LDPC Decoding on the Intel SCC

Andreas Diavastos, Panayiotis Petrides, Gabriel Falcao†, Pedro Trancoso

Computer Science Department
University of Cyprus

†Department of Electrical and Computer Engineering
University of Coimbra

CASPER: Computer Architecture, Systems and Performance Evaluation Research
Motivation

Challenges

- Simple core architectures
- Limited access to external main memory
- Hardware cache-coherency?
- Programmability

How do we exploit the available parallelism in an efficient and effective way?
Outline

- Introduction
- Intel SCC Many-core
  - Top- and Tile-Level Architecture
  - Memory Hierarchy
- LDPC Decoding
- Parallel LDPC Decoding and Results
  - Distributed Parallel Decoder
  - Shared Parallel Decoder
  - Parallel Multi-codeword Decoder
- Conclusions
Outline

• Introduction
• Intel SCC Many-core
  + Top- and Tile-Level Architecture
  + Memory Hierarchy
• LDPC Decoding
• Parallel LDPC Decoding and Results
  + Distributed Parallel Decoder
  + Shared Parallel Decoder
  + Parallel Multi-codeword Decoder
• Conclusions
Introduction

- **Purpose**
  + Computation and communication intensive algorithm
  + Harvest the available parallelism

- **Proposal**
  + Three LDPC Decoding Implementations:
    - Distributed Parallel Decoder
    - Shared Parallel Decoder
    - Parallel Multi-codeword Decoder
Outline

• Introduction
• **Intel SCC Many-core**
  + Top- and Tile-Level Architecture
  + Memory Hierarchy
• LDPC Decoding
• Parallel LDPC Decoding and Results
  + Distributed Parallel Decoder
  + Shared Parallel Decoder
  + Parallel Multi-codeword Decoder
• Conclusions
Intel Single-chip Cloud Computer
Intel Single-chip Cloud Computer

Tile Architecture
Intel Single-chip Cloud Computer

Tile Architecture

Two P54C cores
  + 16KB L1 cache
  + 256KB L2 cache

16KB Message Passing Buffer (MPB)
  + 8KB / core

Traffic Generator
  + Mesh performance testing

Mesh Interface Unit (MIU)
  + Connect to Router
Intel Single-chip Cloud Computer

Memory Architecture

- Private off-chip
- Shared off-chip
- Shared on-chip
- L2 cache
Outline

• Introduction
• Intel SCC Many-core
  + Top- and Tile-Level Architecture
  + Memory Hierarchy
• LDPC Decoding
• Parallel LDPC Decoding and Results
  + Distributed Parallel Decoder
  + Shared Parallel Decoder
  + Parallel Multi-codeword Decoder
• Conclusions
LDPC Decoding†

- **Introduced in 1960s (Low-Density Parity-Check)**
  - Communication Standards (e.g. Satellite communication)
  - Storage Standards
- **Error Correcting Codes (ECC)**
  - Used for **error-free data transmission** through noisy and high error probability channels
- **Communication Impact**
  - Extremely high number of messages exchanged between nodes of the graphs for large dimension LDPC codes (e.g. the 64800-bit DVB-S2 satellite communication standard)
- **Tanner Graph**
  - Underlying data structure to support the decoding of LDPC codes
- **Decoding Graph**

LDPC Decoding

Decoding Graph

Check Nodes

Bit Nodes

Input Pn Probabilities

Message Updates

Output Corrected Bits

LDPC Decoding

Execution Flow

Parallel Update of Bit Nodes

Check Nodes

Core 0

CN0

Core 1

CN1

Core 2

CN2

... Core 47

CNn

Bit Nodes

BN0

BN1

...

BNm

Kernel 1 (Update of Bit Nodes)

LDPC Decoding†

Execution Flow

Kernel 2
(Update of Check Nodes)

Outline

• Introduction
• Intel SCC Many-core
  + Top- and Tile-Level Architecture
  + Memory Hierarchy
• LDPC Decoding
• Parallel LDPC Decoding and Results
  + Distributed Parallel Decoder
  + Shared Parallel Decoder
  + Parallel Multi-codeword Decoder
• Conclusions
Parallel LDPC Decoders on the Intel SCC

Distributed Parallel Decoder

WHILE decoding
+ Read data from your private memory
+ Calculate the update messages
+ Broadcast updates to all cores

Implementations
+ RCCE Broadcast (bcast)
+ RCCE send/receive (send/recv)
+ MPB Broadcast (fastbcast)
Parallel LDPC Decoders on the Intel SCC

Distributed Parallel Decoder Results

Normalized Throughput

Baseline Scenario:
Sequential execution on one SCC core using private memory

Scalability
Send/Recv: For small number of cores
Fastbcast: For large number of cores
Bcast: Too much runtime overhead
Parallel LDPC Decoders on the Intel SCC

Distributed Parallel Decoder Results

Communication dominates the execution!
Parallel LDPC Decoders on the Intel SCC

Shared Parallel Decoder

One Decoding Graph shared among all cores

- No message exchange needed
- All data updates are stored in the shared memory, thus visible to all cores
Parallel LDPC Decoders on the Intel SCC

Shared Parallel Decoder Results

Normalized Throughput

Scalability for up to 4-core configuration

Results are still not better than the baseline

Large number of off-chip shared memory accesses

Shared memory is not cached due to the lack of hardware cache-coherency
Parallel LDPC Decoders on the Intel SCC

Parallel Multi-codeword Decoder

One Decoding Graph per core
- Decoding graph stored in each cores private memory
- Each core used as a separate decoder
- SCC is simultaneously decoding 48 codewords
Parallel LDPC Decoders on the Intel SCC

Parallel Multi-codeword Decoder Results

Normalized Throughput

- Near to linear speedup for the **multi**
- Increase shared scalability using the **shared**
  - Replicate the best shared configuration (4-cores)
Outline

• Introduction
• Intel SCC Many-core
  + Top- and Tile-Level Architecture
  + Memory Hierarchy
• LDPC Decoding
• Parallel LDPC Decoding and Results
  + Distributed Parallel Decoder
  + Shared Parallel Decoder
  + Parallel Multi-codeword Decoder
• Conclusions
Conclusions

• Development of parallel LDPC decoders on the research processor Intel SCC

• Proposal of three different implementations to overcome limitations of the algorithm and the parallel architecture

• Almost linear speedup for the Parallel Multi-codeword Decoder
Thank You!

CASPER Group
University of Cyprus
Computer Architecture, Systems and Performance Evaluation Research
Visit us:
www.cs.ucy.ac.cy/~pedro/casper

Email: diavastos@cs.ucy.ac.cy

Acknowledgement:
We would like to thank Intel Labs for lending the Intel SCC research processor.