Embedded Linux Conference
Europe

OpenIoTSummit
Europe

Preempt-RT Latency Benchmarking of the Cortex-A53 processor

Paul Thomas, AMSC
Paul.thomas@amsc.com

THE LINUX FOUNDATION

AMSC

- Founded in 1987
- Headquartered near Boston, Massachusetts
- Specializing in the design and manufacture of power systems and superconducting wire

Preempt-RT Latency Benchmarking of the Cortex-A53

- Software and Hardware setup
- Basic Latency Tests
- UDP Ping Pong Ethernet Latency
- Real-World ADC Interrupt Latency

Why Real-Time Linux?

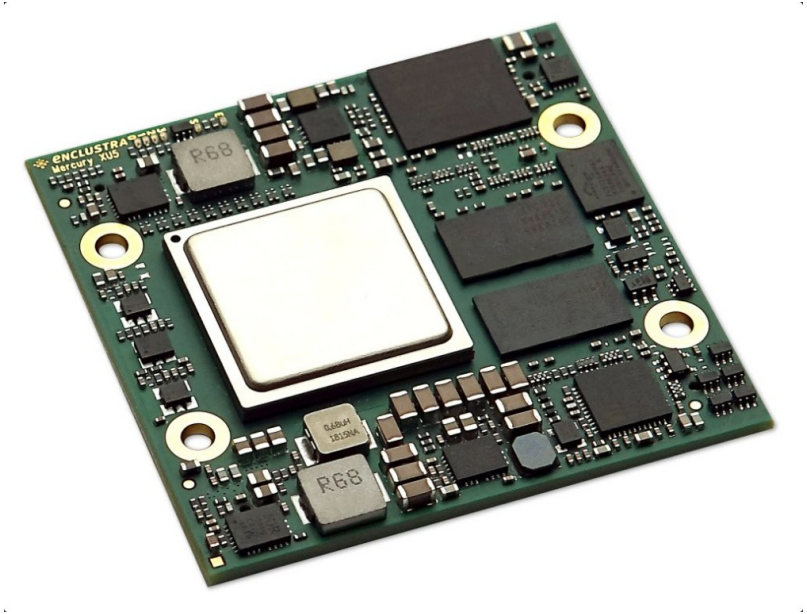
- Stable and supported code base
- Deep APIs (networking, fs, IPC, etc...)
- Good latency performance
- Real Time Linux collaborative project aims to mainline Preempt RT

Processors and Boards that use the ARM Cortex-A53

- Xilinx Zynq UltraScale+ MPSoC
- NXP i.MX 8
- Raspberry PI 3
- ODroid-C2

Hardware Setup

- ARM Cortex A53
  - Module: Enclustra Mercury XU5
  - SOC: Xilinx MPSoC XCZU5EV
  - 8 stage pipeline
  - 1.3 Ghz
- ARM Cortex A9
  - Board: Zedboard
  - SOC: Xilinx XC7Z020
  - 10+ stage pipeline
  - 666 MHz

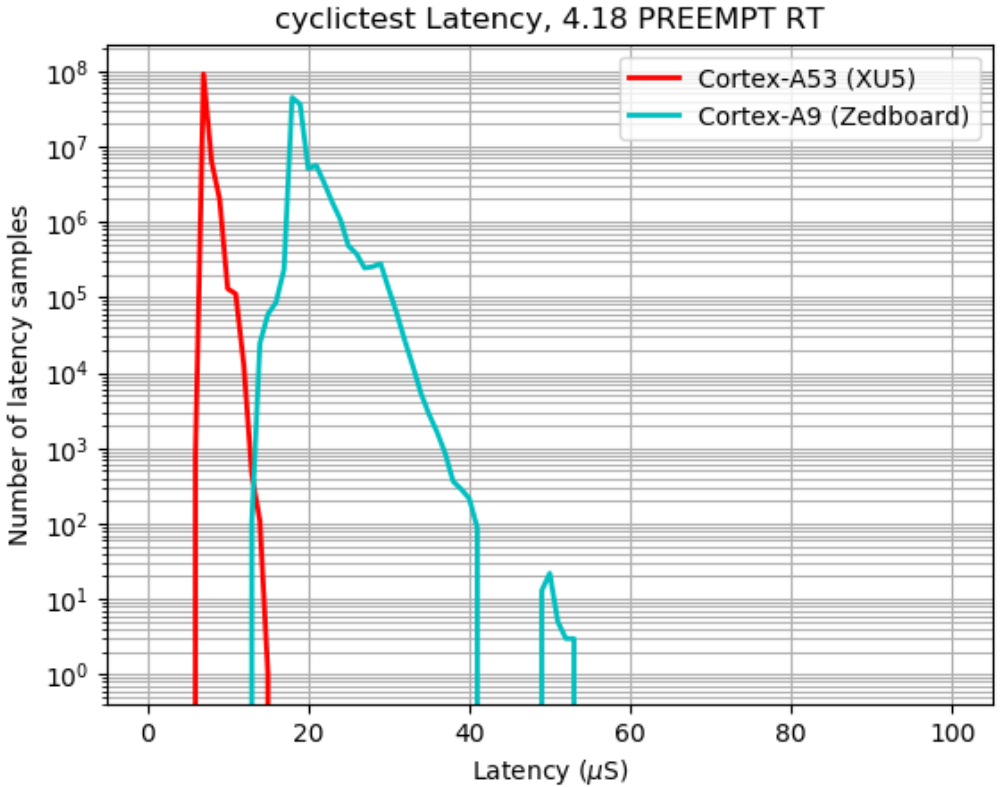


Kernel

- Starting point is 4.18
- PREEMPT_RT 4.18-rc8-rt1 patch applied
  - https://cdn.kernel.org/pub/linux/kernel/projects/rt/4.18/older/patch-4.18-rc8-rt1.patch.xz
- Zynqmp firmware and clock driver patch applied
  - https://patchwork.kernel.org/project/linux-arm-kernel/list/?series=5175

Cyclictest Results

- Cortex-A53
  - Maximum: 17 µS
  - Mode: 7 µS
- Cortex-A9
  - Maximum: 54 µS
  - Mode: 19 µS



Cpusets Shielding

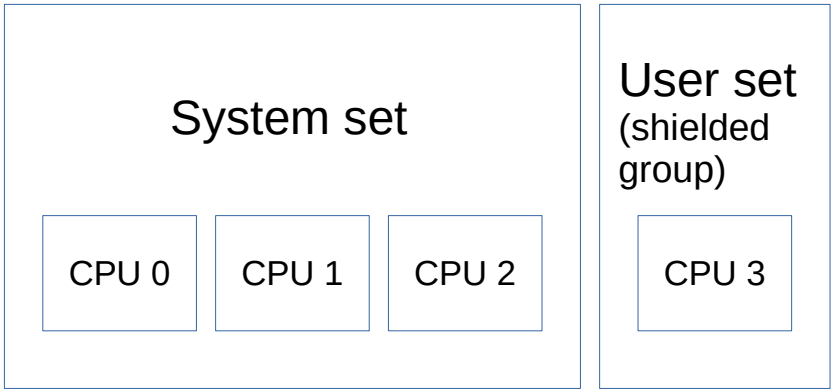
- Kernel portion is CPUSETS
- Userspace management via cpuset
  - https://github.com/lpechacek/cpuset

    “Cpuset is a Python application that forms a wrapper around the standard Linux filesystem calls to make using the cpusets facilities in the Linux kernel easier”

- CPUSETS is an effective way to shield 1 or more cores from scheduling ordinary tasks

CPUSET and Loading Configuration

CPUSET Configuration[1]



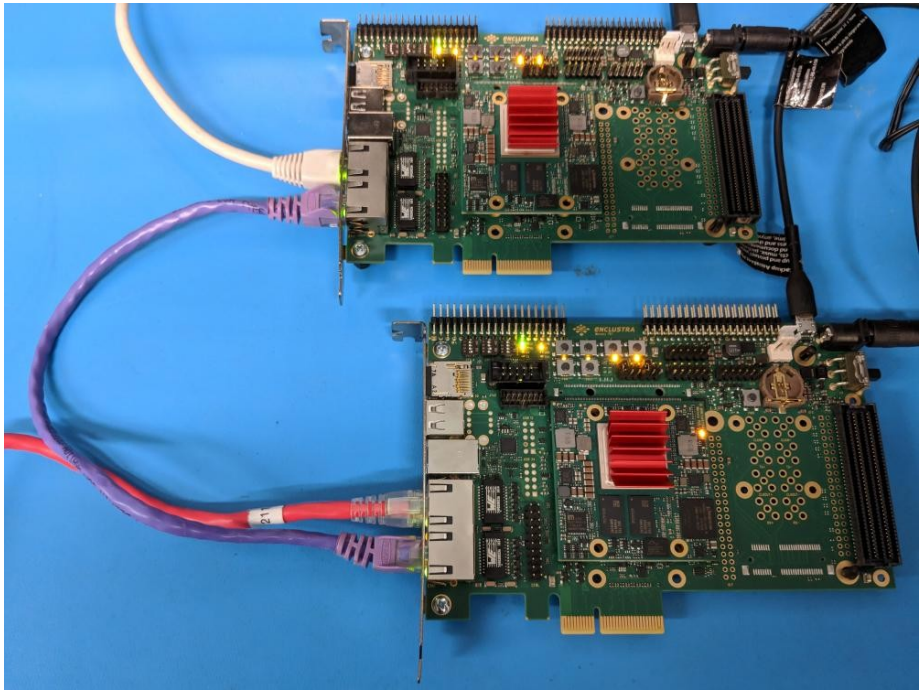
- Test Configuration
  - System Set
    - Cyclictest w/ priority 98
    - Cyclictest w/ priority 99
    - Stress[2]
  - User Set
    - Cyclictest w/ priority 98
    - Cyclictest w/ priority 99 (results presented)

Cortex-A9 is only dual core, System set is just 1 core[1]

[2]stress --cpu 8 --vm 8 --vm-bytes 20MB

UDP Ping Pong Test

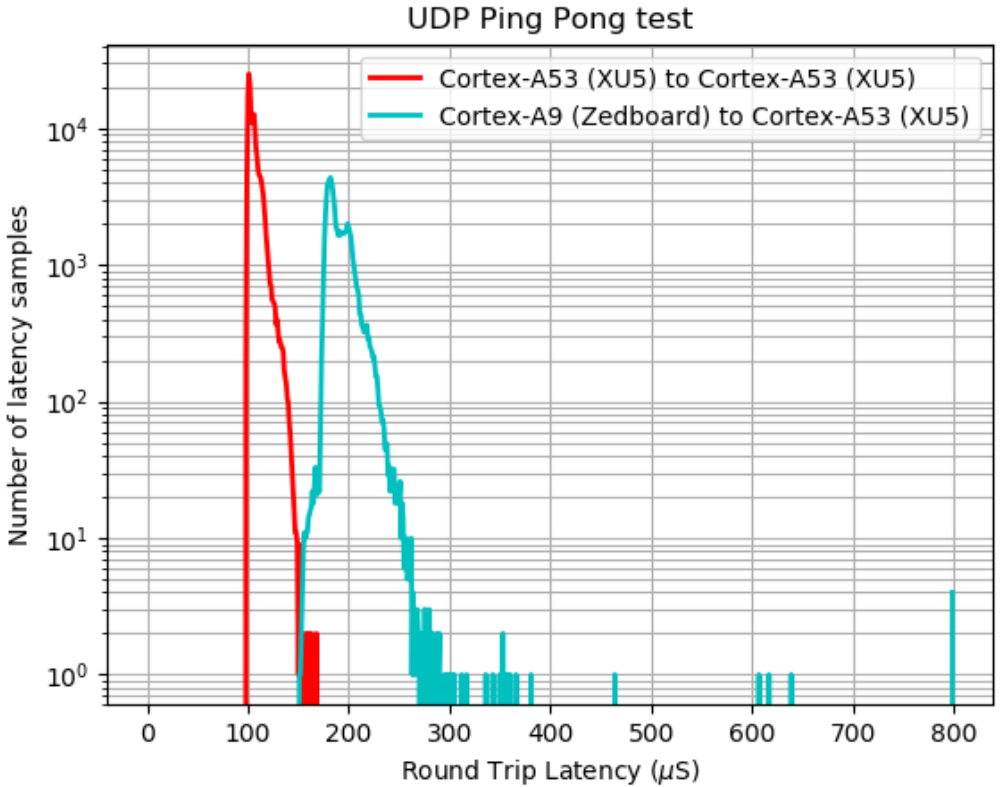
- For Cortex-A53 to Cortex-A53 test dedicated Ethernet port is used
- Zedboard has a single Ethernet port so it is shared with SSH connections
- CPUSETs not used because it adversily affected performance
- IRQ affinity changed to last CPU



UDP Ping Pong Results

- Cortex-A53
  - Maximum: 168 µS
  - Mode: 101 µS
- Cortex-A9
  - Maximum[1]: >800 µS
  - Mode: 182 µS

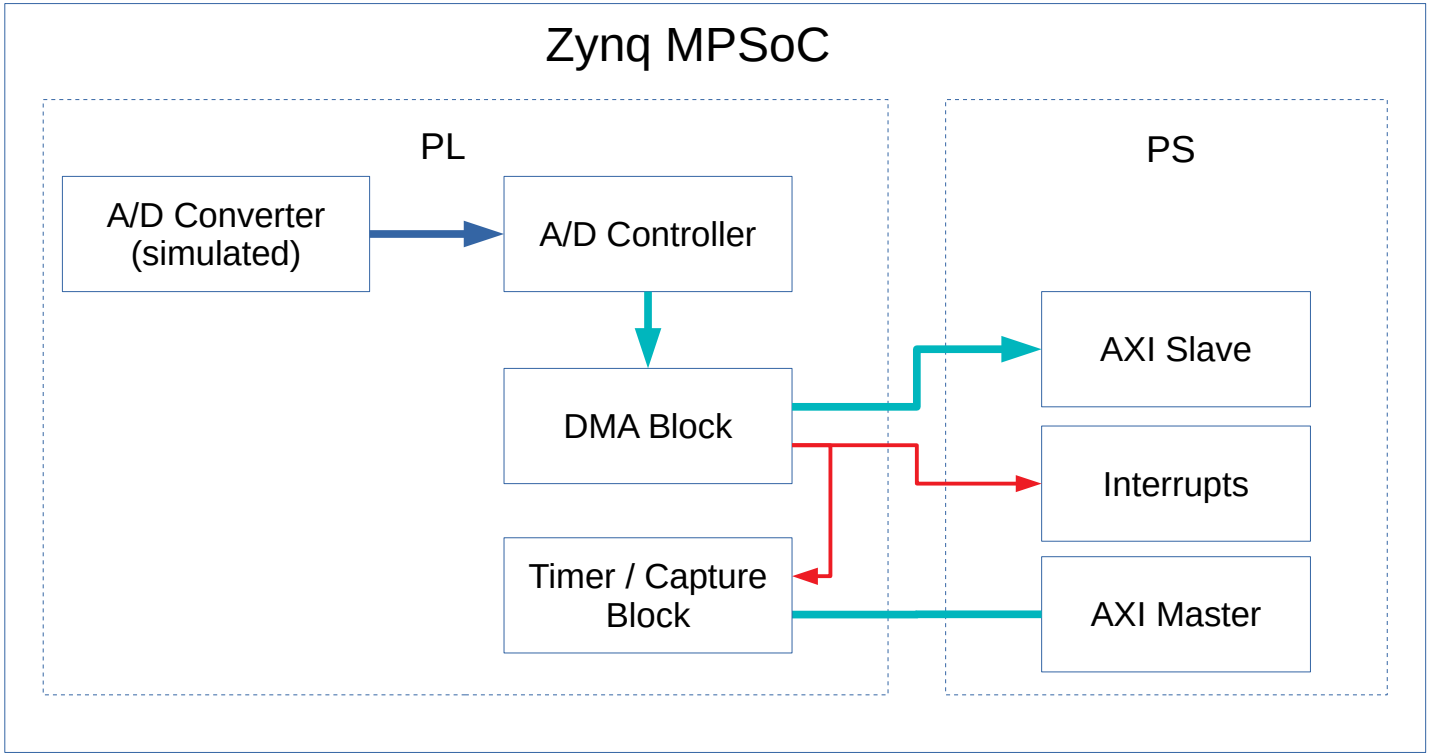
[1]800 µS was largest bin



Real World Test

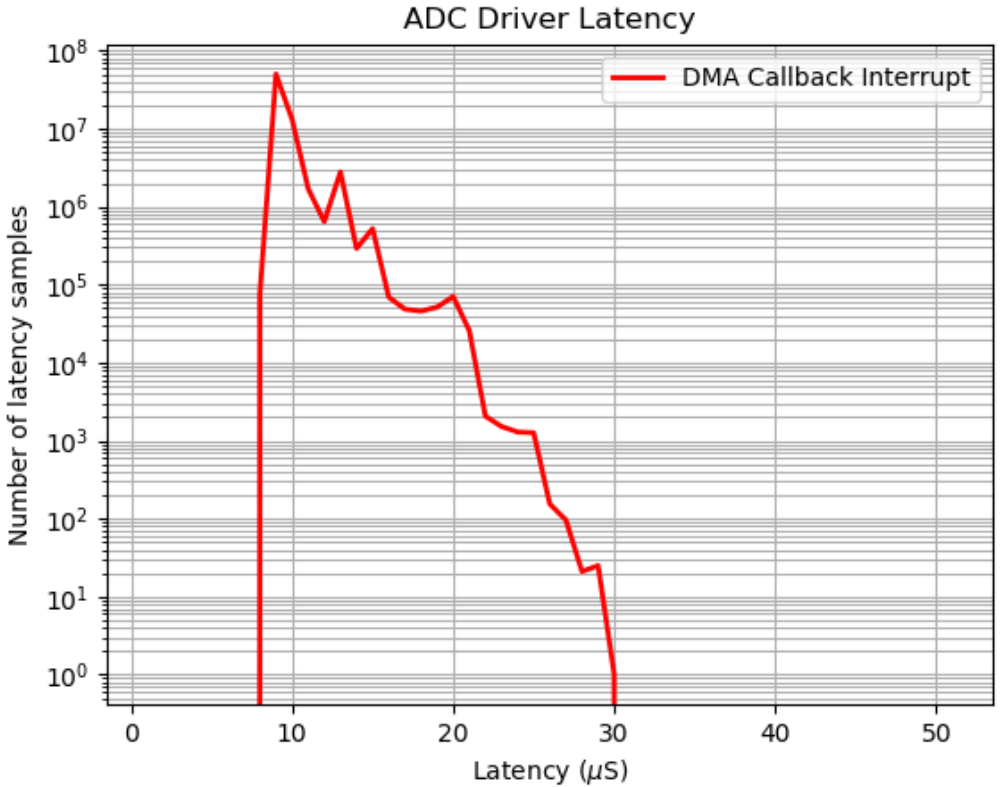
- Analog to Digital Converter Driver
  - Using Industrial I/O (IIO) subsystem
  - DMA Engine based
  - Performance captured using Hardware Timer

Hardware Configuration



IIO Driver DMA Interrupt Latency

- Maximum: 30 µS
- Mode: 9 µS



Timer Capture Function

- Hardware Timers with a capture function are common in SoCs and Microcontrollers
- Upon the trigger event the present value of the free running timer is stored to the Load Register
- In the kernel ISR (DMA Engine callback in this case) both the timer value as well as the stored Load Register from the event can be read and the latency calculated

Conculsions

- Cortex-A53 is a very low latency core
- Using the Programmable Logic to de-couple SPI bus is very effective

Future Work

- Investigate UDP Path Latencies
- Investigate difference between cyclictest and ADC driver results

Special Thanks

- Rajan Vaja from Xilinx
  - Helped us get up and running on the 4.18 kernel
- Enclustra
- My loving wife and family

Embedded Linux Conference Europe

OpenIoTSummit Europe

THE LINUX FOUNDATION