Reinventing a parallel machine from the past

Andrés Amaya García

aa1399@bristol.ac.uk
About me…

- Andrés Amaya García
- Graduated from the University of Bristol in 2015
- MEng Computer Science
"Once upon a time, nine (rainy) months ago...
The Transputer

OpenTRANSPUTER.org
Transputer applications
Transputer applications

Amstrad TV set-top box contains ST20
Transputer applications

Amstrad TV set-top box contains ST20

HETE-2 contains T805 Transputers
Project objectives
Project objectives

- Open-source.
Project objectives

- Open-source.
- Supports Transputer instruction set.
Project objectives

- Open-source.
- Supports Transputer instruction set.
- Different micro-architecture.
The Internet of Things (IoT) is all about connectivity!
Project objectives

- Open-source.

- Supports Transputer instruction set.

- Different micro-architecture.
Project objectives

- Open-source.
- Supports Transputer instruction set.
- Different micro-architecture.
- New external communication mechanism and I/O interface.

OpenTRANSPUTER.org
How does the Transputer work?

Open TRANSPUTER.org
Occam

- Occam is a high-level programming language developed at Inmos hand-in-hand with the Transputer.
- Explicit concurrency and interprocess communication.
Occam

- Occam is a high-level programming language developed at Inmos hand-in-hand with the Transputer.
- Explicit concurrency and interprocess communication.
The Devil is in the detail
The Devil is in the detail
Interprocess communication

- Special Transputer instructions implement Occam primitives efficiently.
- Communication performed either through channel in memory or physical links.
But I want to hear about the OpenTransputer!
Processor components

- 16 input link controllers
- OpenTransputer CPU
- Output link controllers
- Memory
- 15 I/O pin handlers

OpenTRANSPUTER.org
OpenTransputer

16 input link controllers

OpenTransputer CPU

Output link controllers

Memory

15 I/O pin handlers

Processor components
OpenTransputer

16 input link controllers

OpenTransputer CPU

Output link controllers

Memory

15 I/O pin handlers
OpenTransputer

16 input link controllers

Output link controllers

OpenTransputer CPU

Memory

15 I/O pin handlers

Processor components

OpenTRANSPUTER.org
CPU

- Control Unit
- Fetch Unit
- Memory
- Datapath (Register file, AU, LU, Memory addressing, etc.)
CPU

- Control Unit
  - Fetch Unit
  - Memory
  - Datapath (Register file, AU, LU, Memory addressing, etc.)
CPU

Control Unit

Fetch Unit

Memory

Datapath
(Register file, AU, LU, Memory addressing, etc.)
CPU

Control Unit

Fetch Unit

Memory

Datapath
(Register file, AU, LU, Memory addressing, etc.)
Control unit

- Microcoded control unit.
- Control signals stored in Read-Only Memory (ROM).
- Area savings and potentially less complex.
- Designed with a microcode strategy.
- Control signals generated by hardwired logic.
- Potentially faster than ROM.
Control unit

Microinstructions (human-readable)
Control unit

- Microinstructions (human-readable)
- Verilog HDL (Not so human-readable)

Run tools
Control unit

Microinstructions (human-readable)

Run tools

Verilog HDL (Not so human-readable)

Integrate

Control Unit
OpenTransputer microinstructions

CJ0  ROMp0(0)
    CmpconstfromA
    Condall(CJ1,CJ2);

CJ1  AfromB
    BfromC
    OfromClear
    Gotoplus1;
Control Unit

Fetch Unit

Memory

Datapath
(Register file, AU, LU, Memory addressing, etc.)

CPU
Inmos Transputer datapath
OpenTransputer datapath

‘Wider’ datapath

OpenTRANSPUTER.org
OpenTransputer datapath

More parallelism
OpenTransputer datapath

Shadow registers

Write
Read
Register
File
Write
Write
Write

Address from microinstruction
Addr
DataOut
ROM const

Address from microinstruction
Addr
DataOut
ROM const

LU

DataIn
DataOut

Word Address

Memory

Word Address

Byte Address

Block Move Unit

OpenTRANSPUTER.org
OpenTransputer
CPU

16 input link controllers

Output link controllers

Memory

15 I/O pin handlers

Processor components

OpenTRANSPUTER.org
External communication

Inmos Transputer  OpenTransputer

OpenTRANSPUTER.org
Autonomous link controllers

- There are 16 input ports and a single output port.
- Extended the original Transputer controllers to support virtual channels allowing an arbitrary number of processes to be queued to perform output operations.
Message routing

Initial address

| Layer: 0 | RTE: 10 | RTC: 00 |
| Layer: 0 | RTE: 10 | RTC: 00 |
| Layer: 1 | RTE: 10 | RTC: 00 |
| Layer: 2 | RTE: 10 | RTC: 00 |
| Layer: 3 | RTE: 10 | RTC: 00 |

OpenTransputer (sender)

OpenTransputer (receiver)

Turning point

Route towards edge (RTE)
Route towards core (RTC)
Message routing

Initial address

Layer: 3
RTE: 10
RTC: 00

Layer: 2
RTE: 10
RTC: 00

Turning point

Layer: 1
RTE: 10
RTC: 00

Layer: 0
RTE: 10
RTC: 00

Route towards edge (RTE)
Route towards core (RTC)
Message routing

Initial address

Layer: 3
RTE: 10
RTC: 00

Layer: 2
RTE: 10
RTC: 00

Layer: 1
RTE: 10
RTC: 00

Layer: 0
RTE: 10
RTC: 00

Turnig point

OpenTransputer (sender)

OpenTransputer (receiver)

Route towards edge (RTE)
Route towards core (RTC)
Message routing

Initial address

Layer: 3
RTE: 10
RTC: 00

Layer: 2
RTE: 10
RTC: 00

Layer: 1
RTE: 10
RTC: 00

Layer: 0
RTE: 10
RTC: 00

Layer: 0
RTE: 10
RTC: 00

Turning point

Route towards edge (RTE)
Route towards core (RTC)
Message routing

Initial address

<table>
<thead>
<tr>
<th>Layer: 0</th>
<th>RTE: 10</th>
<th>RTC: 00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer: 1</td>
<td>RTE: 10</td>
<td>RTC: 00</td>
</tr>
<tr>
<td>Layer: 2</td>
<td>RTE: 10</td>
<td>RTC: 00</td>
</tr>
<tr>
<td>Layer: 3</td>
<td>RTE: 10</td>
<td>RTC: 00</td>
</tr>
</tbody>
</table>

Open Transputer (sender)

Open Transputer (receiver)

Turning point

Route towards edge (RTE)
Route towards core (RTC)
OpenTransputer

CPU

16 input link controllers

Output link controllers

OpenTransputer CPU

Memory

15 I/O pin handlers
OpenTransputer I/O interface

- I/O pins expose the same interface as communication links.
- Introduced an instruction (confio) that can be used to configure the I/O pins.
So… what about performance?
Synthesis

Synthesised design for both the ZedBoard XC7Z020-CLG484 FPGA and a 180nm manufacturing process.
## Comparing synthesis results

<table>
<thead>
<tr>
<th></th>
<th>Inmos Transputer</th>
<th>OpenTransputer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Area</strong></td>
<td>64 mm²</td>
<td>3.69 mm²</td>
</tr>
<tr>
<td><strong>Manufacturing technology</strong></td>
<td>1000 nm</td>
<td>180 nm</td>
</tr>
</tbody>
</table>
# Comparing synthesis results

<table>
<thead>
<tr>
<th></th>
<th>Inmos Transputer</th>
<th>OpenTransputer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Area</strong></td>
<td>64 mm²</td>
<td>3.69 mm²</td>
</tr>
<tr>
<td><strong>Manufacturing technology</strong></td>
<td>1000 nm 5.6</td>
<td>180 nm</td>
</tr>
</tbody>
</table>

[OpenTRANSPUTER.org](https://opentransputer.org)
Comparing synthesis results

<table>
<thead>
<tr>
<th></th>
<th>Inmos Transputer</th>
<th>OpenTransputer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Area</strong></td>
<td>64 mm²</td>
<td>3.69 mm²</td>
</tr>
<tr>
<td><strong>Manufacturing technology</strong></td>
<td>1000 nm</td>
<td>180 nm</td>
</tr>
</tbody>
</table>

OpenTRANSPUTER.org
Comparing cycle counts

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Inmos Transputer</th>
<th>OpenTransputer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ld1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>startp</td>
<td>12</td>
<td>3-5</td>
</tr>
<tr>
<td>endp</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>move</td>
<td>8 + 2w*</td>
<td>6 + 5w*</td>
</tr>
</tbody>
</table>

*w is the number of words to copy.
Conclusion

- OpenTransputer based on the Transputer architecture.
- Different micro-architecture to take advantage of modern manufacturing technologies
- New external communication mechanism.
- New I/O interface.
Conclusion

- OpenTransputer based on the Transputer architecture.
- Different micro-architecture to take advantage of modern manufacturing technologies.
- New external communication mechanism.
- New I/O interface.

Thank you!

Andrés Amaya García
aa1399@my.bristol.ac.uk