ECE 697J – Advanced Topics in Computer Networks

Course Summary

12/04/03

Final Projects

- Any questions?
- Please send slides by Tuesday noon via email
  - Or bring it on memory stick or CD (not diskette)
- What I am looking for
  - A good idea on how to tackle problem, e.g.:
    - Project A: How and where to implement different components
    - Project B: How to generate different memory channel loads
    - Project C: Identification of key components of IP forwarding
  - Some initial results (important!)
  - A good plan on how to get the remaining results
    - A prediction on what they will look like
  - Any other interesting observation

Course Summary

- Networks
  - Functionality and performance
- Routers
  - Packet forwarding
  - Processing Engines
  - Scalability
- Network Processors
  - Architectures and design-tradeoffs
  - Commercial NPs
  - IXP1200 hardware and software
- Lab Exercises
  - Simulators and tools

Flashback

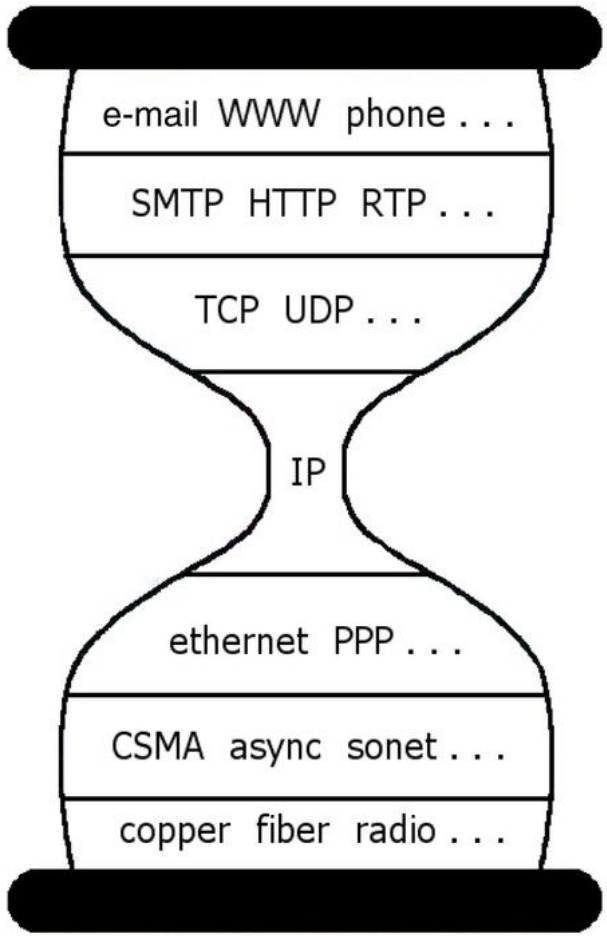
- One slide from each lecture…

What is this Course About?

- Course discusses **network systems** (i.e., routers)
  - Functionality of routers
  - How to implement them efficiently
  - How to expand their functionality for next-generation networks
- This is a broad area – we'll focus on
  - Data-plane (not control plane, routing protocols, …)
  - Packet-based networks (in particular IP networks)
  - Concepts of router functions (not how to setup a Cisco system)
  - Intel IXP network processor (example of a programmable router)

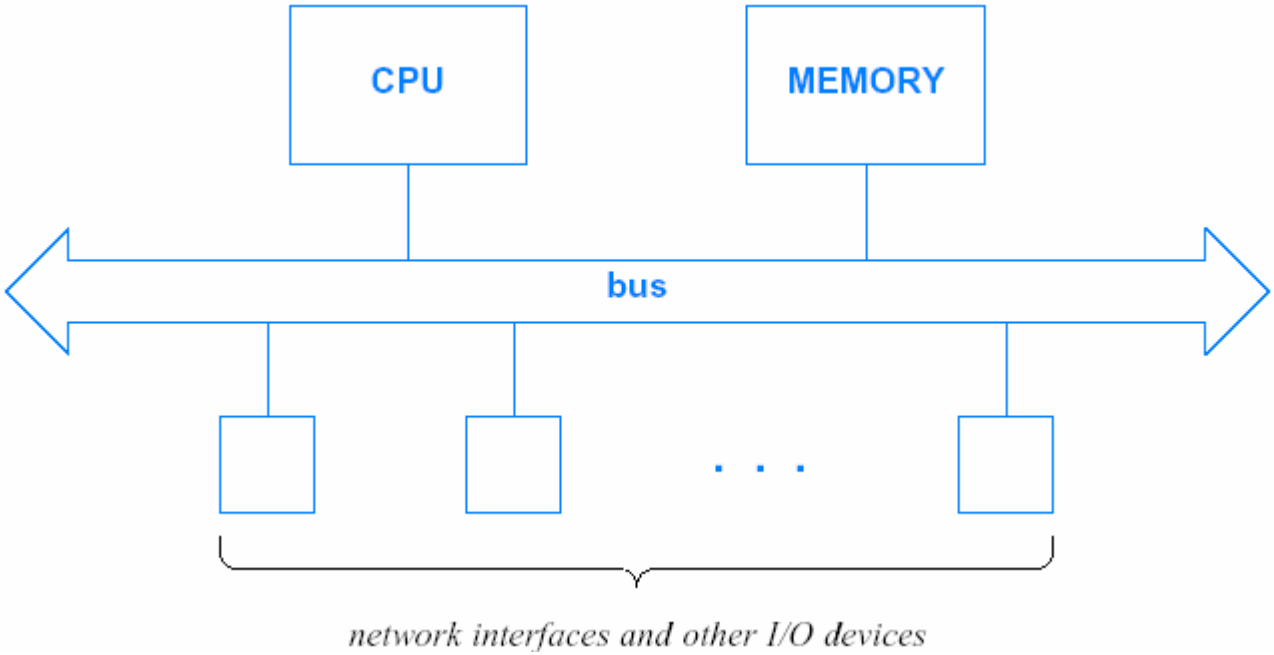
Network Protocols

- Why are several network protocols necessary?
  - Different protocols solve different problems
  - Need a mix of functionality depending on application
- How is interoperability ensured?
  - Common protocol (hourglass model)
  - In the Internet: IP
- What is a protocol suite?
  - A coordinated set of protocols
  - E.g.: HTTP over TCP over IP over Ethernet
- What is a protocol stack?
  - The software that implements a layered protocol suite



Packet Processing on Host

- "Conventional Computer System":
  - Single CPU, memory, 1+ I/O devices, bus interconnect



- Network Interface Card (NIC) used for communication

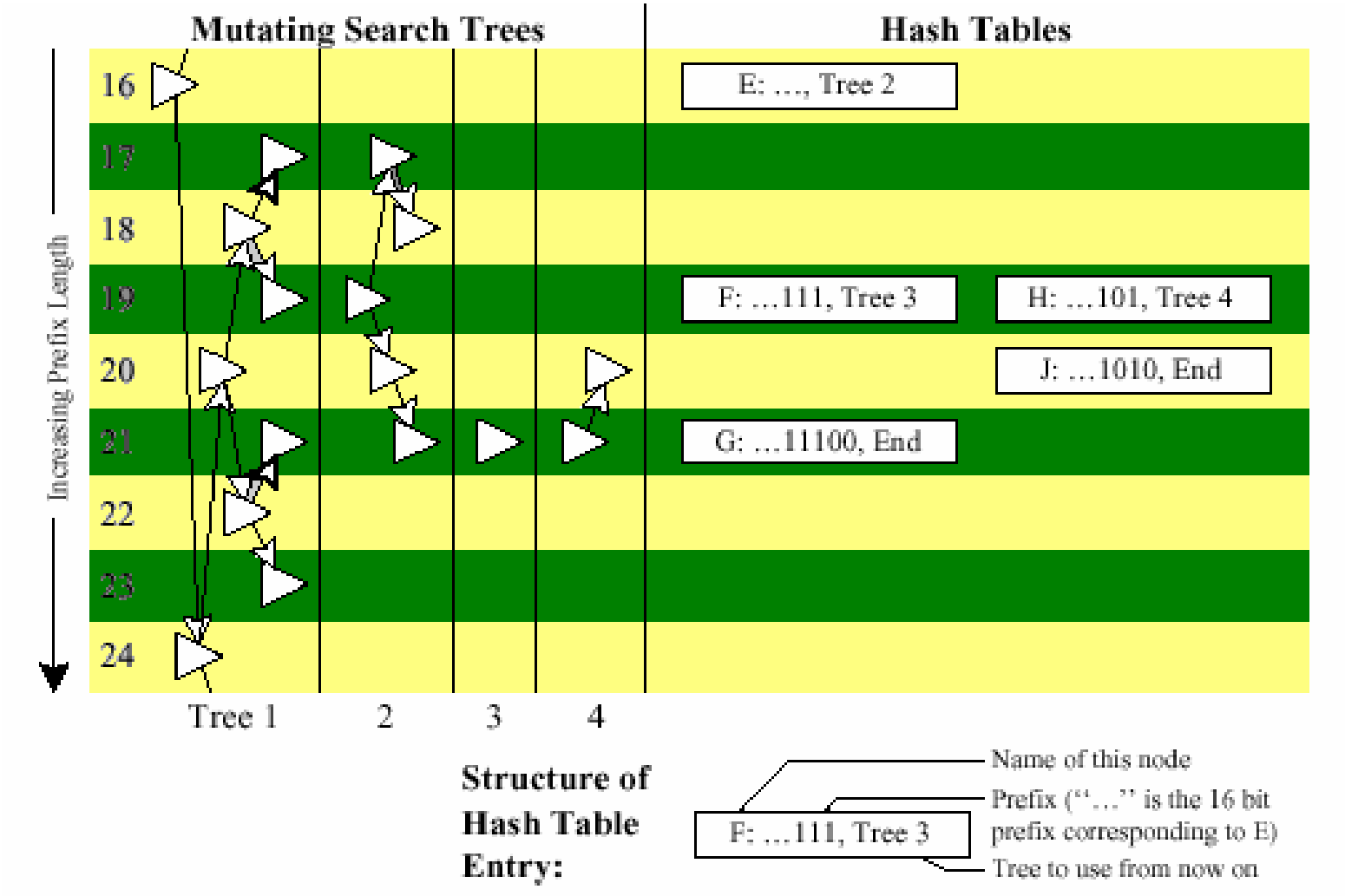
System Goals

- *"To allow it to run arbitrarily long, a network system must be designed with limits on all resources and the limits must be fixed independent of arriving traffic; designs that violate this principle will not be considered"*
- Examples?
- Related statement: "Network systems should be designed to handle worst case traffic."
- How is this different from a general workstation?

Packet Processing Functions

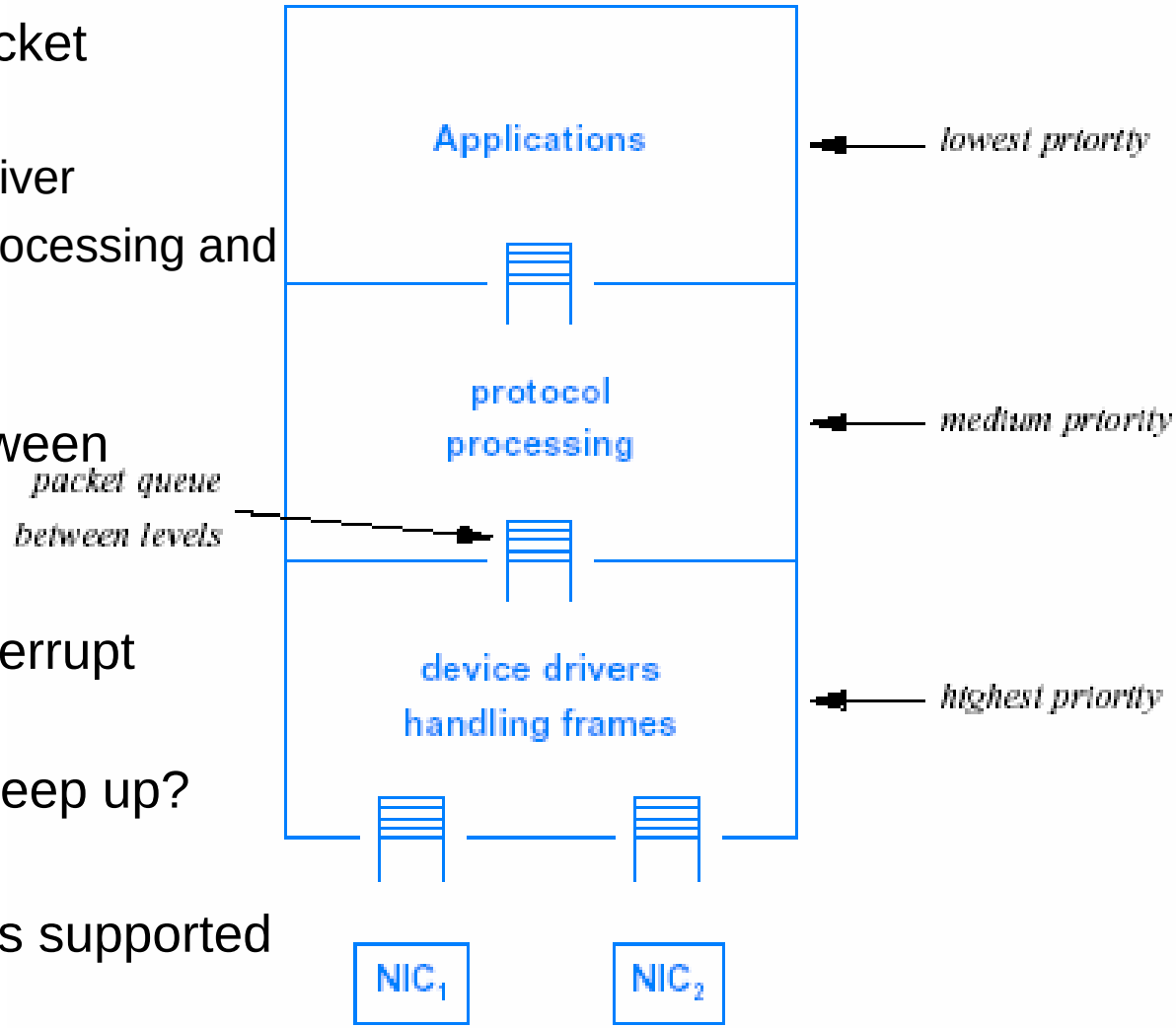
- Basic network system functionality
  - Address lookup
  - Error detection and correction
  - Fragmentation/re-assembly
  - Queuing
  - Scheduling
  - Security
  - Traffic measurement/shaping
  - Protocol demultiplexing
  - Packet classification

Mutating Binary Search (example:)



Packet Processing and Interrupts

- Interrupt levels for packet processing:
  - Highest to device driver
  - Lower to protocol processing and application
  - Why?
- Requires queues between interrupt levels
  - Why?
- Processing in high interrupt should be kept brief
- What if CPU cannot keep up?
  - Livelock
- Only few priority levels supported



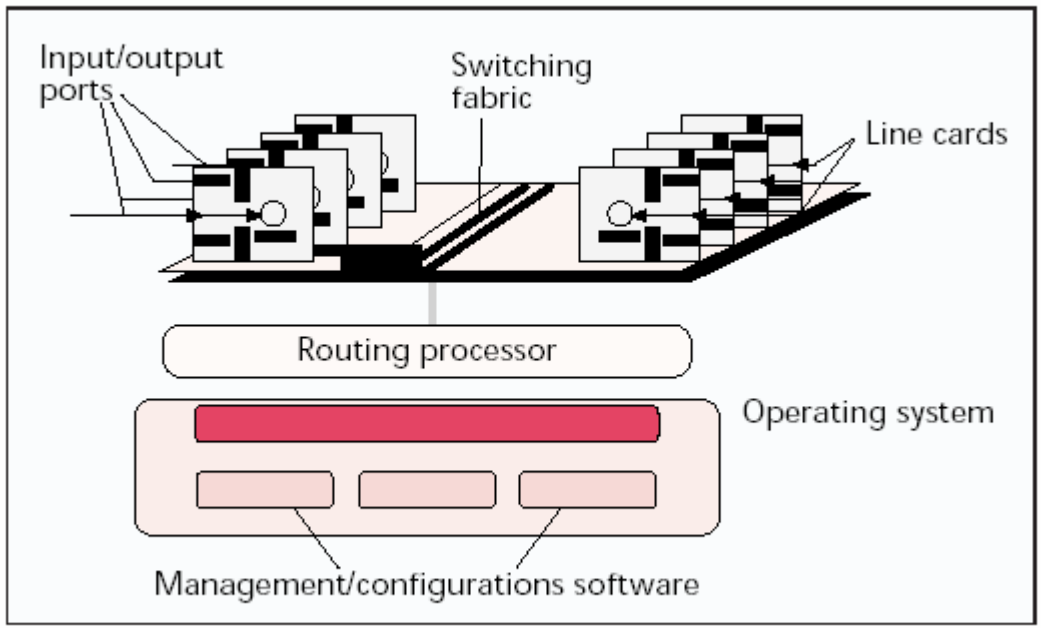
Time Per Packet

- Aggregate packet rate determines time per packet (on single CPU)

Technology	Time per packet for small packets (in µs)	Time per packet for large packets (in µs)
10Base-T	51.20	1,214.40
100Base-T	5.12	121.44
OC-3	3.29	78.09
OC-12	0.82	19.52
1000Base-T	0.51	12.14
OC-48	0.21	4.88
OC-192	0.05	1.22
OC-768	0.01	0.31

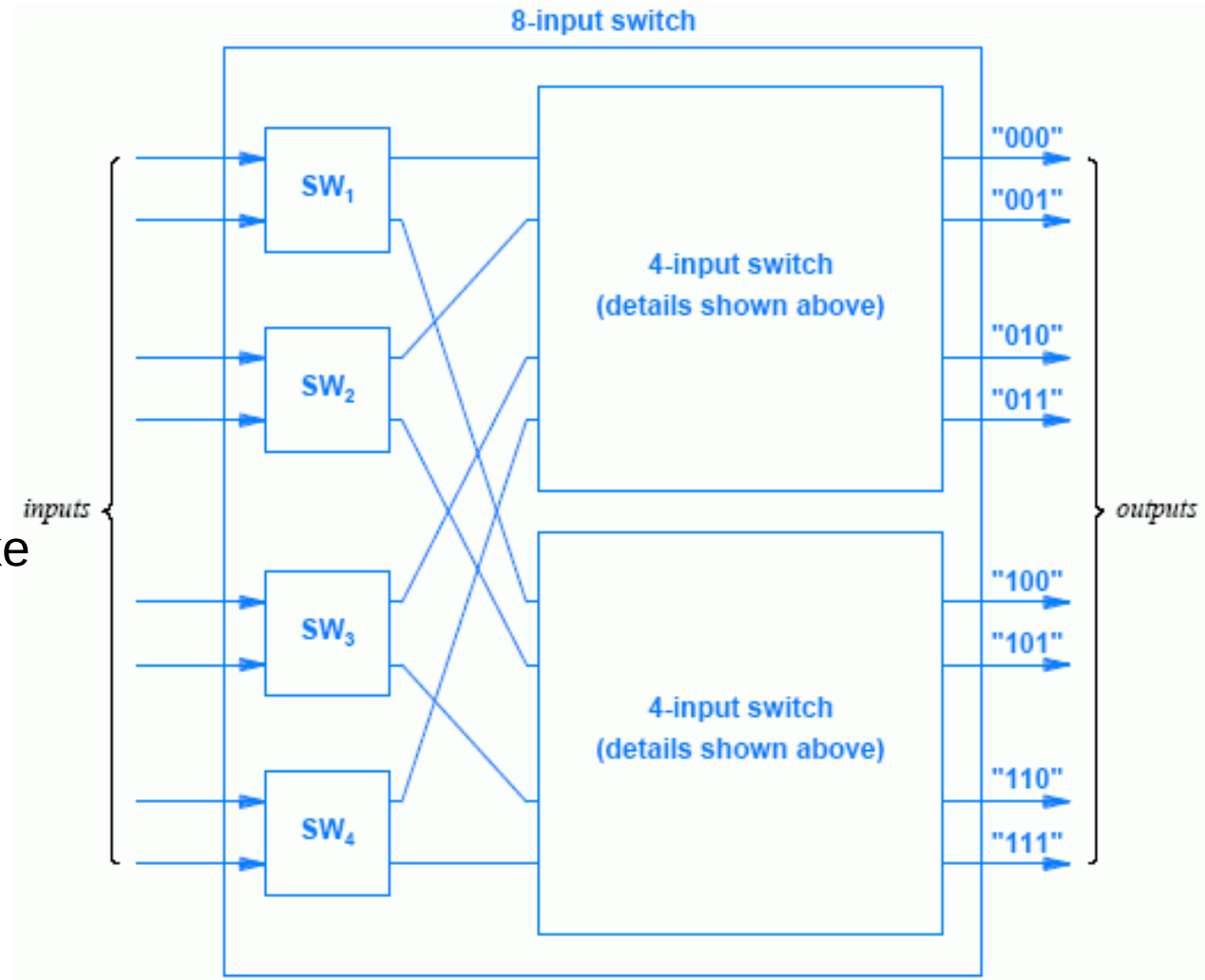
- Packet processing requires in the order of 100s of instructions per packet
- Single CPU router lacks **scalability**

Router Architecture



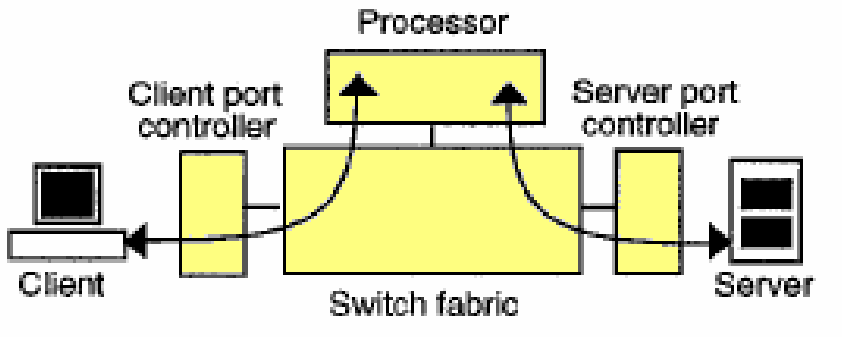
Banyan Switch

- 8-port:
  - Recursive extension
- Self-routing:
  - Based on output port "label" each stage can make local decision
  - Recursive structure will lead to correct destination



Operational Blueprint

Example flow trough an L5 switch



Phase II

•In an application layer proxy, the processor remains on the data path to copy data between the two connections.

•The L5 splices the two TCP connections to get out of the data path.

•TCP splicing requires translating TCP sequence numbers, which is done by the port controllers to ensure fast layer 5 switching.

Effect of caching repairs packets in ARM

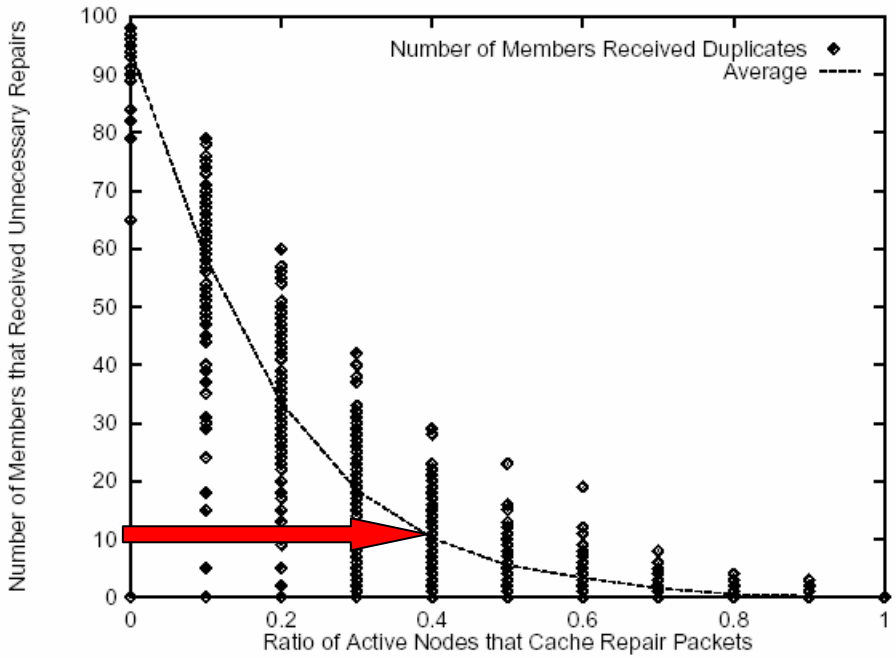


Fig. 10. Effect of caching repairs in ARM and number of receivers that receive unnecessary repairs (random loss, 1000 nodes, group size 100, degree 4). All nodes active. No nodes cache fresh data packets. Nodes that cache repair packets are picked randomly. Source always caches repairs.

Integrated Design Approach

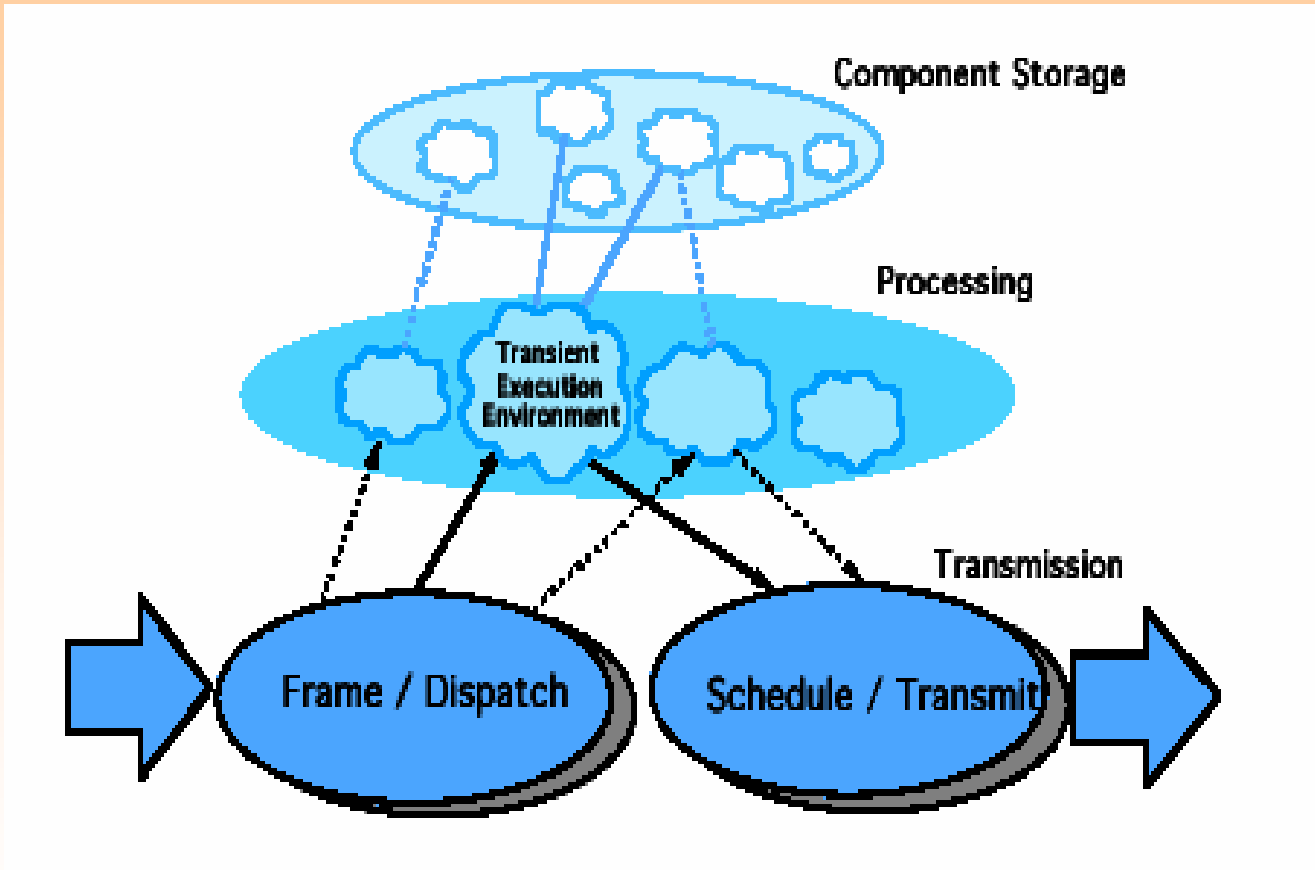
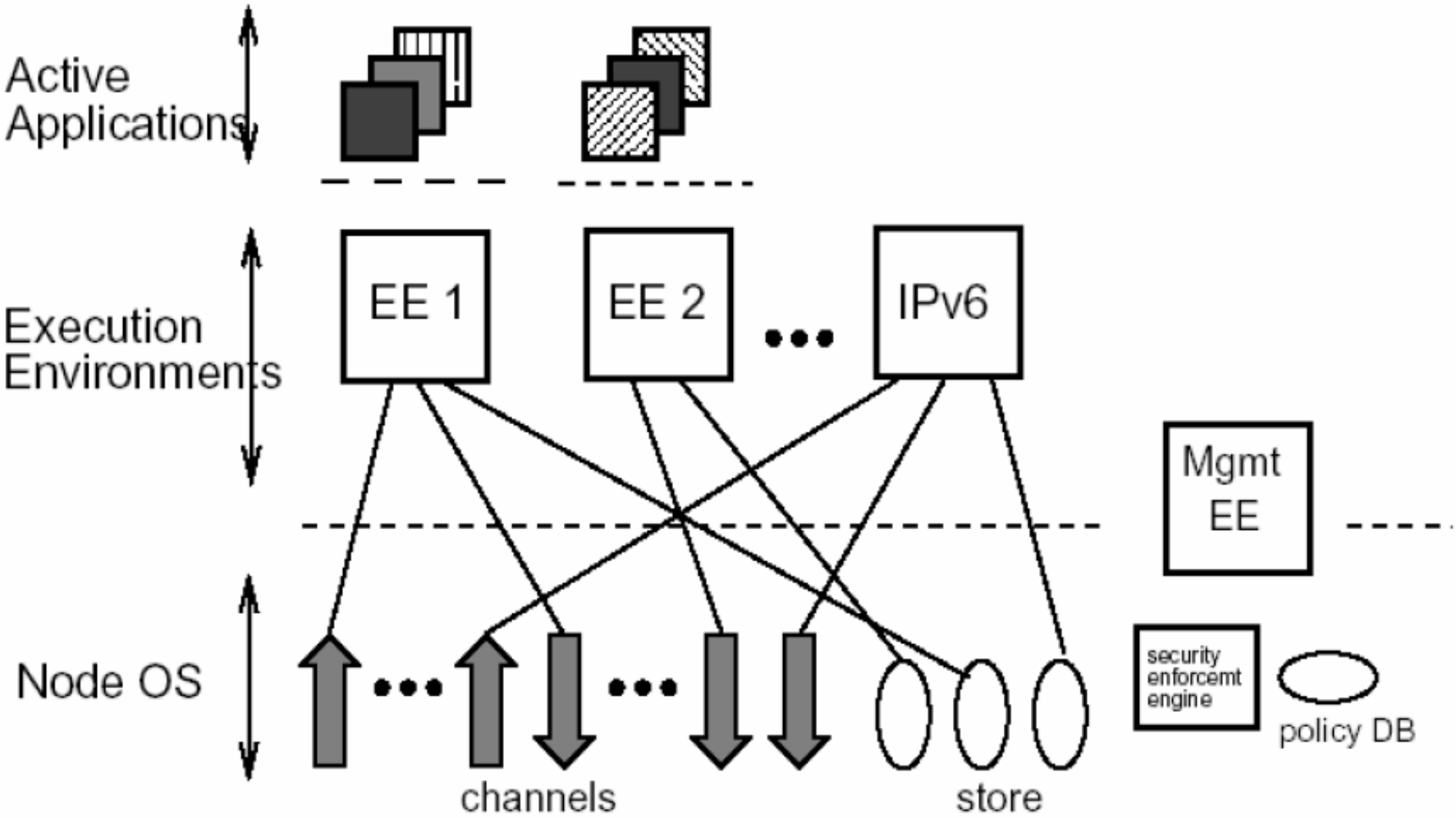
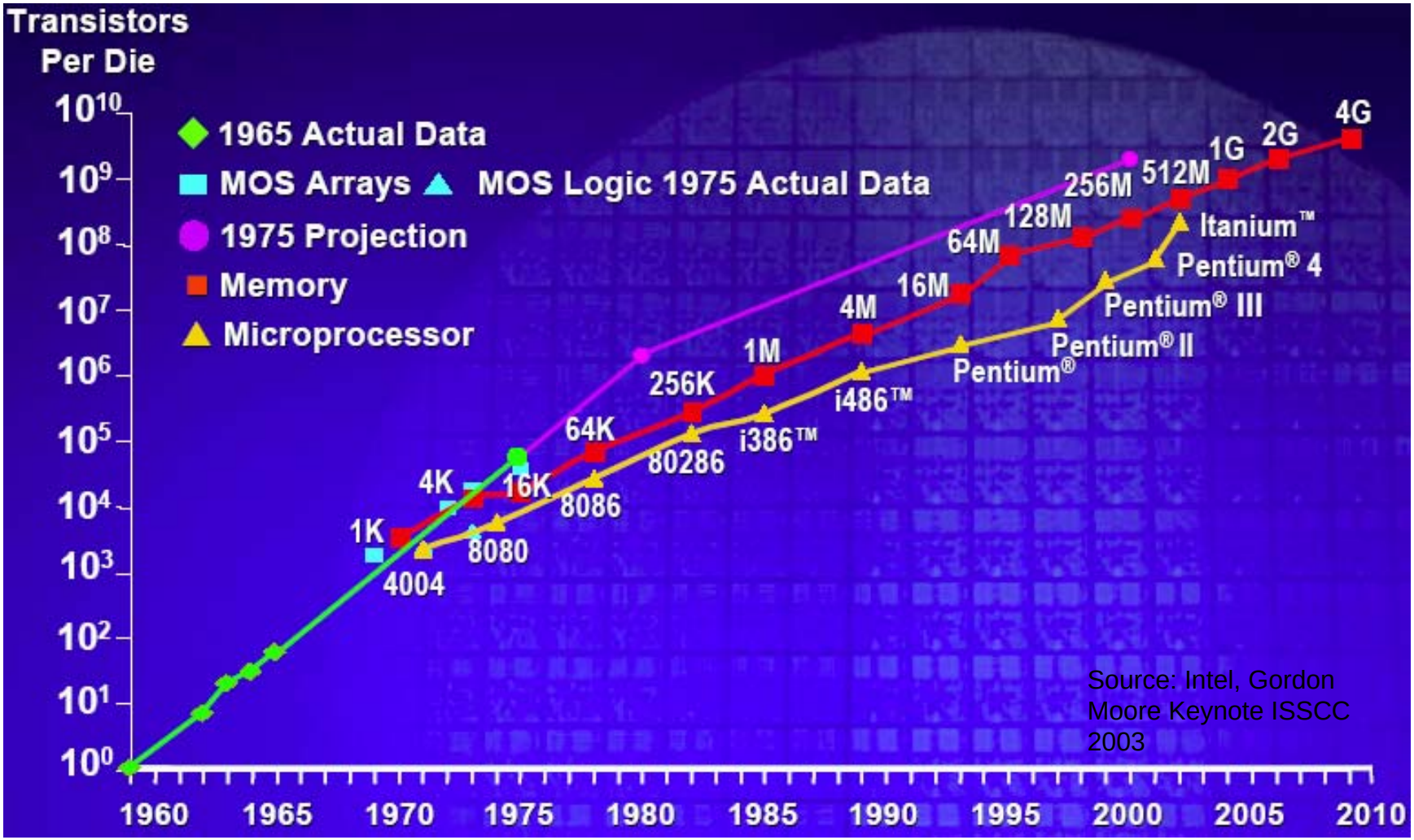


Figure 1. Active Node Organization

NodeOS

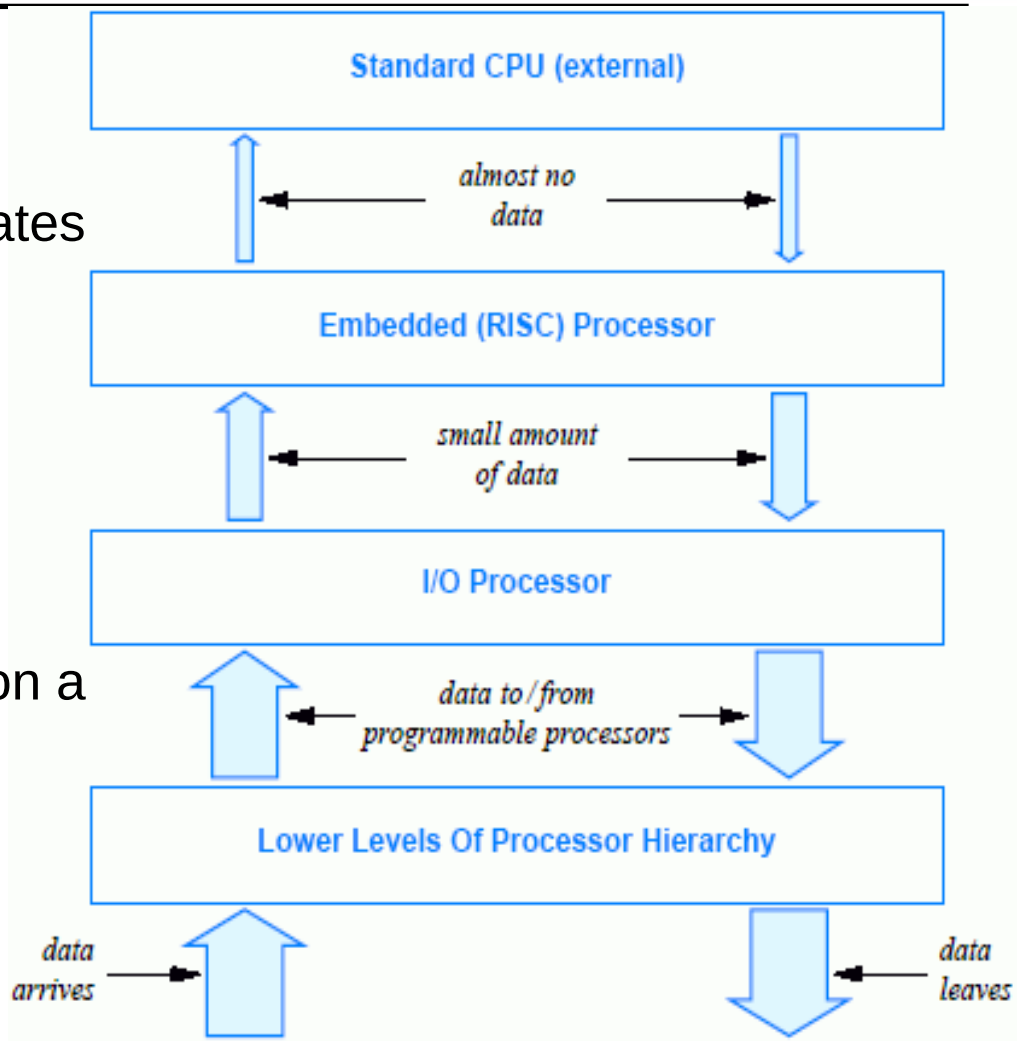

Active Applications
Execution Environments
Node OS
EE 1
EE 2
IPv6
Mgmt EE
security enforcemt engine
policy DB
channels
store

Moore's Law Data



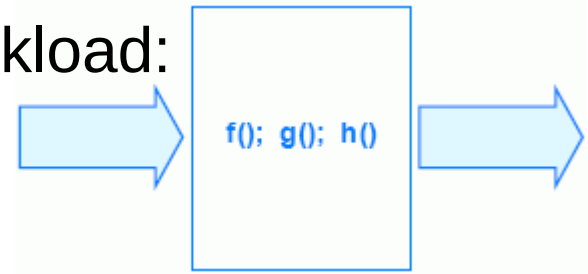
Processor Hierarchy

- What is a processor hierarchy? Why would it be used?
- Processor hierarchy accommodates tasks of different complexity and different frequency
  - Low level of hierarchy: simple, frequent processing
  - High level of hierarchy: occasional, complex processing
- What kind of levels can we find on a router?
  - Several levels of data path processing
  - Several levels of control path processing

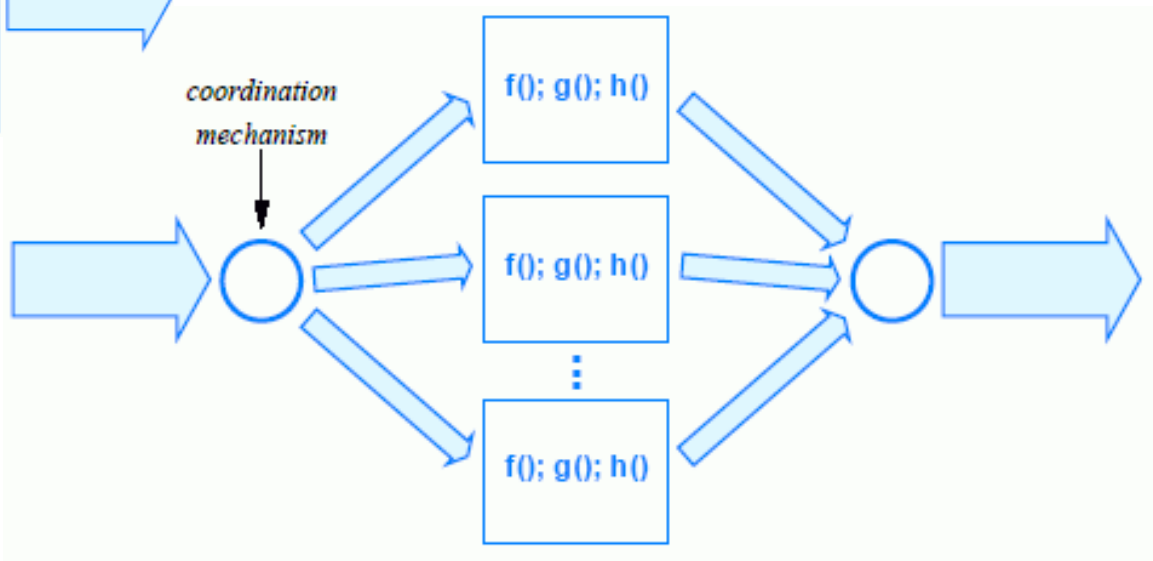


Parallel Architectures

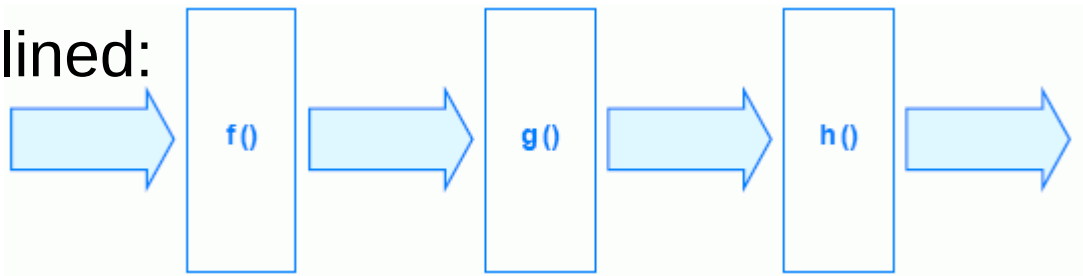
- Workload:



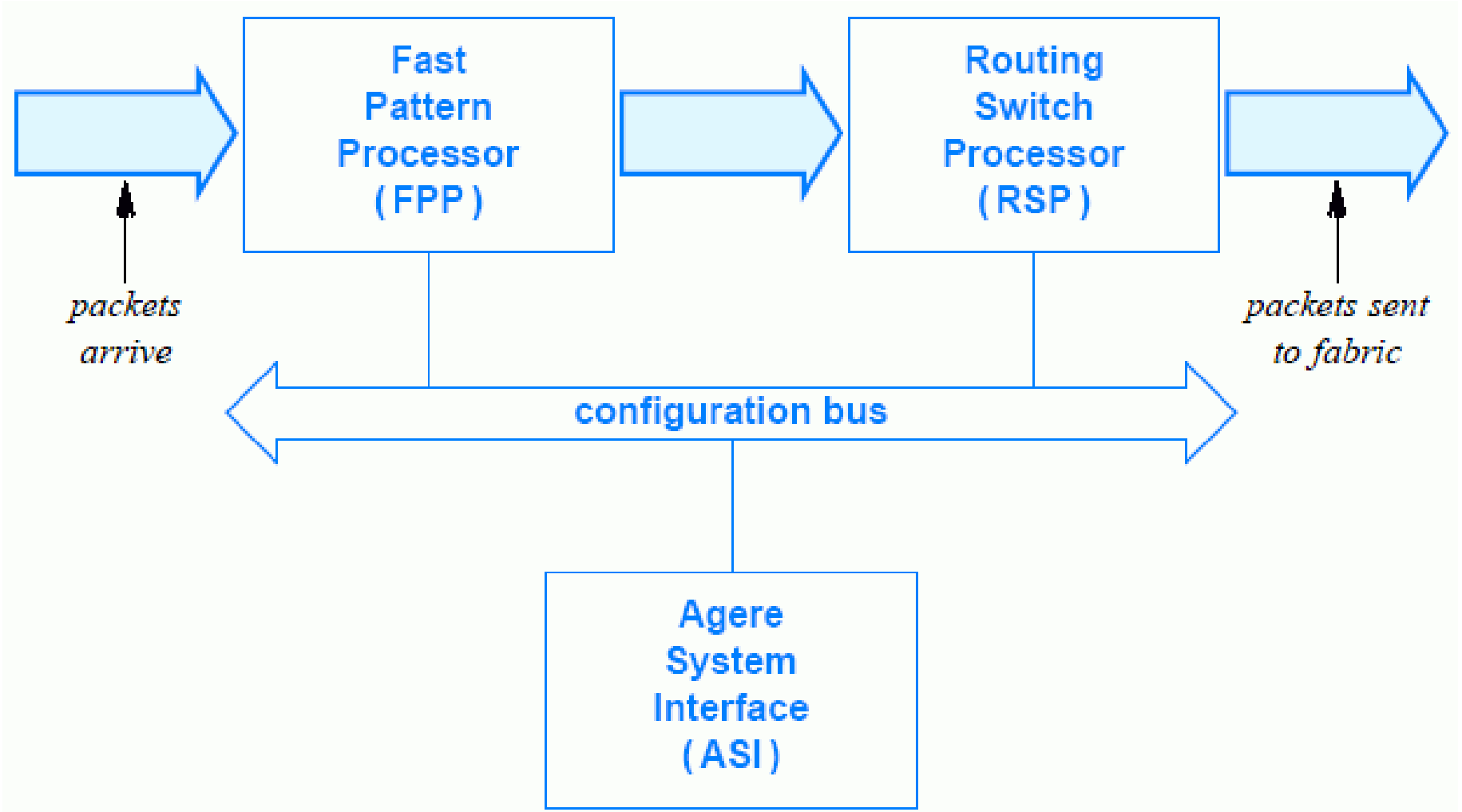
- Parallel:



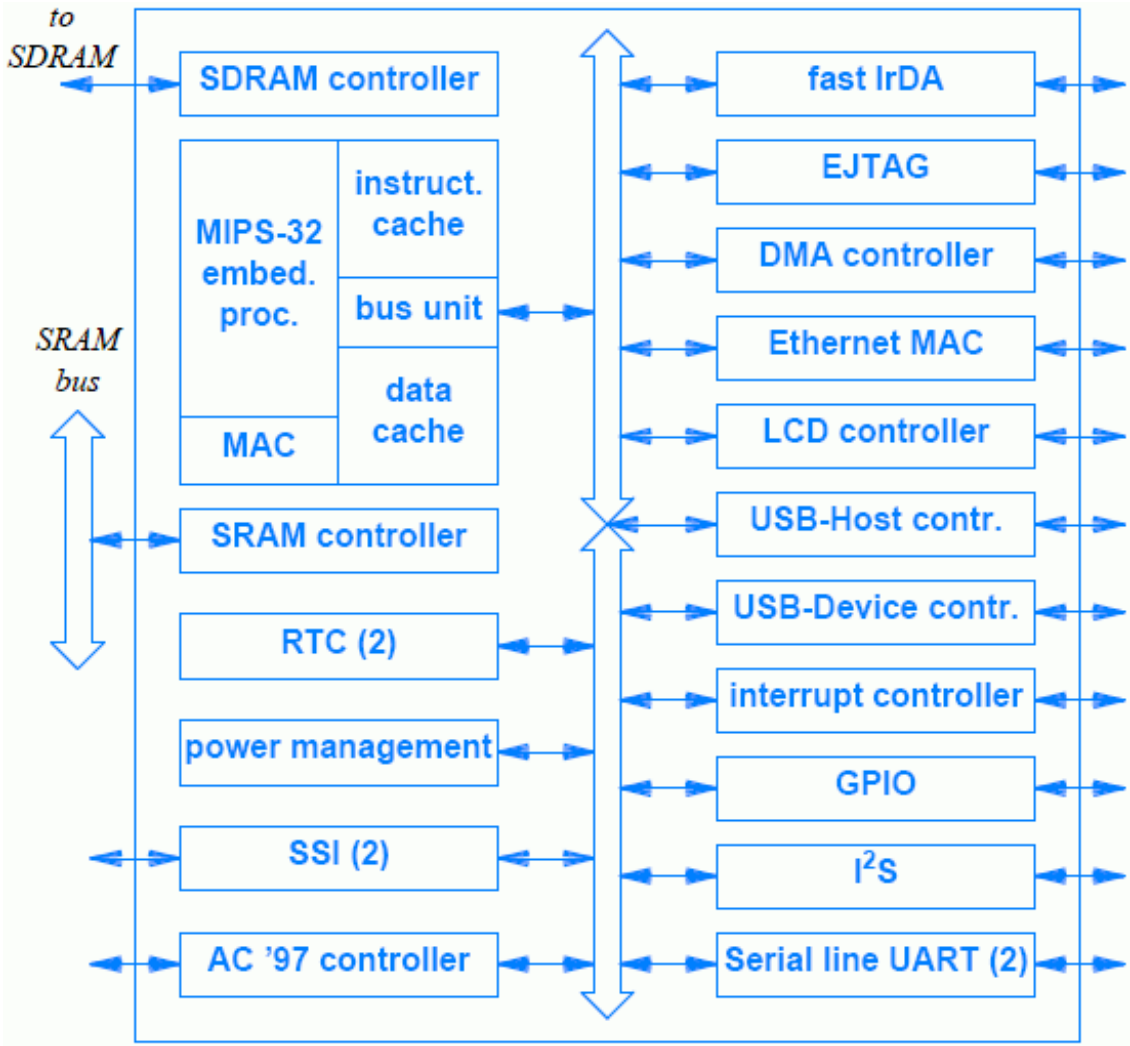
- Pipelined:



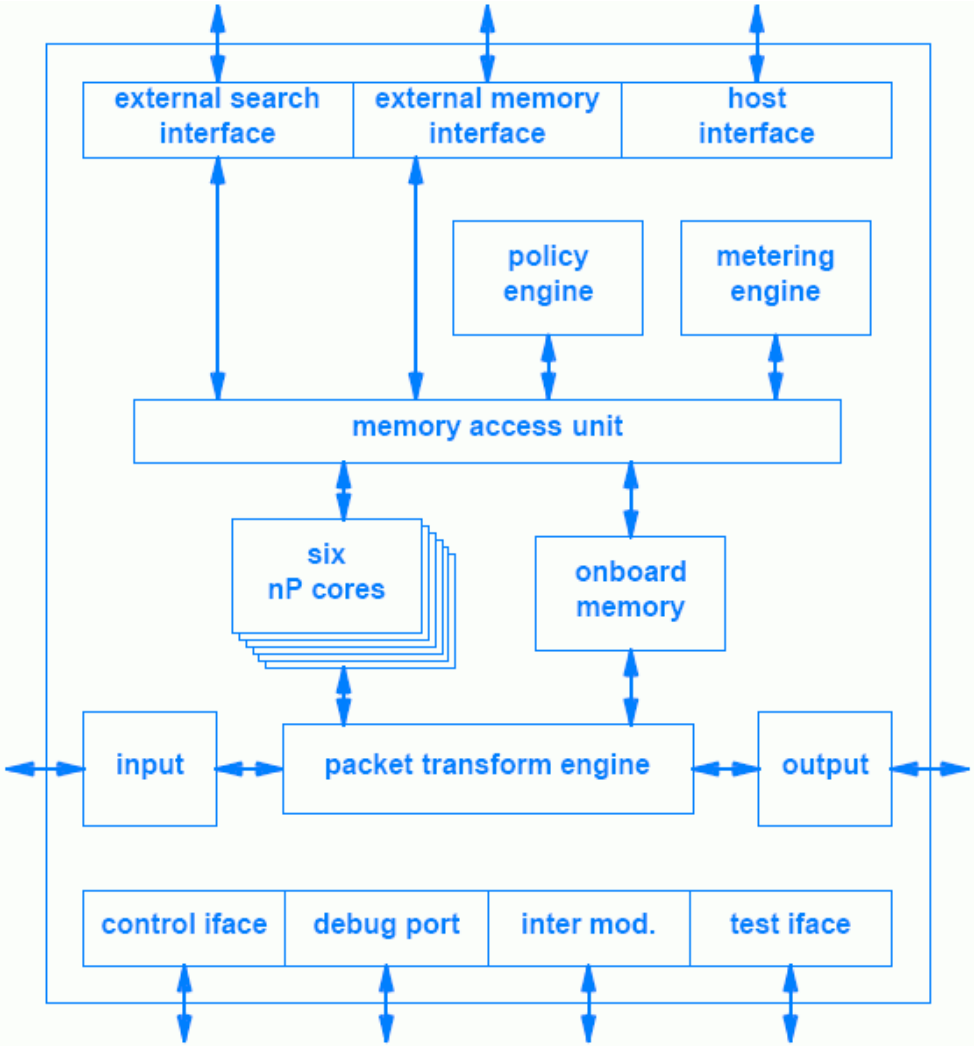
Agere


Fast
Pattern
Processor
(FPP)
Routing
Switch
Processor
(RSP)
packets
arrive
packets sent
to fabric
configuration bus
Agere
System
Interface
(ASI)

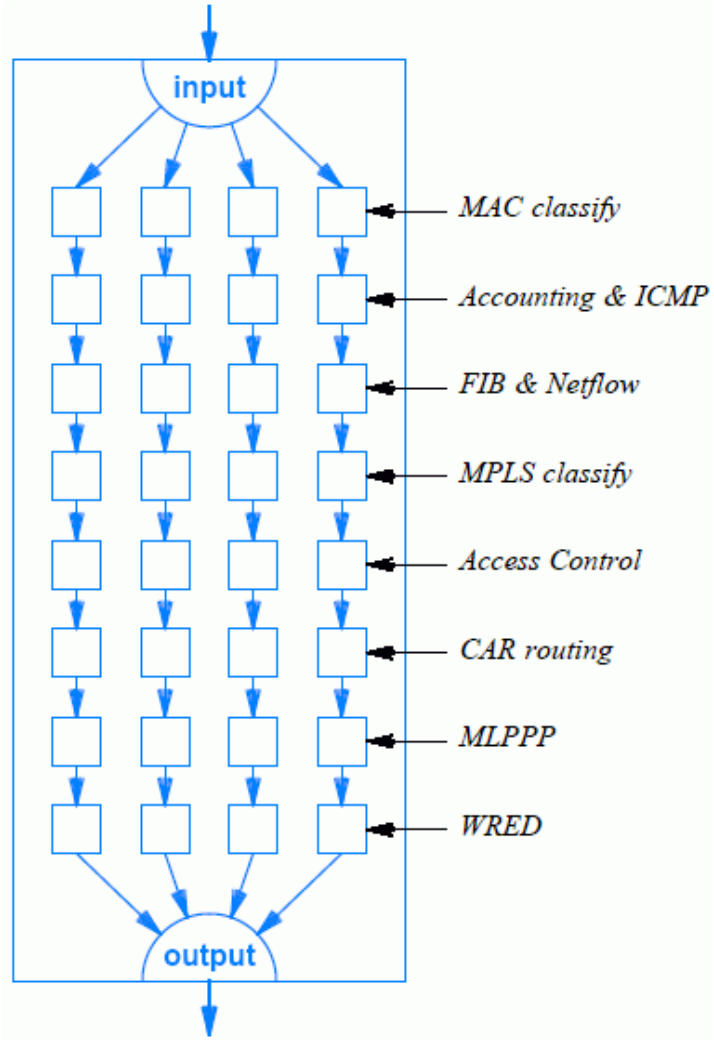
Alchemy Au1000



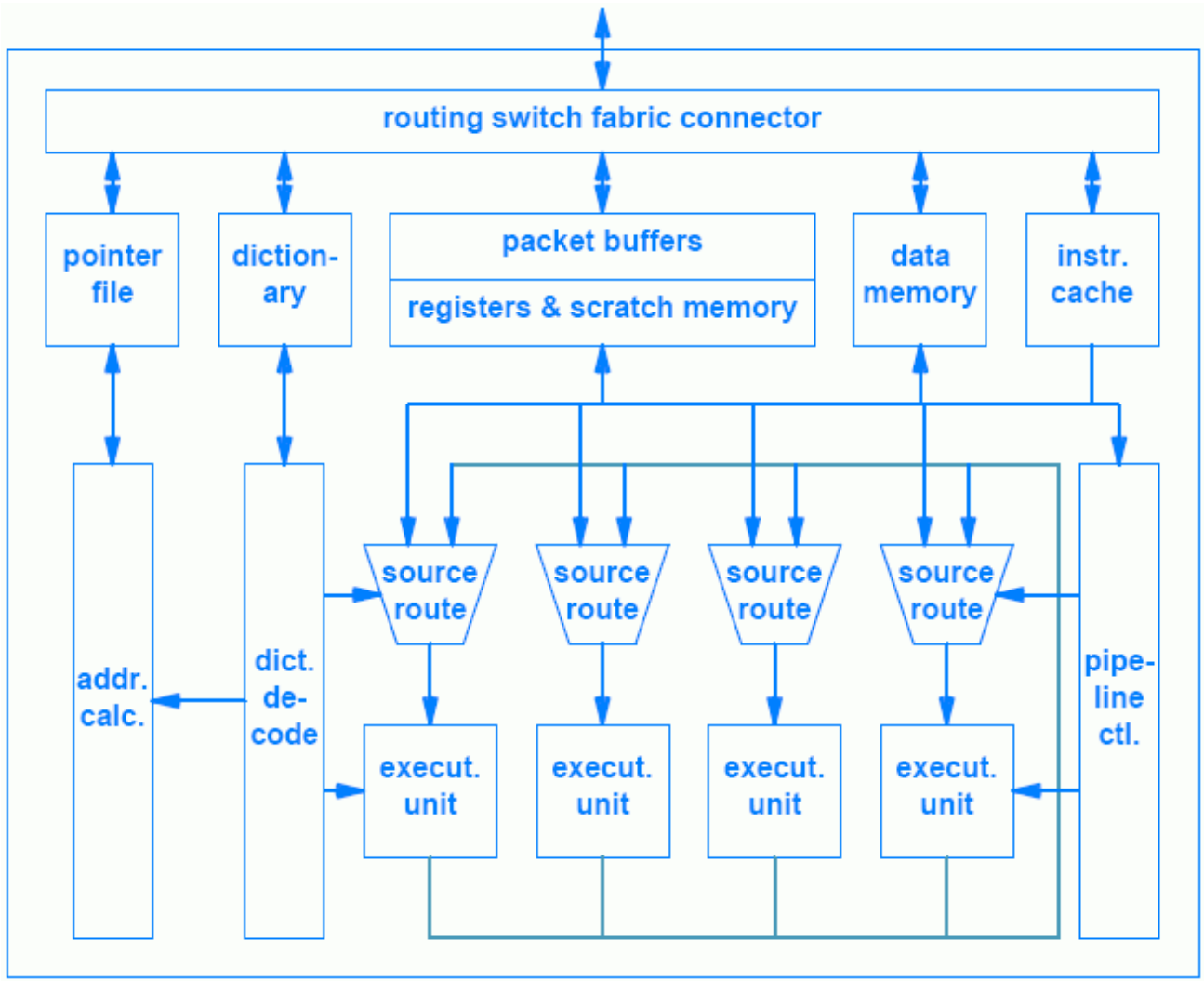
AMCC nP7510


external search interface
external memory interface
host interface
policy engine
metering engine
memory access unit
six nP cores
onboard memory
input
packet transform engine
output
control iface
debug port
inter mod.
test iface

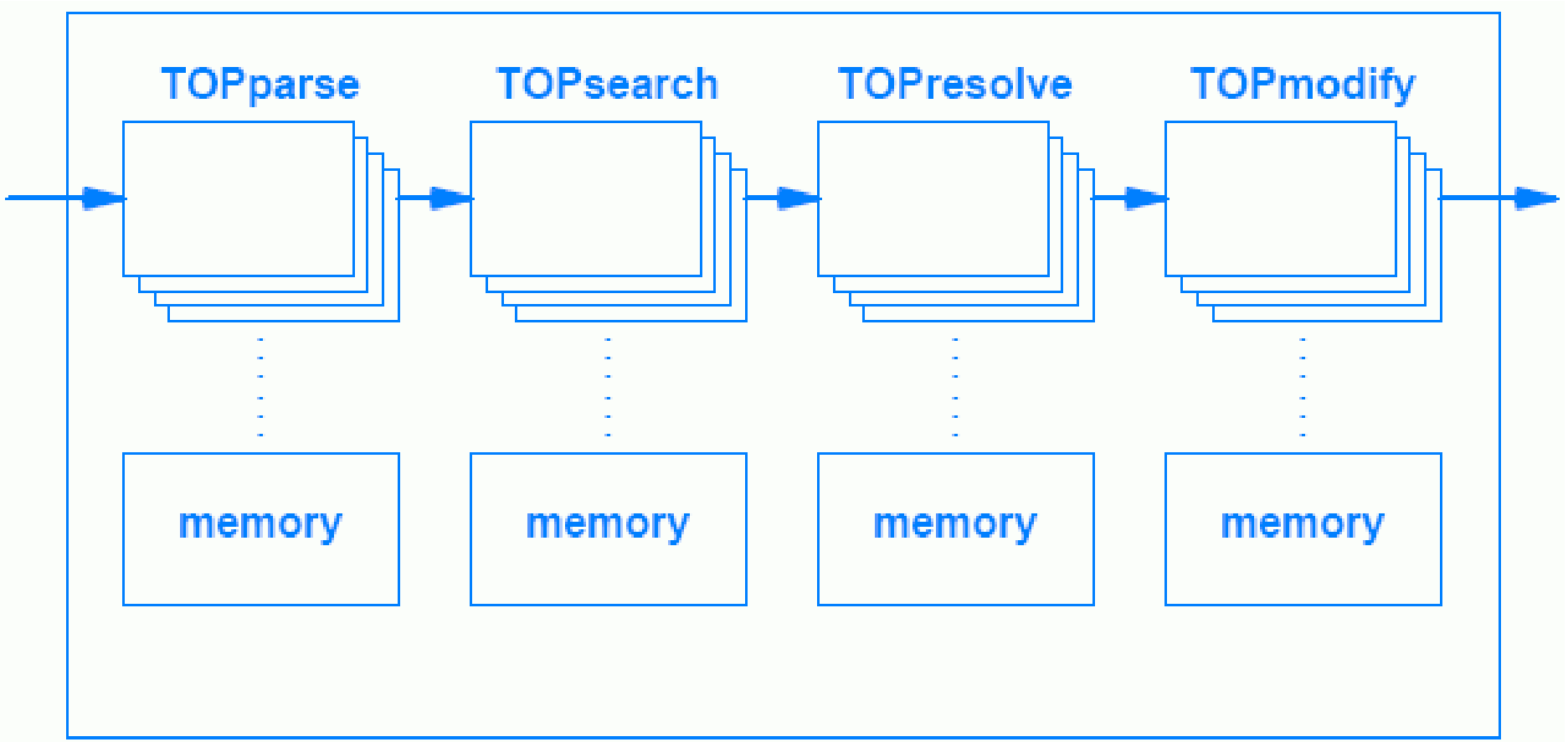
Cisco PXF


input
MAC classify
Accounting & ICMP
FIB & Netflow
MPLS classify
Access Control
CAR routing
MLPPP
WRED
output

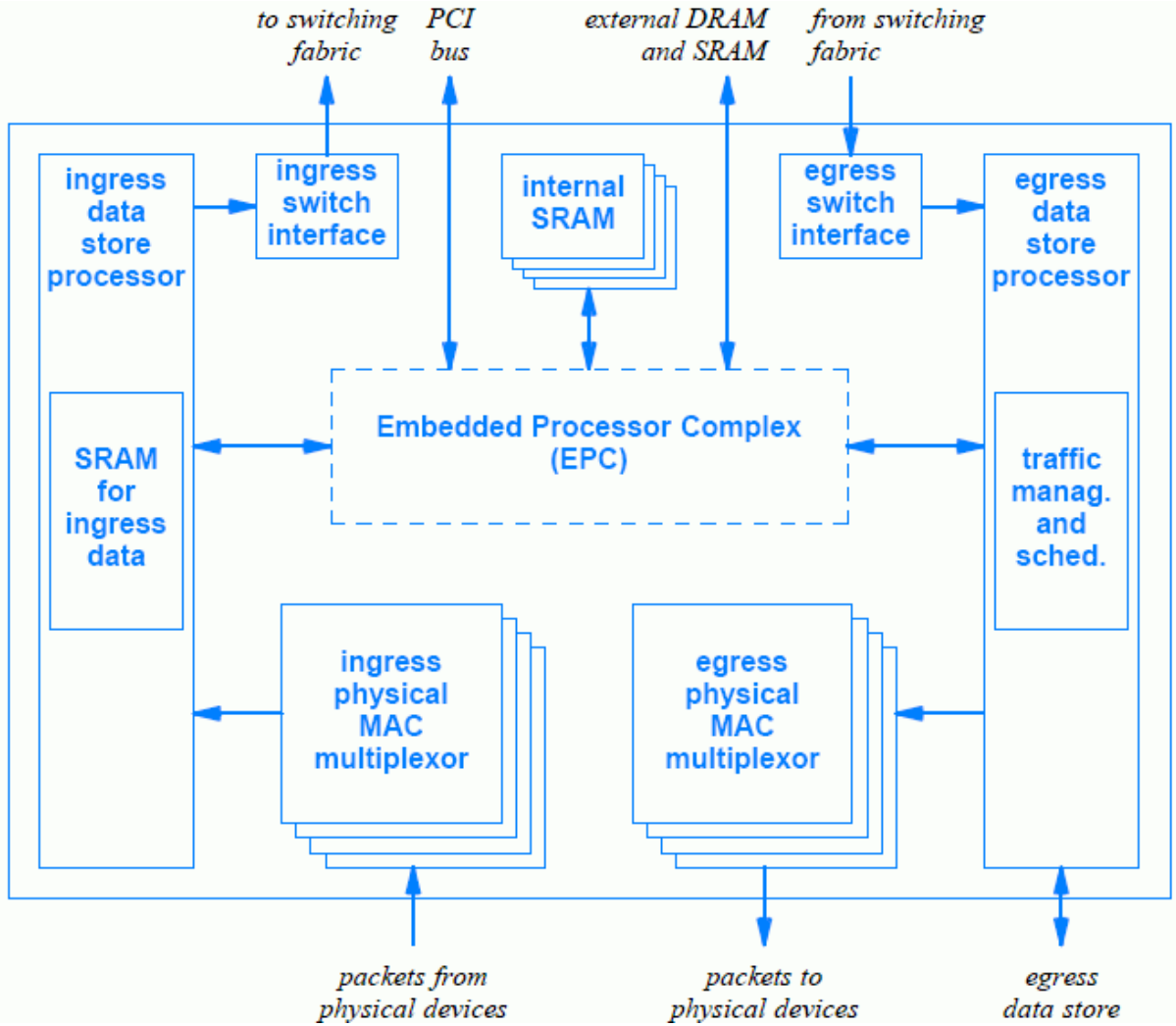
Cognigine RCU



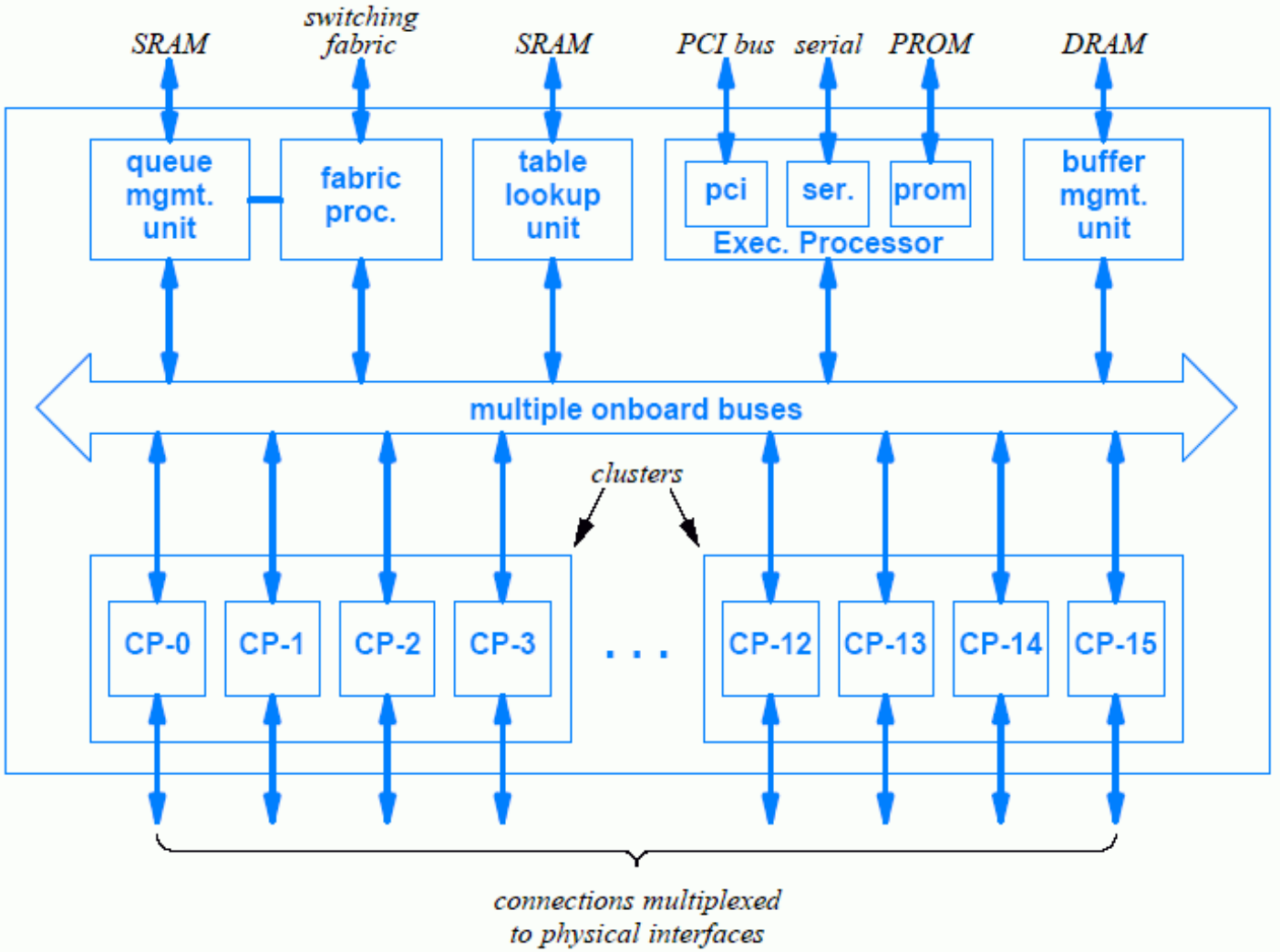
EZchip NP-1


TOPparse
TOPsearch
TOPresolve
TOPmodify
memory
memory
memory
memory

IBM PowerNP


to switching fabric
PCI bus
external DRAM and SRAM
from switching fabric
ingress data store processor
ingress switch interface
internal SRAM
egress switch interface
egress data store processor
SRAM for ingress data
Embedded Processor Complex (EPC)
traffic manag. and sched.
ingress physical MAC multiplexor
egress physical MAC multiplexor
packets from physical devices
packets to physical devices
egress data store

Motorola C-Port

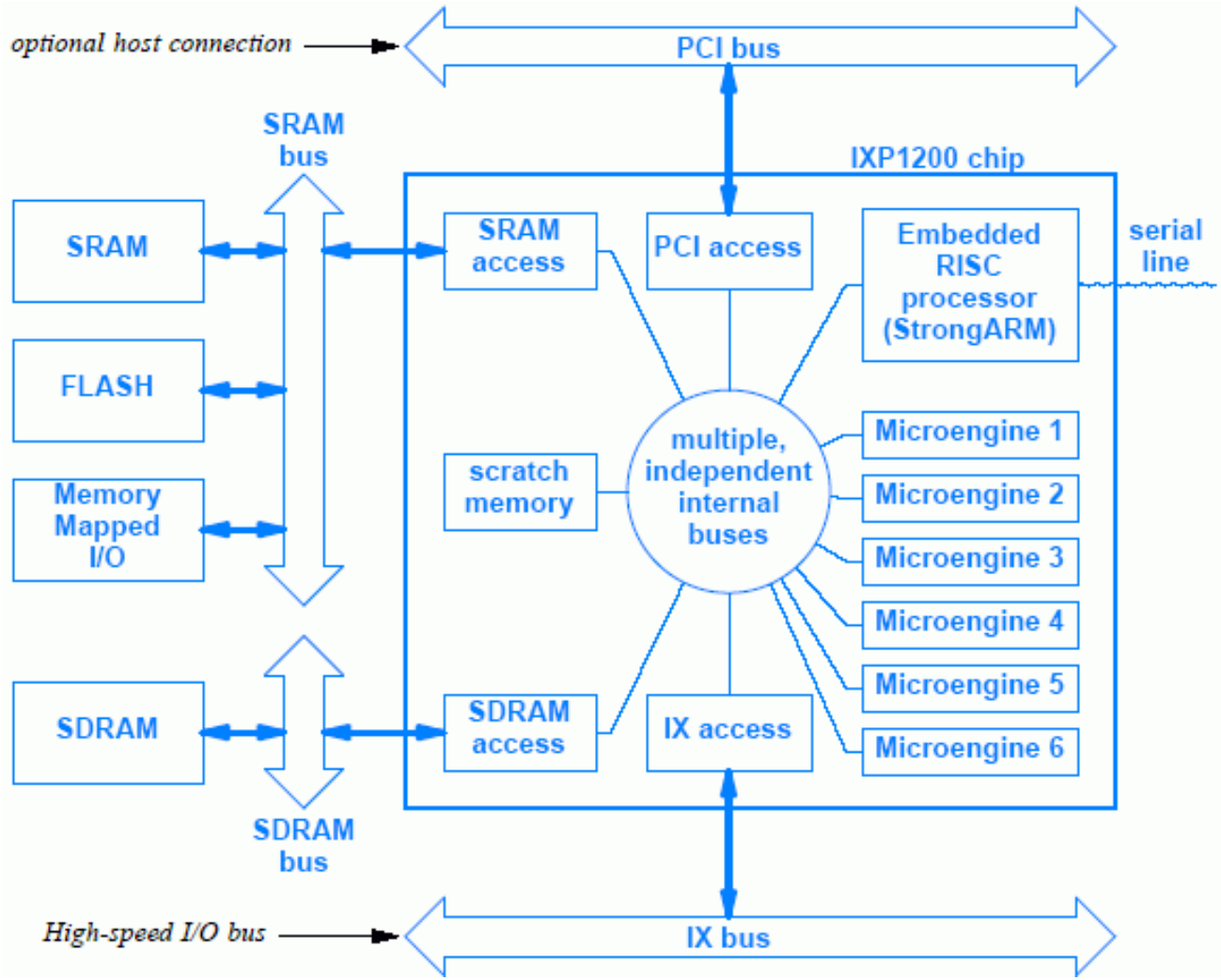


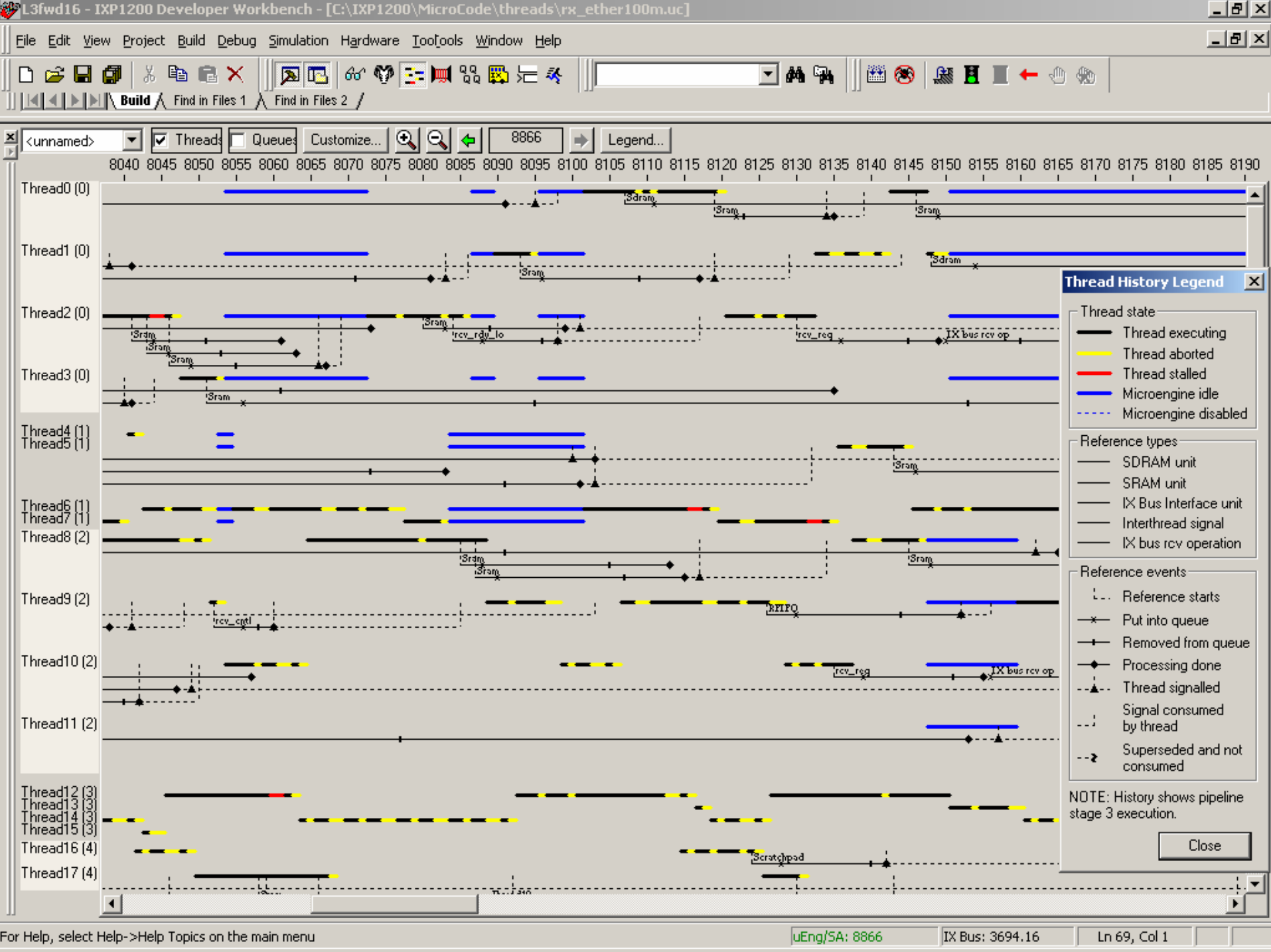
NP Architectures

- Numerous different design goals
  - Performance
  - Cost
  - Functionality
  - Programmability
- Numerous different system choices
  - Use of parallelism
  - Types of memories
  - Types of interfaces
  - Etc.
- We consider
  - Design tradeoffs on high level (qualitative tradeoffs)
  - Impact of different configurations on one particular architecture (quantitative tradeoffs)

IXP1200

StrongArm and uE Summary




L3fwd16 - IXP1200 Developer Workbench - [C:\IXP1200\MicroCode\threads\rx_ether100m.uc]
File Edit View Project Build Debug Simulation Hardware Tools Window Help
Build
Find in Files 1
Find in Files 2
<unnamed>
Threads
Queues
Customize...
8866
Legend...
8040 8045 8050 8055 8060 8065 8070 8075 8080 8085 8090 8095 8100 8105 8110 8115 8120 8125 8130 8135 8140 8145 8150 8155 8160 8165 8170 8175 8180 8185 8190
Thread0 (0)
Thread1 (0)
Thread2 (0)
Thread3 (0)
Thread4 (1)
Thread5 (1)
Thread6 (1)
Thread7 (1)
Thread8 (2)
Thread9 (2)
Thread10 (2)
Thread11 (2)
Thread12 (3)
Thread13 (3)
Thread14 (3)
Thread15 (3)
Thread16 (4)
Thread17 (4)
Sram
Sdram
rcv_rdy_lo
rcv_req
IX bus rcv op
rcv_cntl
RFIFO
Scratchpad
Thread History Legend
Thread state
Thread executing
Thread aborted
Thread stalled
Microengine idle
Microengine disabled
Reference types
SDRAM unit
SRAM unit
IX Bus Interface unit
Interthread signal
IX bus rcv operation
Reference events
Reference starts
Put into queue
Removed from queue
Processing done
Thread signalled
Signal consumed by thread
Superseded and not consumed
NOTE: History shows pipeline stage 3 execution.
Close
For Help, select Help->Help Topics on the main menu
uEng/SA: 8866
IX Bus: 3694.16
Ln 69, Col 1

ALU Operators

Operator	Meaning
+	Result is src_1 + src_2
-	Result is src_1 - src_2
B-A	Result is src_2 - src_1
B	Result is src_2
~B	Result is the bitwise inversion of src_2
AND	Result is bitwise *and* of src_1 and src_2
OR	Result is bitwise *or* of src_1 and src_2
XOR	Result is bitwise *exclusive or* of src_1 and src_2
+carry	Result is src_1 + src_2 + carry from previous operation
~AND	Result is bitwise (*not* src_1) *and* src_2
AND~	Result is bitwise (src_1 *and* (*not* src_2)
+IFsign	If the operation two instructions prior to the current operation caused the sign condition then the result is src_1 + src_2; otherwise the result is src_2
+4	Result is src_1 + src_2 with the first 28 bits set to zero
+8	Result is src_1 + src_2 with the first 24 bits set to zero
+16	Result is src_1 + src_2 with the first 16 bits set to zero

Next Generation NPs

- What market will they be used in?
  - Where can NPs make money?
- What should they look like?
  - Architectural features?
- What are current bottlenecks?
  - Performance limitations
- What features would be nice?
  - What functions need hardware support
- What are limitations on scalability?

The End

- This concludes the journey through the world of network systems

Thank You!

- Thanks for taking this course.
- Thanks for presenting papers.
- Thanks for doing projects.
- Thanks for participating in class discussions.

I hope you found this course interesting and you learned about networking concepts and systems in this exciting area of research.