Associative Computing

- Associative computing is particularly well suited to processing records of data in a tabular format.
- As illustrated, each Processing Element (PE) of the SIMD associative computing array can store a record of this tabular data in its memory.

<table>
<thead>
<tr>
<th>Student Name</th>
<th>ID</th>
<th>Grade</th>
<th>Search</th>
<th>STEP1 Mask</th>
<th>STEP2 Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>07</td>
<td>66</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Gary Heath</td>
<td>05</td>
<td>95</td>
<td>1 1</td>
<td>1 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Peter Smith</td>
<td>11</td>
<td>87</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>John Smith</td>
<td>04</td>
<td>76</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Terry Stanley</td>
<td>02</td>
<td>100</td>
<td>1 1</td>
<td>1 0</td>
<td>1 0</td>
</tr>
<tr>
<td>Will Hanson</td>
<td>01</td>
<td>84</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Jane Antony</td>
<td>06</td>
<td>94</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Mark Briggs</td>
<td>13</td>
<td>86</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Cliff Patel</td>
<td>09</td>
<td>76</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Mir Lee</td>
<td>19</td>
<td>83</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Goby Carson</td>
<td>03</td>
<td>83</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Gillian Roger</td>
<td>08</td>
<td>26</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
</tbody>
</table>

Implementing Associative Computing in the ASC Processor

- **Associative search**: the Control Unit broadcasts the search key to all PEs to compare with local memory. If search is successful, those PEs are designated *responders*, and they set their Responder bit and the top of their Mask Stack to ‘1’.
- **Process the responders sequentially**: STEP instruction uses Responder Resolution Unit and Mask Stack to process responding PEs one by one.
- Searching for maximum/minimum value in a field uses Falkoff Algorithm, process bit slices from left to right.

<table>
<thead>
<tr>
<th>Student Name</th>
<th>ID</th>
<th>Grade</th>
<th>Search Mask</th>
<th>STEP1 Mask</th>
<th>STEP2 Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>07</td>
<td>66</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Gary Heath</td>
<td>05</td>
<td>95</td>
<td>1 1</td>
<td>1 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Peter Smith</td>
<td>11</td>
<td>87</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>John Smith</td>
<td>04</td>
<td>76</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Terry Stanley</td>
<td>02</td>
<td>100</td>
<td>1 1</td>
<td>0 1</td>
<td>1 0</td>
</tr>
</tbody>
</table>

Table of Contents

- Chapter 1 – Introduction
- Chapter 2 – First Prototypes of an Associative Computing (ASC) Processor
- Chapter 3 – A Scalable Pipelined ASC Processor With Reconfigurable PE Interconnection Network
- Chapter 4 – A Specialized ASC Processor with Reconfigurable 2D Mesh for Solving the Longest Common Subsequence (LCS) Problem
- Chapter 5 – An ASC Processor to Support Multiple Instruction Stream Associative Computing (MASC)
- Chapter 6 – Conclusions and Future Work
In the following slides I present some applications of our processor.

Relational Database Processing: O(|B|)

Intersection, Union, Cartesian Product and Join are basic operations in Database processing. Using associative Search and STEP operations, we can achieve much faster processing time.

Image Processing (Edge Detection Using Convolution)

Table of Contents

- Chapter 1 – Introduction
- Chapter 2 – First Prototypes of an Associative Computing (ASC) Processor
- Chapter 3 – A Scalable Pipelined ASC Processor With Reconfigurable PE Interconnection Network
- Chapter 4 – A Specialized ASC Processor with Reconfigurable 2D Mesh for Solving the Longest Common Subsequence (LCS) Problem
- Chapter 5 – An ASC Processor to Support Multiple Instruction Stream Associative Computing (MASC)
- Chapter 6 – Conclusions and Future Work
Implementing 1-D and 2-D PE Interconnection Network

This version of ASC processor supports both a 1-D and 2-D PE interconnection network for those applications that require a network.

The network is implemented as a large 8xN bit wide NWIN register (where N is the number of PEs), an 8xN bit NWOUT register.

Data enters the network through the NWIN register, which stores data for PE j in bits from 8j to 8j+7, and then that data is routed to the proper place in the NWOUT register.

Table of Contents

- Chapter 1 – Introduction
- Chapter 2 – First Prototypes of an Associative Computing (ASC) Processor
- Chapter 3 – A Scalable Pipelined ASC Processor With Reconfigurable PE Interconnection Network
- Chapter 4 – A Specialized ASC Processor with Reconfigurable 2D Mesh for Solving the Longest Common Subsequence (LCS) Problem
- Chapter 5 – An ASC Processor to Support Multiple Instruction Stream Associative Computing (MASC)
- Chapter 6 – Conclusions and Future Work
ASC Processor’s Pipelined Architecture

- I have implemented a scalable pipelined SIMD Associative (ASC) Processor using Altera FPGAs
  - Field Programmable Gate Arrays (FPGAs) are typically used for designs and can be thought of as programmable hardware
- Five single-clock-cycle pipeline stages are split between the SIMD Control Unit (CU) and the PEs
  - In the Control Unit
    - Instruction Fetch (IF)
    - Part of Instruction Decode (ID)
  - In the Scalar PE (SPE), in each Parallel PE (PPE)
    - Rest of Instruction Decode (ID)
    - Execute (EX)
    - Memory Access (MEM)
    - Data Write Back (WB)

Pipelined ASC Processor’s Performance

- Our pipelined ASC Processor has been implemented on an Altera APEX20KC1000 FPGA with 70 8-bit PEs
  - Other 8-bit processor cores implemented on this FPGA / speed grade have clock speeds ranging from 30 to 106 MHz, typically 60-68 MHz
- Our pipelined ASC Processor has a clock speed of 56.4 MHz, comparable with these other processors
  - With the 5-stage pipeline, our ASC Processor can approach a peak performance of 300 MHz
Reconfigurable PE Interconnection Network

- Our pipelined ASC Processor also has a reconfigurable PE interconnection network.
- Reconfigurable PE network supports associative computing by allowing arbitrary PEs in the PE Array to be connected via
  - Linear array (currently implemented), or
  - 2D mesh (shown in the next chapter)
- Each PE in the PE Array can choose its own connectivity
  - Responders choose to stay in the PE interconnection network, and
  - Non-Responders choose to stay out of the PE interconnection network, so that they are bypassed by any inter-PE communication

Reconfigurable Network Implementation

- Data switch
  - Passes register, broadcast, and immediate data to the PE and to its two neighbors
  - Routes data from the PE’s neighbors to its EX stage
- Reconfigurable network — supports Bypass Mode to remove the PE non-responders from the network
  - Will be needed by MASC Processor

ASC Processor’s Network Performance

- Performance of ASC Processor degrades as number of PEs is increased with Bypass Mode present
  - Due to the long path from the first PE to the last PE in the PE array
- 4-PE ASC Processor requires 2152 LEs and runs at 56.4 MHz with Bypass Mode present
  - When the number of PEs is increased to 50, the clock frequency drops to 22 MHz
- In the future, this delay may be reduced using a pipelined or other multi-hop architecture
Table of Contents

- Chapter 1 – Introduction
- Chapter 2 – First Prototypes of an Associative Computing (ASC) Processor
- Chapter 3 – A Scalable Pipelined ASC Processor With Reconfigurable PE Interconnection Network
- Chapter 4 – A Specialized ASC Processor with Reconfigurable 2D Mesh for Solving the Longest Common Subsequence (LCS) Problem
- Chapter 5 – An ASC Processor to Support Multiple Instruction Stream Associative Computing (MASC)
- Chapter 6 – Conclusions and Future Work

Overview of LCS Algorithm

- Given two strings, find the LCS common to both strings
  - Example:
    - String 1: AGACTAGGTA
    - String 2: ACTGAG
      - AGACTGAGGTA
      - -ACTGAG - -
      - -ACTGA - G-
      - A- -CTGA - G-
      - A- -CTGAG - -
  - The time complexity of this algorithm is clearly O(nm)

PE’s Form Coteries

5 x 5 coterie network with switches shown in "arbitrary" settings. Shaded areas denotes coterie (the set of PEs Sharing same circuit)
**Reconfigurable Network in the ASC Processor**

- Key to reconfigurability is the Data Switch inside each PE:
  - The Data Switch is expanded to connect to its four neighbors (N-E-S-W) to form a 2D Reconfigurable Network
  - Data switch has bypass mode to allow PE communication to skip non-responders, so as to support associative computing

![DATA Communication Diagram](image1)

**LCS Algorithm on Reconfigurable 2D Mesh**

![LCS Algorithm Diagram](image2)

**Table of Contents**

- Chapter 1 – Introduction
- Chapter 2 – First Prototypes of an Associative Computing (ASC) Processor
- Chapter 3 – A Scalable Pipelined ASC Processor With Reconfigurable PE Interconnection Network
- Chapter 4 – A Specialized ASC Processor with Reconfigurable 2D Mesh for Solving the Longest Common Subsequence (LCS) Problem
- Chapter 5 – An ASC Processor to Support Multiple Instruction Stream Associative Computing (MASC)
- Chapter 6 – Conclusions and Future Work

**MASC Architecture**

MASC is an MSIMD (multiple SIMD) version of ASC that supports multiple Instruction Streams (ISs)

In our dynamic MASC Processor, tasks are assigned to available ISs from a common pool as those ISs become available

![MASC Architecture Diagram](image3)
1. Initially, IS0 is executing the program
2. When a conditional operation is encountered, IS0 transfers program control to TM0 (first available TM in the TM pool)
3. TM0 allocates Task0 to IS0
4. TM0 allocates Task1 to IS1 (first available IS in the IS pool)
5. TM0 waits for Task0 and Task1 to finish
6. [ IS0 and IS1 perform lines 1 through 5 as necessary ]
7. After both Task0 and Task1 finish, TM0 transfers program control back to IS0

---

**PE Structure**

- A dedicated CU Selector unit is added to the IE stage of each Parallel PE
- The CU Selector is a large multiplexer choosing which TM or IS will control the PE and to which broadcast network the PE should listen

---

**Instruction Stream Tree Structure**

The implementation can easily be scaled up as necessary. The number of TMs limits the degree of nesting, and the number of ISs limits the number of tasks executed in parallel.
Contributions

- My first contribution was to construct the first working ASC Processor and the first scalable ASC Processor [WMPP'04]
- My second contribution was to develop and implement a scalable pipelined ASC Processor [PDCS'05]
- My third contribution was to build on the reconfigurable PE interconnection network developed in conjunction with my pipelined ASC processor to develop a specialized version of this processor designed to support an innovative LCS algorithm that is useful in bioinformatics such as genome sequence comparison [PDCS'06 #1]
- My fourth and final contribution was to develop a first prototype of the MASC Processor, a MSIMD version of ASC to better support control parallelism and increase PE utilization [PDCS'06 #2]

Future Work

- One area of future work would be to explore more computationally-intensive associative computing algorithms that could benefit from this pipelined ASC Processor
- Another area for future research would be to further explore the use of the specialized ASC Processor for genome sequence comparison.
- Yet another area for future research would be to explore the use of this first MASC Processor for various applications