A 10Gbps Ethernet TCP/IP Processor

HOT Chips 15 - August 17, 2003

Jianping Xu, Nitin Borkar, Vasantha Erraguntla, Yatin Hoskote, Tanay Karnik, Sriram Vangal, Justin Rattner

Microprocessor Research, Intel Labs
Outline

- TCP/IP processing challenges
- TCP/IP protocol processor (TIPP) features
- TIPP microarchitecture
- TIPP implementation
- Experimental chip performance
- Summary
TCP/IP Challenges

<table>
<thead>
<tr>
<th>Speed</th>
<th>Packets/sec</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>100MbE</td>
<td>148k</td>
<td>6.72 ms</td>
</tr>
<tr>
<td>1GbE</td>
<td>1.48M</td>
<td>672 ns</td>
</tr>
<tr>
<td>10GbE</td>
<td>14.8M</td>
<td>67.2 ns</td>
</tr>
<tr>
<td>40GbE</td>
<td>59.5M</td>
<td>16.8 ns</td>
</tr>
</tbody>
</table>

General processor cannot keep up.

Architectural Enhancements Required to Close the Gap

Process Improvement Provides This Gain

Process Gain by Moore’s Law

Network Bandwidth Demand
CPU Requirements

Throughput & CPU Utilization (2.4GHz)

Transfer Size

<table>
<thead>
<tr>
<th>Transfer Size</th>
<th>TX</th>
<th>RX</th>
</tr>
</thead>
<tbody>
<tr>
<td>64B</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>1KB</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2KB</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>16KB</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>64KB</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

% CPU utilization

<table>
<thead>
<tr>
<th>Transfer Size</th>
<th>CPU Utilization-TX</th>
<th>CPU Utilization-RX</th>
</tr>
</thead>
<tbody>
<tr>
<td>64B</td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1KB</td>
<td>80.0%</td>
<td>20.0%</td>
</tr>
<tr>
<td>2KB</td>
<td>60.0%</td>
<td>40.0%</td>
</tr>
<tr>
<td>16KB</td>
<td>40.0%</td>
<td>60.0%</td>
</tr>
<tr>
<td>64KB</td>
<td>20.0%</td>
<td>80.0%</td>
</tr>
</tbody>
</table>

Throughput

<table>
<thead>
<tr>
<th>Transfer Size</th>
<th>Throughput-TX</th>
<th>Throughput-RX</th>
</tr>
</thead>
<tbody>
<tr>
<td>64B</td>
<td>1000 Mbps</td>
<td>0 Mbps</td>
</tr>
<tr>
<td>1KB</td>
<td>800 Mbps</td>
<td>200 Mbps</td>
</tr>
<tr>
<td>2KB</td>
<td>600 Mbps</td>
<td>400 Mbps</td>
</tr>
<tr>
<td>16KB</td>
<td>400 Mbps</td>
<td>600 Mbps</td>
</tr>
<tr>
<td>64KB</td>
<td>200 Mbps</td>
<td>800 Mbps</td>
</tr>
</tbody>
</table>
Egress Processing

- Ingress Proc
- Egress Proc
- SPI I/F
- RxBuf
- XBar
- SDRAM I/F
- Host
- MAC
- PHY
- TxBuf
- Exec Unit (UDP Proc)
- IP Fragmentation
- TCP Egress Processing
- Boot PROM
- IP Forwarder
- Egress Processing

Stream from send buffer

Internal Bus
TCP Offload

Why TCP processing?
- 82% of all traffic is TCP
- TCP processing is the bottleneck
  - complex
  - compute intensive
Key Features of TIPP

- **Special purpose processor**
  - Dual frequency, low latency, buffer-free design
  - High frequency execution core
  - Accelerated context lookup and loading

- **Programmability for up-to-date protocols**
  - Programmable design with special instructions
  - Rapid validation and debug

- **Scalable solution**
  - Across bandwidth and packet sizes
  - Extendable to multi-core solution
TIPP Organization

CAM — Content Addressable Memories
TCB — Transmission Control Block
ECore — Execution Core
IROM — Instruction ROM
IDB — Ingress Dispatcher Block
SR — State Register
SP — Scratch Pad
EB — Egress Buffer
# Instruction Set

## General purpose instructions

<table>
<thead>
<tr>
<th></th>
<th>Instruction</th>
<th>Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOAD</td>
<td>A ← data</td>
</tr>
<tr>
<td>2</td>
<td>MOV</td>
<td>A → B</td>
</tr>
<tr>
<td>3</td>
<td>AND</td>
<td>A B → cond</td>
</tr>
<tr>
<td>4</td>
<td>OR</td>
<td>A B → cond</td>
</tr>
<tr>
<td>5</td>
<td>ADD</td>
<td>A B → C</td>
</tr>
<tr>
<td>6</td>
<td>SUB</td>
<td>A B → C</td>
</tr>
<tr>
<td>7</td>
<td>CMP</td>
<td>A B → cond</td>
</tr>
<tr>
<td>8</td>
<td>EQUAL</td>
<td>A B → cond</td>
</tr>
<tr>
<td>9</td>
<td>NOT</td>
<td>A → C</td>
</tr>
<tr>
<td>10</td>
<td>BREQZ / BRNEQZ</td>
<td>label</td>
</tr>
<tr>
<td>11</td>
<td>JMP</td>
<td>label</td>
</tr>
<tr>
<td>12</td>
<td>SHL2</td>
<td>A</td>
</tr>
<tr>
<td>13</td>
<td>NOP</td>
<td></td>
</tr>
</tbody>
</table>

## Special purpose instructions

<table>
<thead>
<tr>
<th></th>
<th>Instruction</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CAM1CLR</td>
<td>index</td>
</tr>
<tr>
<td>2</td>
<td>CAM2CLR</td>
<td>index</td>
</tr>
<tr>
<td>3</td>
<td>CAM1LKUP</td>
<td>key → index, data</td>
</tr>
<tr>
<td>4</td>
<td>CAM2LKUP</td>
<td>key → index, data</td>
</tr>
<tr>
<td>5</td>
<td>CAM2WR</td>
<td>key ← data</td>
</tr>
<tr>
<td>6</td>
<td>CAM2EMPTY</td>
<td>→ cond</td>
</tr>
<tr>
<td>7</td>
<td>TCBWR</td>
<td>index ← data</td>
</tr>
<tr>
<td>8</td>
<td>TCBRD</td>
<td>index → data</td>
</tr>
</tbody>
</table>
High Speed Execution Core

- CAM2
- Key
- CAM1
- TCB
- ALU
- TCB
- Ingress stream
- IDB
- SR
- SP
- Pipelined ALU
- ALU output
- Next address
- Branch address
- Start address
- IROM
- PC
- decode
- IR
- pipestages
- decoded
Packets Reordering

- Packets arrive in random order
- Conventional software solution uses sorting to reorder packets
- Hardware sorting is expensive and cumbersome
- Use CAM to eliminate sorting
  - Store and lookup of packets in CAMs by sequence number
  - Combine adjacent packets in CAMs

Critical for wire speed processing
System Scalability

- Multi-core solution with each core supporting up to 8K connections
- Large number of connections: > 32K to 64K
- 4 signal handshake between CTL and each TIPP

TIPP offload engine
Enabling Circuit Technologies

- Quaternary tree adder architecture
- Dual-$V_T$ design
- Semi-dynamic flip-flops
- High speed instruction ROM
- High bandwidth, leakage tolerant register file
- Adaptive body bias
- Multiple clock domain synchronization
Experimental Chip

- **Chip Area**: $2.23 \times 3.54\text{mm}^2$
- **Process**: 90nm CMOS
- **Interconnect**: 1 poly, 7 metal
- **Transistors**: 460K
- **Pad count**: 306
Evaluation Board
Measured Processing Performance

- 4.4 Gbps, 0.9 V
- 9.64 Gbps, 1.72 V

Processing Rate (Gbps)

Vcc (V)
Power Consumption

Simulated Power Density

Watts/sq.cm. @1.2V, 100C

250-300 200-250 150-200 100-150 50-100 0-50

9.64Gbps, 1.72V

4.4Gbps, 0.9V
Summary

- Programmable hardware engine
  - Line-speed TCP ingress processing
  - Support minimum packet size
- Dual frequency and buffer-free design
  - High speed execution core
  - 9.64Gbps processing at 1.72V, 6.39W in 90nm communication CMOS process
  - Extendable to large number of connections
- Dynamic reordering of packets in hardware
Acknowledgments


• A. Foong, F. Hady for simulation contributions

• S. Borkar, M. Haycock for encouragement and support