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 23, 2005

Outline

- Recap
  - Complexity of network processor design
  - Lab time log
- Network processor architectures
- Summary and homework

Network Processor Architectures

- Primary architecture characteristics
- Packet flow
- Software architecture
- Assigning functionality to processor hierarchy

Primary Characteristics

- Processor hierarchy
- Memory hierarchy
- Internal transfer mechanisms
- External interface and communication mechanisms
- Special-purpose hardware
- Polling and notification mechanisms
- Concurrent and parallel execution support
- Programming model and paradigm
- Hardware and software dispatch mechanisms

Processing Hierarchy

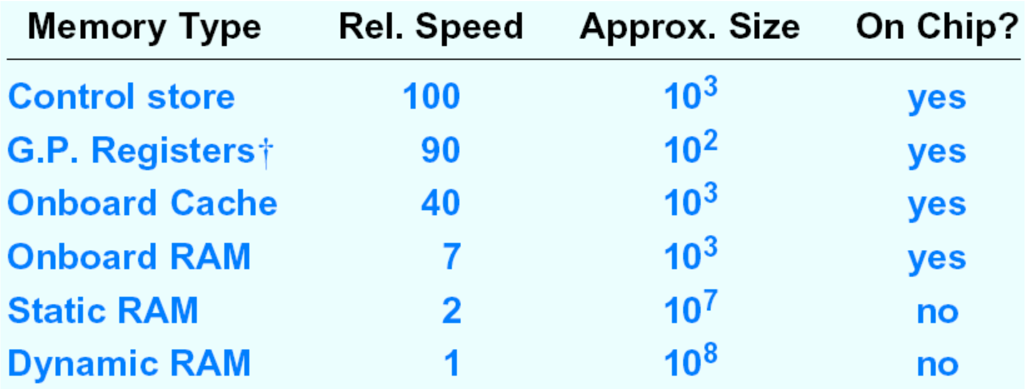
- One or more embedded RISC processors
- One or more specialized coprocessors
- Multiple I/O processors
- One or more fabric interfaces
- One or more data transfer units

Processor Hierarchy – Cont'd

Type	Programmable?	On Chip?
General purpose CPU	y	possible
Embedded processor	y	typical
I/O processor	y	t
Coprocessor	n	t
Fabric interface	n	t
Data transfer unit	n	t
Framer	n	possible
Physical transmitter	n	possible

Memory Hierarchy

- Memory measurements
  - Random access latency
  - Sequential access latency
  - Throughput
  - Cost
  - Internal
  - External



Memory Type	Rel. Speed	Approx. Size	On Chip?
Control store	100	10^3	yes
G.P. Registers†	90	10^2	yes
Onboard Cache	40	10^3	yes
Onboard RAM	7	10^3	yes
Static RAM	2	10^7	no
Dynamic RAM	1	10^8	no

Internal Transfer Mechanisms

- Programmers are free to choose … =>
- Internal bus
- Hardware FIFOs
- Transfer registers
- Onboard shared memory

External Interface and Communication Mechanisms

- Standard and specialized bus interfaces
- Memory interfaces
- Direct I/O interfaces
- Switching fabric interface

Special-purpose Hardware

- Arbitrator
- I/O manager

Polling and Notification Mechanisms

- Handle asynchronous events
  - Arrival of packet
  - Timer expiration
  - Completion of transfer across the fabric
- Two paradigms
  - Polling
  - Notification

Concurrent Execution Support

- Improves overall throughput
- Multiple threads of execution
- Processor switches context when a thread blocks
- Embedded processor
  - Standard operating system
  - Context switching in software
- I/O processors
  - No operating system
  - Hardware support for context switching
  - Low-overhead or zero-overhead

Concurrent Support Questions

- Local or global threads (does thread execution span multiple processors)?
- Forced or voluntary context switching (are threads pre-emptable)?

Hardware and Software Dispatch Mechanisms

- Refers to overall control of parallel operations
- Dispatcher
  - Chooses operation to perform
  - Assigns to a processor

Implicit and Explicit Parallelism

- Explicit parallelism
  - Exposes parallelism to programmer
  - Requires software to understand parallel hardware
- Implicit parallelism
  - Hides parallel copies of functional units
  - Software written as if single copy executing

Network Processor Architectures

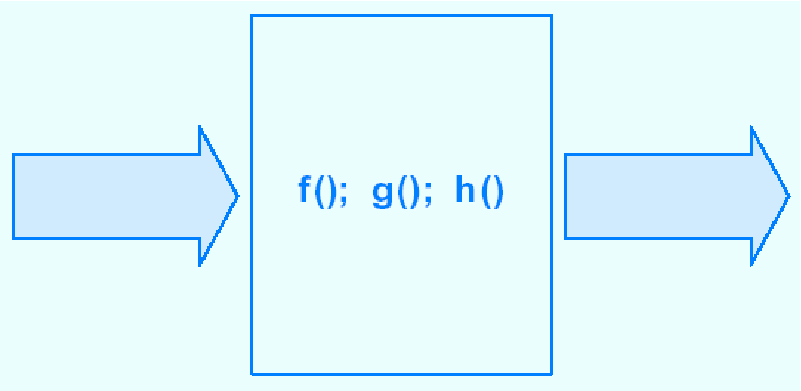
- Primary architecture characteristics
- Architecture styles and packet flow
- Software architecture
- Assigning functionality to processor hierarchy

Architecture Styles

- Embedded processor plus fixed coprocessors
- Embedded processor plus programmable I/O processors
- Parallel (number of processors scales to handle load)
- Pipeline processors

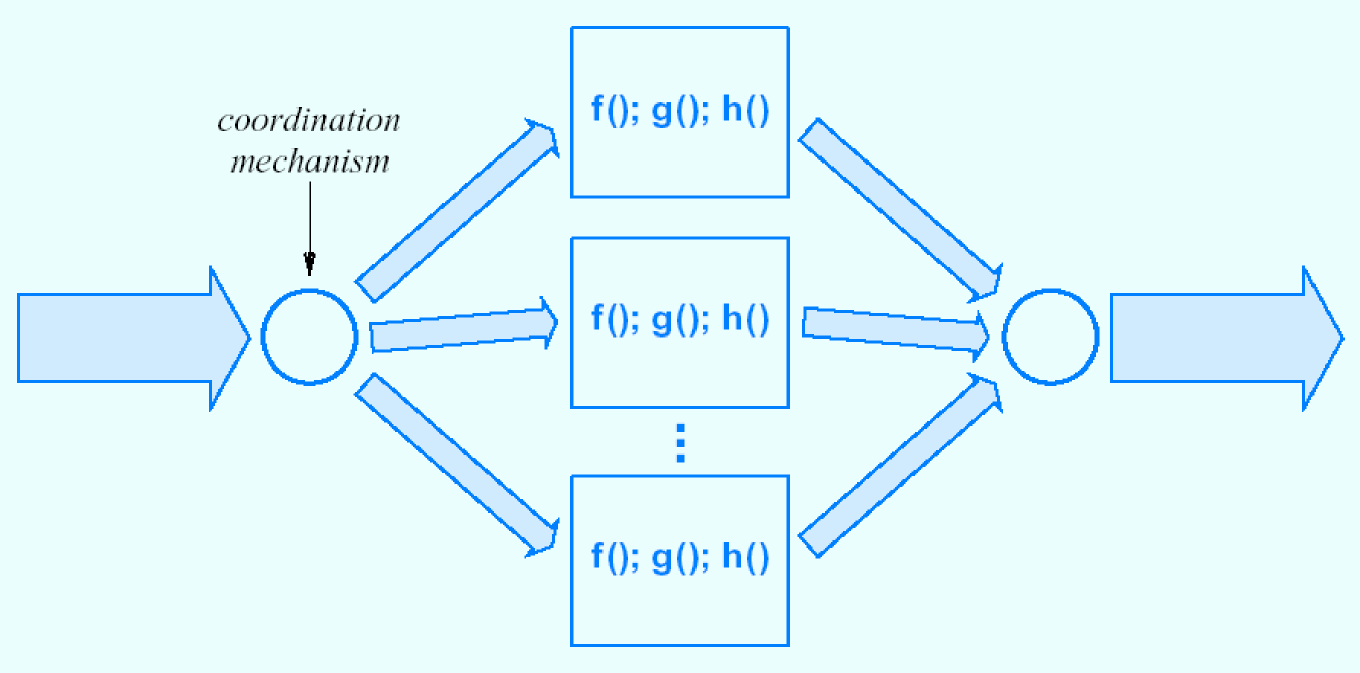
Embedded Processor Architecture

- Single processor
  - Handles all functions
  - Passes packet on
- Known as run-to-completion



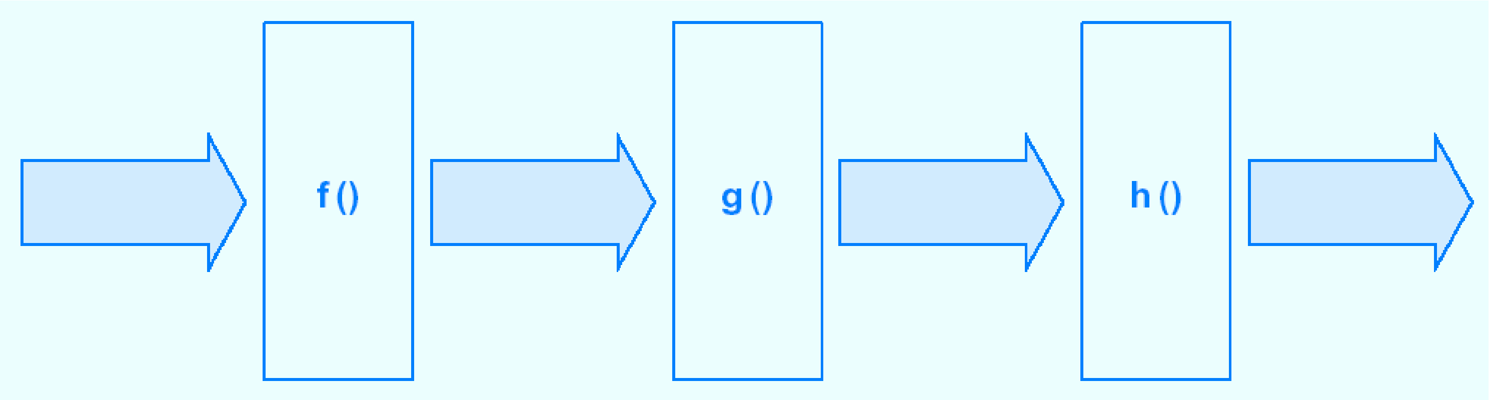
Parallel Architecture

- Each processor handles 1/N of total load



Pipeline Architecture

- Each processor handles one function
- Packet moves through "pipeline"



Clock Rates

- Embedded processor runs at > wire speed
- Parallel processor runs at < wire speed
- Pipeline processor runs at wire speed

Network Processor Architectures

- Primary architecture characteristics
- Architecture styles and packet flow
- Software architecture
- Assigning functionality to processor hierarchy

Software Architecture

- Central program that invokes coprocessors like subroutines
- Central program that interacts with code on intelligent, programmable I/O processors
- Communicating threads
- Event-driven program
- RPC-style (program partitioned among processors)
- Pipeline (even if hardware does not use pipeline)
- Combinations of the above

Example Uses of Programmable Processors

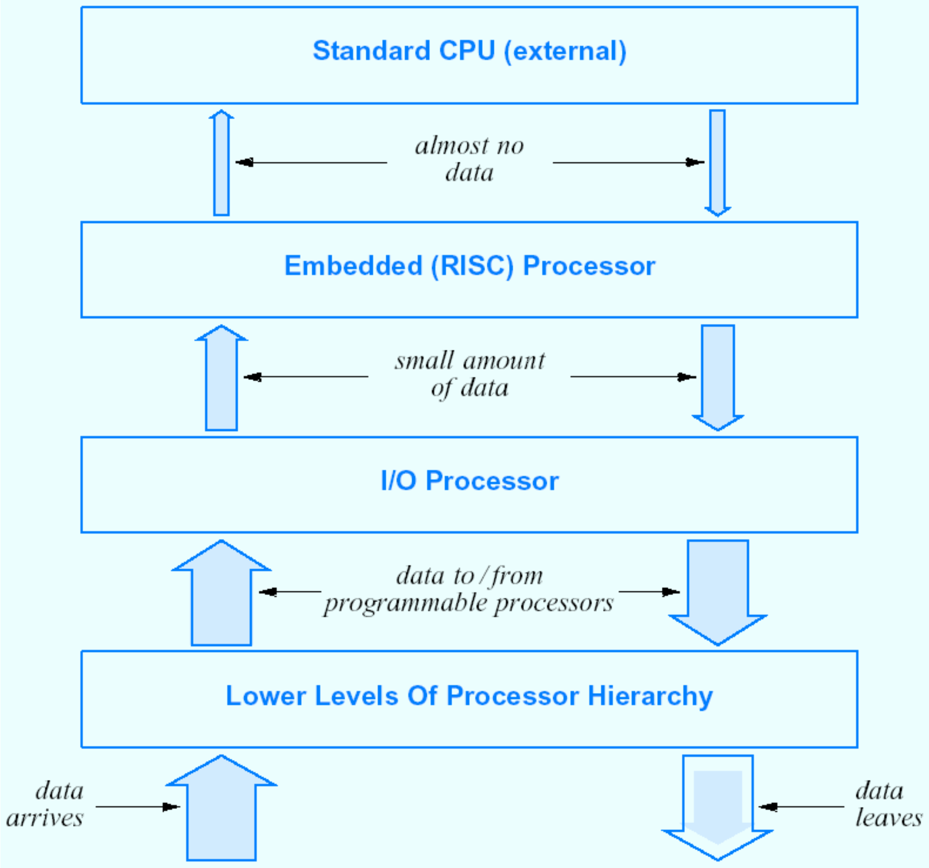
1. Administrative interface
2. Classification
3. Control of I/O processors
4. Exception and error handling
5. Forwarding
6. High-level egress (e.g., traffic shaping)
7. High-level ingress (e.g., reassembly)
8. Higher-layer protocols
9. Low-level egress operations
10. Low-level ingress operations
11. Overall management functions
12. Routing protocols
13. System control

Example Uses of Programmable Processors

- General purpose CPU
  - Highest level functionality
  - Administrative interface
  - System control
  - Overall management functions
  - Routing protocols
- Embedded processor
  - Intermediate functionality
  - Higher-layer protocols
  - Control of I/O processors
  - Exception and error handling
  - High-level ingress (e.g., reassembly)
  - High-level egress (e.g., traffic shaping)
- I/O processor
  - Basic packet processing
  - Classification
  - Forwarding
  - Low-level ingress operations
  - Low-level egress operations

Packet Flow through Hierarchy

- To maximize performance, packet processing tasks should be assigned to the lowest level processor capable of performing the task.


Standard CPU (external)
almost no data
Embedded (RISC) Processor
small amount of data
I/O Processor
data to / from programmable processors
Lower Levels Of Processor Hierarchy
data arrives
data leaves

Outline

- Recap
- Network processor architectures
- Summary and homework