Multi-core programming frameworks for embedded systems

Kaushal Sanghai and Rick Gentile
Analog Devices Inc.,
Norwood, MA
Introduction

- To meet the growing processing demands placed by embedded applications, multi-core architectures have emerged as a promising solution.

- Embedded developers strive to take advantage of extra core(s) without a corresponding increase in programming complexity.

- Ideally, the performance increase should approach “N” times where “N” is the number of cores.

- Managing shared-memory and inter-core communications makes the difference!

- Developing a framework to manage code and data will help to speed development time and ensure optimal performance.

- We target compute intensive and high bandwidth applications on an embedded dual-core processor.
A high performance and low power dual-core embedded processor is selected as the target platform.

A dedicated Level 1 and a shared Level 2 memory, a DMA controller and a flexible peripheral interface are used efficiently to map the application data flow based on the data access granularity.

For the targeted multimedia applications, the data access pattern is studied based on the temporal and spatial locality to create a line, macro-block, frame and group of pictures (GOP) type frameworks.

To achieve a 2x speed-up, the frameworks combine techniques to efficiently manage the shared resources and exploit the known data access pattern in multimedia applications.