Research Platform for Quality of Time (QoT) Stack

Award # CNS-1329755 (UCLA), CNS-1329644 (CMU), CNS-1329644 (UCSD), and CNS-1329650 (UCSB)
Type: Frontier; Start Date: June 2014

Anh Luong (University of Utah)

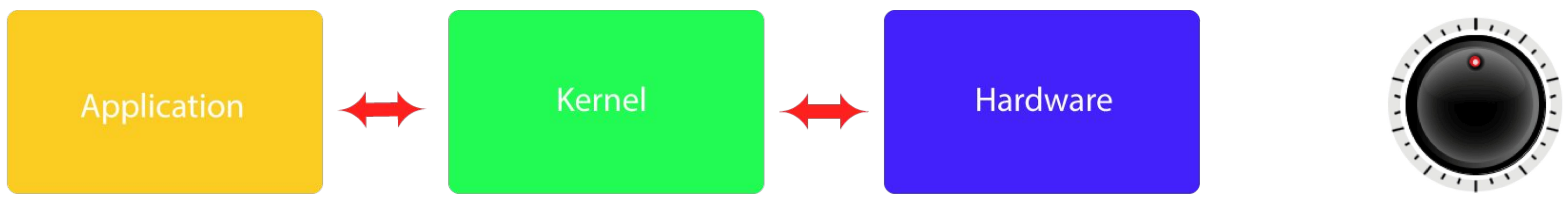
Co-Authors: Neal Patwari (University of Utah & Xandem), Thomas Schmid (Greina Technologies)

Motivation

- What are needs for Quality of Time (QoT) implementation?
  - **Knowing** time - *The current time is **x** with uncertainty σ*
  - **Keeping** time - *Interrupt after **x**, but no later than **x**±σ*
  - **Sharing** time - *Across networked devices*
  - **Controlling** time - *Adapt a local sense of time to balance resources*
- **Goal**: **Adaptable hardware with disciplinable clocks for the above**

Key Idea

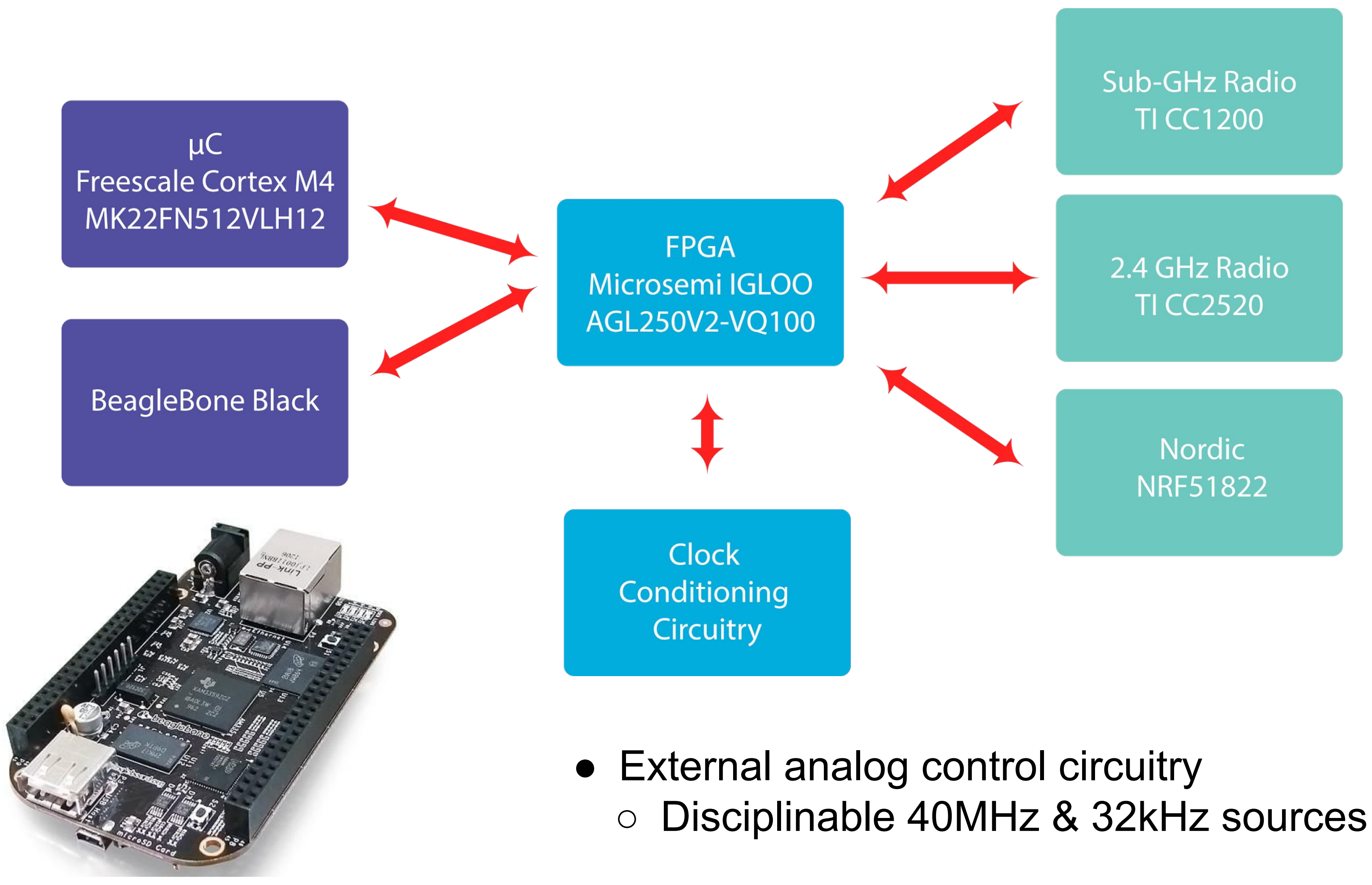
- Need a experimental testbed with “knobs” to test QoT performance
  - Frequency, Phase, Jitter, Clock Distribution, Radio, Processor
- Chronos: platform with fully **software disciplined clock network**



Architecture

- FPGA
  - Clock synthesis - generate common clock source for processor and radios
  - Software defined clock distribution - distribute clock to selected radio and processor
  - Roseline Control Module - control external clock conditioning circuitry, hardware interface for Linux Kernel

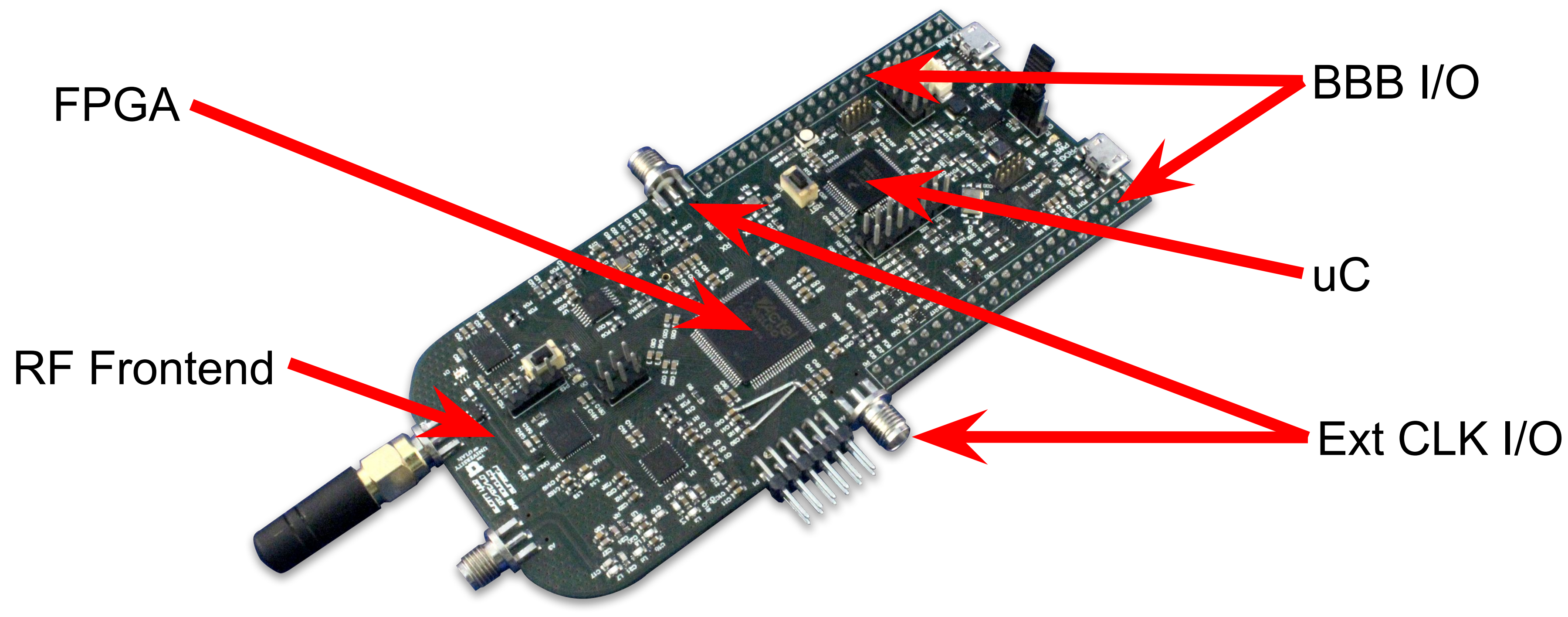
- Processors
  - Freescale
    - Cortex M4
    - **Easy to use**
    - **Low power**
    - Embedded
    - For experiments with low computation requirement
  - Beaglebone Black
    - Full Linux distribution
    - **Roseline QoT stack**
    - For state of the art time synchronization applications



- External analog control circuitry
  - Disciplinable 40MHz & 32kHz sources

- Radio
  - CC1200
    - Multiple selectable sub-GHz operating frequencies
    - Direct access to received signal samples
  - CC2520
    - Well-known standard Zigbee
  - nRF51822
    - Bluetooth BLE
    - Nordic Gazell 2.4GHz

Implementation



Current Research Directions

- **Improved synchronization methods**
  - Clock skew measurement via carrier offset
  - Clock offset measurement via fractional-delay estimation
- **Improved control of oscillators**
  - Frequency via FPGA Clock Conditioning Circuitry
  - Offset via FPGA Clock Delay
  - Jitter via FPGA Synthesized Roseline Clock Discipline Module
- **Improved linux hardware interface**
  - Controllable through Linux Kernel
  - Integration with Roseline QoT Stack