Programming of Massively Parallel Processor Arrays
Experiences & Opportunities

Zain-ul-Abdin, Ph.D.
Outline

• Background & Motivation
• Massively Parallel Processor Arrays
  – Example architectures
• Occam-pi Introduction
  – Overview of Compiler framework
• Autofocus case study
  – Mapping on two target architectures
• Evaluation Results
• Concluding Remarks

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
High-Performance Embedded Computing (HPEC) Challenges

Requirements:
- High Performance
- Energy Efficiency
- Adaptivity

Massively Parallel Processor Arrays

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
Ambric Platform

- 45 brics (360 processors) @300 Mhz
- Max Performance → 1 TOPS
- Max Power → 11 W
- Programming languages
  - aJava
  - aStruct
Adapteva-Epiphany

- Processing Cores → 16
- Max Clock rate → 1GHz
- Max Performance → 32 GFLOPS
- Max Power → 2 W

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
Programmability Challenge

"Programming of Massively Parallel Processor Arrays”, Zain-ul-Abdin
Our Approach

• Occam-pi Language
  – CSP dataflow
  – Explicit concurrency
  – Strong encapsulation

• Mobility features of pi-calculus
  – Expression of Reconfigurability

• Supported by a compiler for allowing portability

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
Occam-pi Code Example

SEQ

PROC octople (CHAN INT in?, out!)
CHAN INT a, b:

PAR

double (in?, a!)
double (a?, b!)
double (b?, out!)

SEQ

in ? sum
in ? x
sum := sum + x
out ! Sum

PAR

in0 ? a
in1 ? b
out ! a + b

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
Dynamic Process Invocation

\[
\text{PROC } A () \\
\ldots \text{ local state} \\
\text{SEQ} \\
\ldots \\
\text{FORKING} \\
\text{SEQ} \\
\ldots \\
\text{FORK } P(n, svr, cli) \\
\ldots \\
\ldots \\
:\]

\textbf{VAL} data is \textit{copied} into a \textit{FORK}ed process

\textbf{MOBILE} data and \textit{channel-ends} are \textit{moved} into a \textit{FORK}ed process

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
Occam-pi to MPPA Compilation

Occam-pi Code

Occam Frontend

ParseOccam

AST

Transformations

SimplifyTypes  SimplifyExpr  SimplifyProcs  Unnest

Ambric Backend

GenerateSOPM

Ambric aStruct, aJava, assembly

XPP Backend

GenerateNML

NML Code

P2012 Backend

GenerateNPM

NPM Code

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
Synthetic Aperture Radar System Data Collection

In-collaboration with SAAB

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
Auto-focus in SAR Systems

• Non-linear flight path causes defocusing
• Auto-focus can find the best compensation by iteratively matching the contributing images
Autofocus Criterion Calculation

\[ |f_r(r,\bar{r})|^2 + |f_r(r,\bar{r})|^2 \]  
(incl. interpolations, e.g. NN or cubic)
Autofocus Criterion Calculation - Mapping on Ambric

Design-I

Design-II

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
Autofous Criterion Calculation
Mapping on Epiphany

"Programming of Massively Parallel Processor Arrays", Zain-ul-Abdin
## Implementation Results

<table>
<thead>
<tr>
<th>Implementations</th>
<th>No. of Cores</th>
<th>Throughput (Pixels/sec)</th>
<th>Speedup</th>
<th>Power (Watts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autofocus-Design-I (Ambric@ 300 Mhz)</td>
<td>70(SRD), 24(SR)</td>
<td>236,386</td>
<td>13.3</td>
<td>6.52</td>
</tr>
<tr>
<td>Autofocus-Design-II (Ambric@ 300 Mhz)</td>
<td>141(SRD), 28(SR)</td>
<td>486,224</td>
<td>27.5</td>
<td>9.8</td>
</tr>
<tr>
<td>Autofocus-Sequential (Epiphany@ 1.0 Ghz)</td>
<td>1</td>
<td>17,668</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Autofocus-Parallel (Epiphany@ 1.0 Ghz)</td>
<td>13</td>
<td>192,857</td>
<td>10.9</td>
<td>2</td>
</tr>
</tbody>
</table>

Energy Efficiency = 2
Summary

• Identified the significance of Massively Parallel Processor Arrays
  + Throughput Cores + Reconfigurability
  + Real Performance advantage
  + Energy efficient through specialization
    – Non-legacy code compliant
    – Limited code base & library support available

• The Future
  + Simplified development based on concurrent programming model of occam-pi
  + Able to express dynamic reconfiguration with a formal basis
Thank you for your attention!

Email: Zain-ul-Abdin@hh.se