CSE398: Network Systems Design

Instructor: Dr. Liang Cheng
Department of Computer Science and Engineering
P.C. Rossin College of Engineering & Applied Science
Lehigh University

March 30, 2005

Outline

- Recap
  - Scaling a network processor
- Examples of commercial network processors
- Design tradeoffs and consequences
- Summary and homework

Augmented RISC Processor

- Alchemy Semiconductor Inc.
- Au1000 chip
  - Instruction cache
  - Data cache

Embedded Processor Plus Coprocessors

- AMCC (Applied Micro Circuits Co.)
- nP7510 chip
  - Six nP cores
- Programming
  - Implicit parallelism
- Coprocessors
  - Meter engine for SNMP's RMON

Pipeline of Homogeneous Processors

- Cisco Systems, Inc.
- Parallel eXpress Forwarding (PXF)
  - MAC classify
  - Accounting & ICMP
  - FIB & Netflow
  - MPLS classify
  - Access control
  - CAR
  - MLPPP
  - WRED

Pipeline of Heterogeneous Processors

- EZchip Co.
  - TOPparse: header field extraction and classification
  - TOPsearch: table lookup
  - TOPresolve: queue management and forwarding
  - TOPmodify: packet header and content modification
- Onboard SRAM
- Interface to external DRAM

Extensive and Diverse Processors

- Hifn (IBM)
  - Embedded processor complex

Homogeneous Parallel Processors plus Controller

- Intel
- IXP1200
    - Embeddd RISC processor (StrongARM)

StrongARM Role (1)

- (a) Single IXP1200
- (b) Multiple IXP1200s

StrongARM Role (2)

- Tasks
  - Bootstrapping
  - Exception handling
  - Higher-layer protocol processing
  - Interactive debugging
  - Diagnostics and logging
  - Memory allocation
  - User interface and/or interface to the GPP
  - Control of packet processors
  - Other administrative functions

StrongARM Role (3)

- StrongARM characteristics
- Reduced Instruction Set Computer (RISC)
- Thirty-two bit arithmetic
  - Configurable in two modes ?
- Vector floating point provided via a coprocessor
- Byte addressable memory
  - Virtual memory support
  - Single, uniform address space
  - Includes memories and devices
- Built-in serial port
- Facilities for a kernelized operating system

Outline

- Recap
- Examples of commercial network processors
- Design tradeoffs and consequences
- Summary and homework

Low Development Cost vs. Performance

- The fundamental economic motivation
- ASIC costs $1M to develop
- Network processor costs programmer time

Programmability vs. Processing Speed

- Programmable hardware is slower
- Flexibility costs...

Speed vs. Functionality

- Generic idea
  - Processor with most functionality is slowest
  - Adding functionality to NP lowers its overall "speed"

Speed

- Can be evaluated by
  - Packet rate
  - Data rate
  - Burst size
  - CBR, VBR

Per-Interface Rates vs. Aggregate Rates

- Per-interface rate important if
  - Physical connections form bottleneck
  - System scales by having faster interfaces
- Aggregate rate important if
  - Fabric forms bottleneck
  - System scales by having more interfaces

Lookaside Coprocessors vs. Flow-Through Coprocessors

- Flow-through pipeline
  - Operates at wire speed
  - Difficult to change
- Lookaside
  - Modular and easy to change
  - Invocation can be bottleneck

Uniform Pipeline vs. Synchronized Pipeline

- Uniform pipeline
  - Operates in lock-step like assembly line
  - Each stage must finish in exactly the same time
- Synchronized pipeline
  - Buffers allow computation at each stage to differ
  - Synchronization expensive

Explicit Parallelism vs. Cost and Programmability

- Explicit parallelism
  - Hardware is less complex
  - More difficult to program
- Implicit parallelism
  - Easier to program
  - Slightly lower performance

Parallelism vs. Strict Packet Ordering

- Increased parallelism
  - Improves performance
  - Results in out-of-order packets
- Strict packet ordering
  - Aids protocols such as TCP
  - Can nullify use of parallelism

Stateful Classification vs. High-Speed Parallel Classification

- Static classification
  - Keeps no state
  - Is the fastest
- Dynamic classification
  - Keeps state
  - Requires synchronization for updates

I/O Performance vs. Pin Count

- Bus width
  - Increase to produce higher throughput
  - Decrease to take fewer pins

Programming Languages

- A three-way tradeoff
- Can have two, but not three of
  - Ease of programming
  - Functionality
  - Performance

Multithreading: Throughput vs. Ease of Programming

- Multiple threads of control can increase throughput
- Planning the operation of threads that exhibit less contention requires more programmer’s effort

Traffic Management vs. High-Speed Forwarding

- Traffic management
  - Can manage traffic on multiple, independent flows
  - Requires extra processing
- Blind forwarding
  - Performed at highest speed
  - Does not distinguish among flows

Backward Compatibility vs. Architectural Advances Lookup

- Backward compatibility
  - Keeps same instruction set through multiple versions
  - May not provide maximal performance
- Architectural advances
  - Allows more optimizations
  - Difficult for programmers

Parallelism vs. Pipelining

- Both are fundamental performance techniques
- Usually used in combination: pipeline of parallel processors
    - How long is pipeline?
    - How much parallelism at each stage?

Outline

- Recap
- Examples
- Design tradeoffs and consequences
- Summary and homework

Summary

- Many different architectures
- Many design tradeoffs