cache

- SMP 2
  - L1 cache

- SMP 3
  - L1 cache

- RT1
  - TIM

- L2 cache

- LIM

- DDR

- T₀
  - E₀
  - ISR
  - Main()

- T₁
  - E₁
  - ISR
  - Main()

- E₂
  - ISR
  - Main()

- Periodic Interrupts
  - T₀ = T₁
- Consistent Execution Times
  - E₀ = E₁ = E₂

Execution Time Variability

Disable branch predictor during critical code execution, or permanently.
Coherent Message Passing in AMP systems

- L2 Cache for SMP Cluster
- L2 LIM for Real-Time
- L2 Scratchpad for coherent message passing
Summary

- Working with our partner SiFive
  - We have been able to architect a complex SoC FPGA that provides
    - Determinism and a rich OS within the same multi core CPU cluster

- Periodic Interrupts
  - $T_0 = T_1$

- Consistent Execution Times
  - $E_0 = E_1 = E_2$
THANK YOU

https://tmt.knect365.com/risc-v-summit