A Milliflow Aggregation Processor

Bapi Vinnakota
Objectives

- Milliflow aggregation systems
  - Voice and other edge systems
- SoC architecture
  - High-level features
  - System dataflow
- Component architecture
  - Custom cores
  - Packet interfaces
- Results
  - Physicals
  - Performance
- Conclusion
Milliflow Aggregation Systems

- Milliflow aggregation nodes
  - Aggregate thousands of *milliflows*
  - The bandwidth of a milliflow is less than one thousandth of the bandwidth of the aggregate bandwidth

- Stringent constraints
  - Real time flows
  - Small packet sizes
  - System power dissipation is limited

- Custom Milliflow Aggregation Processor (Magpie) for VoIP systems
  - Experimental prototype chip, not designed for production
Magpie Architecture

16MB SRAM
- External Memory Interface
- DSP Bus Interface
- Data Cache
- RISC Processor Scheduler
- Instruction Cache

512 kbytes On-chip SRAM
- High Speed Internal Bus with Distributed DMA
- Packet Core Egress
- Program Memory

256MB SDRAM
- External Memory Interface
- Utopia Interface
- PCI Interface
- Data Cache
- RISC Processor Network Stack
- Program Memory
- Instruction Cache

Digital Enterprise Group Architecture and Planning
Magpie High-Level Description

- **Key Functionality**
  - VoIP (RFC1489), VoATM (I366.2), VolPoATM (RFC1483)
  - RTP(RTCP)/UDP/IP, RTP/AAL2 muxing, AAL5, TCP/IP
  - Jitter buffer algorithms and statistics support

- **Performance**
  - OC-3 5-ms, 2*OC-3 10 ms, OC-12 20 ms

- **Functional Interfaces**
  - 32-bit 66 MHz PCI (v.2.1)
  - 16-bit 50 MHz Utopia Level II
  - 32-bit 33 MHz VxBus (functionally glueless interface to IXS farm)

- **Memory Interfaces**
  - 64-bit SDRAM interface up to 256MB
  - 32-bit ZBT SSRAM interface up to 16MB

- **Architecture**
  - Three packet engines, two control cores, DMA engines
Packet Interface to DSP Farm Data Packet Flow

**SDRAM**

Jitter Buffer (part of SDRAM)

- **Injection Interface** monitors status of 4K flows in hardware.

- **Traffic core** identifies flows to serviced. Uses state data to release payload and/or blank packet.

- **Ingress core** fetches flow state data from local memory. Stores payload in DRAM. All ops in software.

- **Packet data** aggregated and streamed by interface.

**Diagram Notes:**
- **Internal SRAM (GMEM)**
- **Packet Interface to DSP Farm Data Packet Flow**
- **DMA** interfaces for data transfer.
DSP Farm to Packet Interface Data Packet Flow

Interface polls, extracts packets from DSP farms and aggregates into memory.

Inverse Jitter Buffer (part of SDRAM)

DMA creates space between packets to enable in-place header updates.

SDRAM

Internal SRAM (GMEM)

VxBus

Packet Core Egress

DMA

Egress core creates WAN headers, updates state data and sends packet data to backplane.

NIF

MAC
Packet Engine Architecture

RISC-based core with custom instructions for functional acceleration
## Instruction Set Class/ Application Matrix

<table>
<thead>
<tr>
<th></th>
<th>Flow Assoc.</th>
<th>Payload Ops.</th>
<th>Header Ops</th>
<th>Shaping Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Flow</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Bit Manip.</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Packet L &amp; A</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>In Memory</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Each class of instructions is applicable to multiple flow processing operations.
Packet Engine ISA

• Bit Manipulation
  – Header explosion/aggregation, Endian rearrange, Bit insert/extract

• Packet Logic and Arithmetic Operations
  – Checksums, CRCs, In-place addition, Reg. controlled adds, In-place pattern matching, Bit counts, Shift-logic operations

• Memory Operations
  – In-memory copy, Tree/List/Pointer manipulation, Multi-register loads, Byte-aligned operations

• Program flow control
  – Predicate-based branches, Hardware loopds, Variable-count loops, Multi-index loops, Multi-way decisions

• Multi-instruction links
  – Coordinate control/data instructions
ISA Applications

• Flow association
  – CAM emulation, Hash functions, Heap-based look-up, Table look-up, Tree/List-based look-up

• Payload operations
  – Packetization/Cellification, AAL2 SAR/RTP Muxing, Payload encoding/encryption, Jitter buffer insertion

• Header Operations
  – RTP/IP/UDP/ATM Header parsing/assembly, Checksum/CRC calculation/verification, Field updates

• Traffic shaping, flow control
  – Jitter buffer extraction, Calendar queuing, Statistics updates, GCRA algorithms, Hierarchical queuing
ISA Application Examples

- Checksum for IP/UDP
  - ELOOP // Identifies number of bytes to be loaded
  - LST EPLD // Loads data into register file
  - SMAD // Single cycle checksum up to 16 bytes
- Tree traversal for flow association
  - ELOOP // Controls tree traversal
  - LST ETREE // Looks for match/follows appropriate branch/NOPs
- Software SAR for AAL5/AAL2 (I363.2, I366.2, RFC 1483)
  - VLOOP // Identifies segment for SAR
  - LST EMCPY // Copies over data
- ATM Header extraction
  - EEXTR // Explodes target data into up to 5 registers
Distributed DMA engine designed to offload routine data movement tasks
Aggregates packets to reduce data movement overhead
DSP Interface Functions

DSP interface integrates per-packet DSP push/pull and polling with QoS support.
Network interface presents multiple external interface in common format
Aggregates small packets to reduce data transport overhead
Absorbs network jitter
Packet Engine Application Architecture

Pipelined data flow enables compute/movement overlap
Memory sized to support multiple frames of data in flight in each engine simultaneously
# Magpie Physicals

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>0.18µ</td>
</tr>
<tr>
<td>Packet Core Gates/Size</td>
<td>~1M Gates/~7.5mmx4.5mm</td>
</tr>
<tr>
<td>Gate/Transistor Count/Die Size</td>
<td>~10M/~70M/~15.5mmx15.5mm</td>
</tr>
<tr>
<td>Clock Rate/ Power</td>
<td>125MHz/4W/~2mW/flow</td>
</tr>
<tr>
<td>Package</td>
<td>625-pin PBGA</td>
</tr>
</tbody>
</table>
## Magpie Performance

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target throughput</td>
<td>2016 Flows</td>
</tr>
<tr>
<td>Per flow data rate/packet rate</td>
<td>64 Kbps/128 Kbps/200 pps</td>
</tr>
<tr>
<td>Aggregate data/packet rate</td>
<td>155 Mbps/258 Mbps/403 Kpps</td>
</tr>
<tr>
<td>Cycles/packet at 125 MHz</td>
<td>310</td>
</tr>
<tr>
<td>Ingress actual cycles/functions</td>
<td>170 - flow association, header processing, replacement, replacement, checksum, flow state update</td>
</tr>
<tr>
<td>Traffic actual cycles/functions</td>
<td>90 – flow status check, jitter buffer timing check, packet release</td>
</tr>
<tr>
<td>Egress actual cycles/functions</td>
<td>130 – flow association, header replacement, checksum creation, state update, packet release</td>
</tr>
</tbody>
</table>
Conclusion

- **Milliflows present at high density flow aggregation nodes**
  - The bandwidth of the aggregate flow more than 1000 times that of component flows

- **Systems to service aggregation nodes are required to meet stringent constraints**
  - Real-time service for several thousand component flows under tight power constraints

- **Developed an experimental SoC IC targeting real-time milliflow aggregation**
  - Functional acceleration for primary functions in VoIP packet processing
  - Results in an effective power-performance trade-off

- **Thanks to**
  - Sunil Chaudhari, Kumar Ganapathy, Saleem Mohammadali, Jonathan Liu, Saurin Shah, David Gilbert, T.H. Lu, Sameer Nanavati, Jennifer Donnelly, Carl Alberola, Manoj Mehta