A manycore space computer with FPGA co-processing

Brandon Beresini
beresini@maxentric.com

Scott Ricketts
sricketts@maxentric.com

Michael B. Taylor
michael.b.taylor@angstromush.com
Rad-hard Unified Scalable Heterogeneous Architecture (RUSH)
Today’s talk

• Motivation
• RUSH Architecture
• Preliminary Results
Promising candidates for next-generation space computing

- **CPU**
  - e.g. RAD750

- **Manycore**
  - e.g. Maestro

- **FPGA**
  - e.g. Virtex 5QV

- **ASIC**
  - e.g. OPERA RHBD

- Higher Performance, Better Power efficiency
- Greater Programmability, Lower NRE Costs
Manycore (or tiled) architectures offer programmability and scalability

- Array of general purpose tiles
- Low latency between tiles
- Coherent shared memory
- Message passing for efficient sync
Mesh networks manage intra-chip communication

Dynamic Networks give access to all of the fundamental resources of the architecture

- Tiles
- L1, L2 Caches
- DRAM
- I/O Devices

[Malone, MAPLD 2009]
Running Example: Maxentric’s Manycore Software Defined Radio
An optimal processing platform might have a mix of these:

- **CPU**
  - e.g. RAD750

- **Manycore**
  - e.g. Maestro

- **FPGA**
  - e.g. Virtex 5QV

- **ASIC**
  - e.g. OPERA RHBD

**Greater Programmability, Lower NRE Costs**

**Higher Performance, Better Power efficiency**
We might want to offload the Viterbi decoder to an FPGA.
Integration of heterogeneous processing systems is challenging

<table>
<thead>
<tr>
<th>Hardware integration</th>
<th>Providing access across chip boundaries to architectural resources</th>
<th>e.g. physical layer protocols, access to memory systems on CPUs, logic cores on FPGAs, etc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software integration</td>
<td>Developing software that uses the correct semantics and interfaces for cross-chip communication</td>
<td>e.g. message formats, synchronization, data streaming model, etc</td>
</tr>
</tbody>
</table>
Integration requires exposing the underlying architectural resources of the two chips in the system

- Tile
- Memory System
- I/O

Manycore

• IP Blocks
• I/O

FPGA
Tiled architectures like Maestro have mechanisms to handle these challenges

<table>
<thead>
<tr>
<th>Hardware integration</th>
<th>Expose memory systems, tiles and I/O to I/O devices through packet-based mesh network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software integration</td>
<td>Provide message passing, DMA, and shared memory as simple and powerful models for data movement and synchronization</td>
</tr>
</tbody>
</table>
RUSH applies the tiled model across the heterogeneous system.

Diagram:

- **7x7 Tile Array**
- **Manycore**
- **FPGA**

Each tile (VT-0 to VT-8) contains an IP block.
Integrating Maestro and an FPGA (SDR Example continued)
RUSH places the FFT and Viterbi blocks into virtual tiles

Maestro

FPGA

Virtual Tile

FFT Block

Virtual Tile

Viterbi Block

I/O Port

WiMAX Rx

WiMAX Tx

MAC

Linux

MAUI

Virtual Tile

FFT Block

Viterbi Block

Maestro

FPGA

Virtual Tile

FFT Block

Viterbi Block

I/O Port

WiMAX Rx

WiMAX Tx

MAC

Linux

MAUI

Virtual Tile

FFT Block

Viterbi Block

I/O Port

WiMAX Rx

WiMAX Tx

MAC

Linux

MAUI

Virtual Tile

FFT Block

Viterbi Block

I/O Port

WiMAX Rx

WiMAX Tx

MAC

Linux

MAUI

Virtual Tile

FFT Block

Viterbi Block

I/O Port

WiMAX Rx

WiMAX Tx

MAC

Linux

MAUI
RUSH places the FFT and Viterbi blocks into virtual tiles.
A tile can add a data cache that is coherent with the cross-chip shared memory system
FPGA-side prebuilt blocks provide integration with RUSH communication idioms.
Evaluation Methodology

• Application: Software Defined WiMAX Transceiver
  – Frame size: 2048
  – 40Mbits/s QPSK at 11k frames/s
  – K=7, ½-rate encoder

• Analyzed complexity for demanding kernels
  – FFT
  – Viterbi

• Estimated resource allocation for kernels
Estimates of potential efficiency gains

- FFT (8.2% vs. 1.2%)  6.8X
- Viterbi (44.9% vs. 3.5%)  12.8X

Maestro

Virtex 5QV
Future Work: Extending beyond kernels to systems of systems
Conclusion

• Virtual tiles and the cross-chip mesh network apply the tiled model across the heterogeneous system
• Applying the tiled model creates a unified heterogeneous architecture
• The unified architecture enables developers create high performance and power efficient systems
Cross-Chip Communication: FPGA Side

- XAUI Transceiver
- Packet Reshaper
- Clock-Xing FIFO
- IDIOM BLOCK
- IP BLOCK

10 Gbps XAUI connection to Maestro

64b @156.25 MHz

User clock frequency
Cross-Chip Communication: Tilera Side

• Since FPGA is configurable, Tilera side is relatively more important for Phase I
• We prototyped connection with XAUI CX4 Loop-Back cable on TileExpressPro-20G
• Subsequent slides explain Tilera side comm.
Tilera Side Communication

TilePro64

Send Gateway Tile

CX4 Cable (XAUI)

Receive Gateway Tile
Tilera Side Communication

TilePro64

CX4 Cable (XAUI)

xgbe/0

xgbe/1

2,6

3,4
Tilera Side Communication

TilePro64

CX4 Cable (XAUI)

xgbe/0

xgbe/1

2,6

3,4
Tilera Side Communication

TilePro64

CX4 Cable (XAUI)
Tilera Side Communication

TilePro64

CX4 Cable (XAUI)

xgbe/0

xgbe/1

3,4

2,6
Tilera Side Communication

TilePro64

3,4
2,6

xgbe/0
xgbe/1

CX4 Cable (XAUI)
Tilera Side Communication

TilePro64

CX4 Cable (XAUI)
xgbe/0

xgbe/1
Round Trip XAUI latency XAUI Loopback cable
Round Trip Latency: 440 ns

Good Reason to Expect Latency will be Similar for FPGA<->Tilera

440 ns (308 cycles @ 700 MHz)
Throughput Test Setup

TilePro64

xgbe/0

xgbe/1

CX4 Cable (XAUI)
Throughput Test Setup

TilePro64

10 Gbps Link

CX4 Cable (XAUI)
XAUI communication at 10 Gbps
Small Packets

RUSH XAUI Throughput

Packet Size (Words)

Throughput (Gbps)

Fixed transmit time
Large Packets

RUSH XAUI Throughput

Marginal transmission rate of 10 Gbps
XAUI communication

• Latency: 440ns (308 cycles at 700MHz)
• Throughput: Approaching 10 Gbps
  – minimum underlying XAUI packet size
    • 21 words; 15 words are data payload
    • 1 to 15 words of data are the same cost
• Stability: 24+ hr run
  – rt latency packet latency varies by <=1 cycle
• Reliability: 24+ hr rur
  – 0 data errors