Network Processors

Hassan Shojania

Agenda

- History
- Challenges, features and applications
- Example application/routing scenario
- NP architecture
- Case study: IXP2400
- Software
- Scalability & future

History

- First generation (1980s)
  - General purpose CPU, minicomputer → adaptable
  - Few connections/slow links
- Second generation (mid 1990s)
  - Increased speed and density
  - Specialized hardware functions
  - Offload of functions (e.g. classification) from CPU
- Third generation (late 1990s)
  - More and more specialized HW → ASIC
  - Decentralized: Multiple HW → complexity
  - Protocol consolidation: IP/Ethernet → less flexibility

Tradeoff: programmability for speed

Today’s story

- Convergence (voice/data/multimedia)
  - Faster pace of changes
  - New services/applications
- Shorter product lifecycle
- Fast time to market
- More complex
  - QoS, VPN, MPLS
  - Not store-and-forward anymore
  - Encryption, compression, classification
    - Two order of magnitude

➔ Programmability needed again (1st gen. hallmark)

➔ But at high performance (3rd gen. hallmark)

Design questions

- What most important tasks to optimize?
- What HW-assist units to include?
- What I/O interface needed?
- What size instruction/data store needed?
- What memory tech., interconnects?
- What level software support? languages/tools,...

Many *possibilities* → Many solutions

More than 30 NP vendor by Jan. 2003!

Not many around these days ☺

MSOffice8

Slide 5

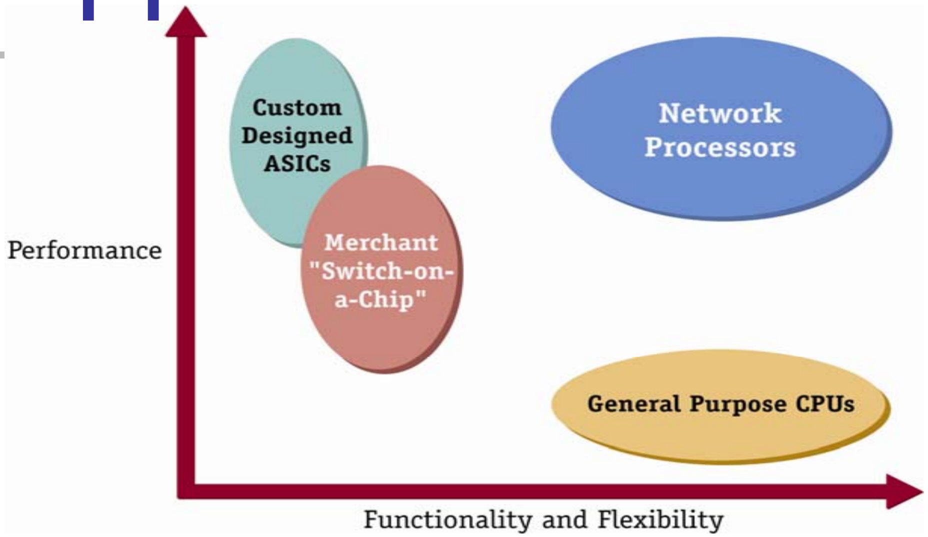
MSOffice8 Ask Prof. Leon-Garcia about it!

, 3/7/2006

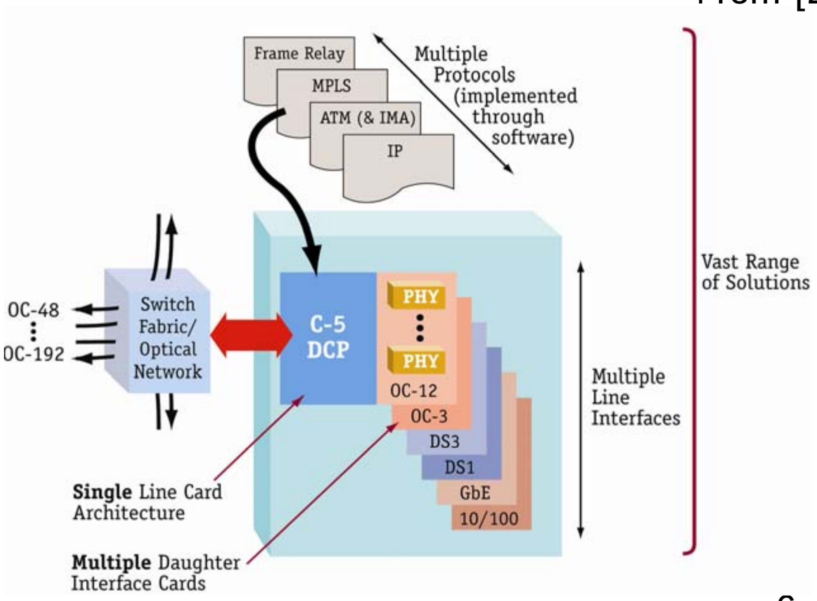
MSOffice1

NP: The new approach

- Key attributes
  - Programmability
  - Simple prog. model
  - Maximum flexibility
  - High processing power
  - High functional integration
  - Open prog. interface
- Universal applicability
  - Interface/protocol range
  - Programmability at all levels



From [2]


Frame Relay
MPLS
ATM (& IMA)
IP
Multiple Protocols (implemented through software)
OC-48
OC-192
Switch Fabric/ Optical Network
C-5 DCP
PHY
PHY
OC-12
OC-3
DS3
DS1
GbE
10/100
Multiple Line Interfaces
Vast Range of Solutions
Single Line Card Architecture
Multiple Daughter Interface Cards

Slide 6

MSOffice1 Bringing both flexilibilty and performance -- point to the axes

System designers focusing on higher-level service rather than constant change

Benefits:
- universal networking applications
- faster time-to-market
- longer time in market --> focus on high-level service
- Scalable performance
- Lower system cost
- Higher availability
- Continous innovation

All level of protocol stack: 2-7

, 3/4/2006

MSOffice2

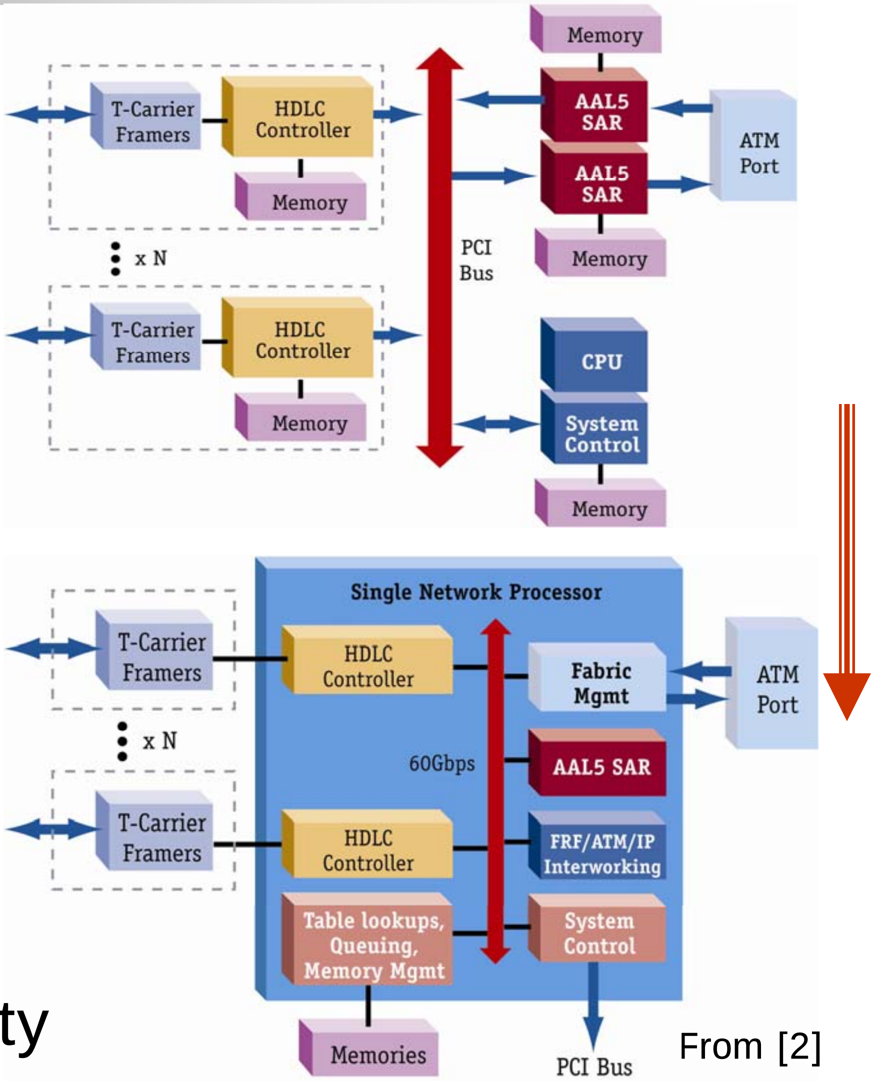
Towards integrated system

- Coprocessor engines
  - Bottlenecks
  - Classification/queuing
- Lower-level functions
  - SONET framer
  - Higher port density

➔ Lower cost

➔ Higher Performance

Lower interconnection penalty



From [2]

Slide 7

MSOffice2 Similarly: software development cycle is reduced across product generations through "Stable Programming Interfaces"

, 3/4/2006

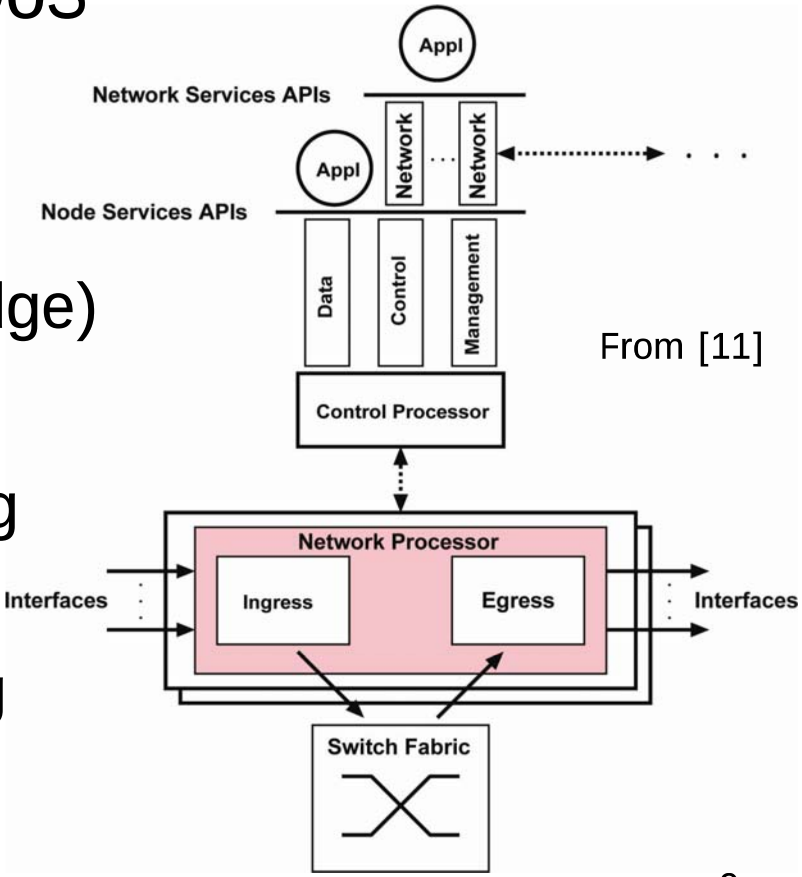
What’s an NP?

- Packet processing/forwarding
- Compared to a GPP
  - Simpler arithmetic/caching
  - Multiple execution threads → Parallel packet processing
  - Special functional units
- Location in network:
  - Edge: Intelligent stateful processing
  - Core: Aggregated traffic flows
- Tuned towards:
  - Control-plane: Sys. mgmt., routing updates, protocol mgmt.
  - Data-plane: Packet processing (general term)

MSOffice3

Applications

- Load balancing → distributing overload to servers
- Traffic differentiation → QoS
- Network security
- Terminal Mobility
  - Tunneling and bundling (edge)
- Active networking
  - No more passive forwarding
  - Code carried in packets
  - More data-plane processing
  - Exposing router state

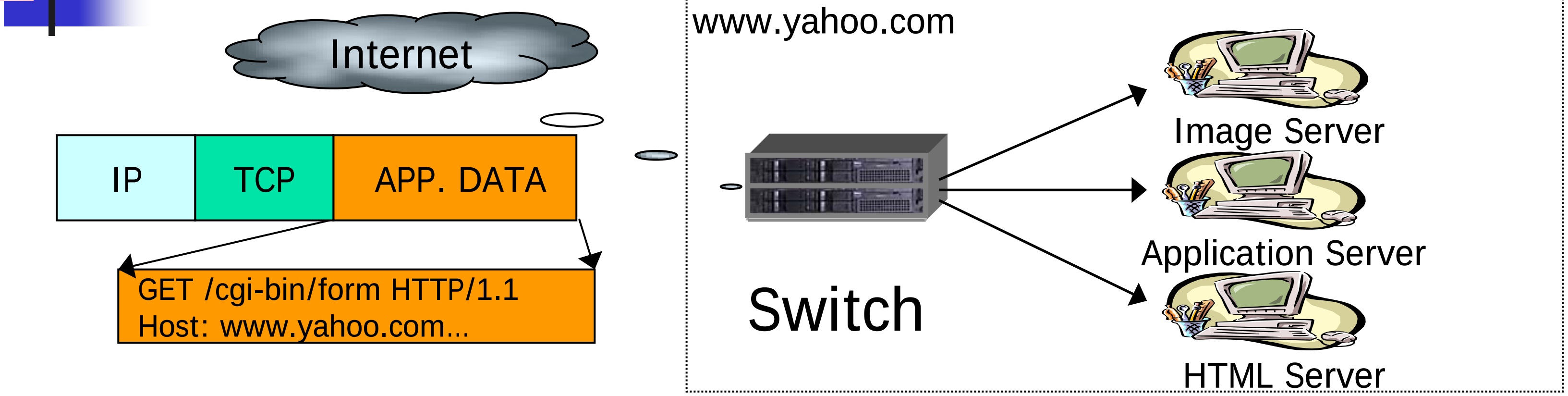


From [11]

Slide 9

MSOffice3 convergence of mobile & IP
3GPP and SCTP(stream control transmission protocol)
, 3/4/2006

Example app: Content-aware switch



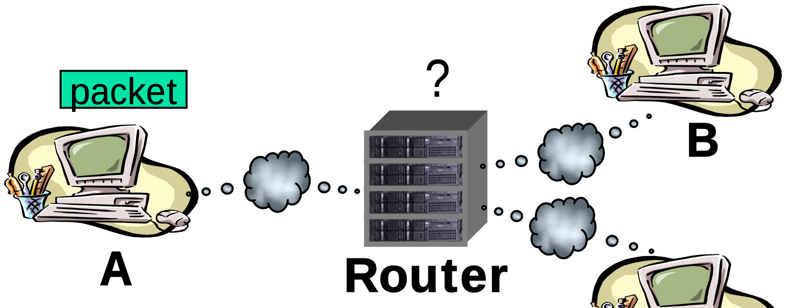
From [5]

- Web-server front-end
- One virtual IP
- Examining above TCP/IP layer (layer 5)
- Advantageous:
  - Better load balance: distributed
  - Faster response: caching
  - Better resource management: database partitioning

MSOffice6

Packet processing

- Different from normal processing
  - I/O centric vs. processing centric
  - Real-time vs. best-effort
  - Many simple tasks vs. few complex tasks
  - State: per-flow vs. per program
  - Buffering: dependence in flow, e.g. CRC calculation in ATM
  - Atomic context process (seq. packets)
- Router functions
  - Packet receive
  - Route-table lookup
  - Classification
  - Metering
  - Congestion avoidance
  - Packet scheduling
  - Packet transmit



C

From [4]

Slide 11

MSOffice6 Buffering size might grow: internal/external SDRAM/DRAM Memory speed

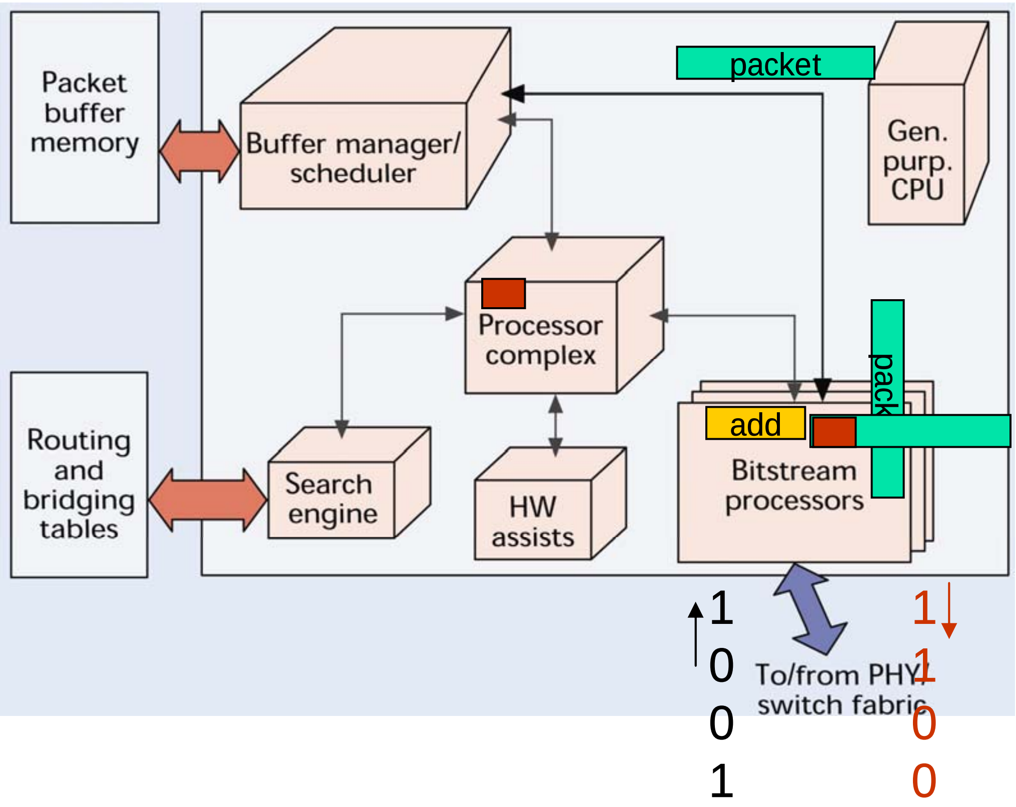
, 3/5/2006

MSOffice7

Routing example

- Serial stream in
- Packet (framing)
- Target extraction
- Packet buffering
- Packet processing
- Router lookup
- Scheduling
- Transmission
- Packet update
- Serial stream out

From [3]



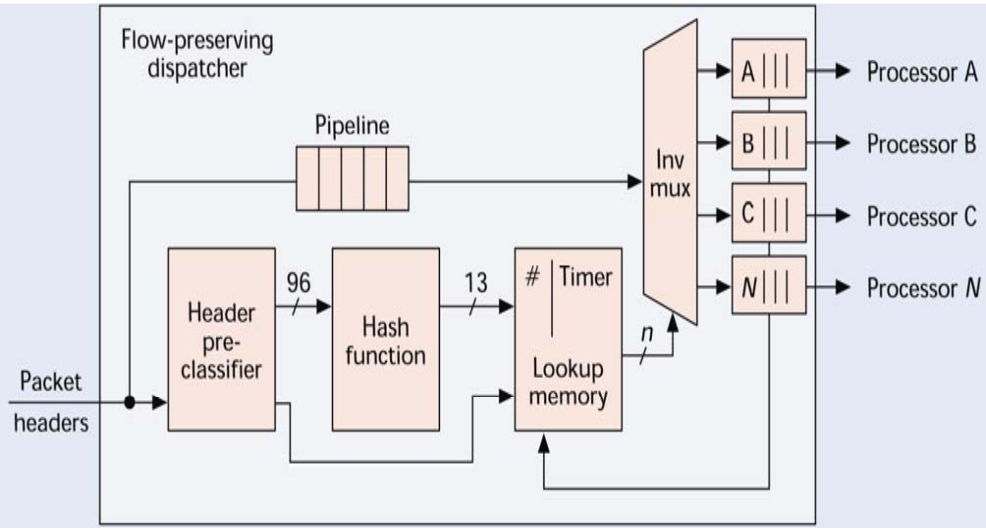
Slide 12

MSOffice7 Packet classification?

, 3/5/2006

Ex: Load-balancing dispatcher

- Packet → NP
- Preserving flow order
  - TCP fast retransmit
  - BW waste
- Flow state
  - Shared memory
  - SMT/CMP NP



From [3]

- Flow classification problem
  - Assigning each flow to a fixed NP
    - Flow identity: src/dst IP and port, transport protocol ID
  - Other app: firewall, NAT, network monitoring
  - Eliminating inter-NP synchronization

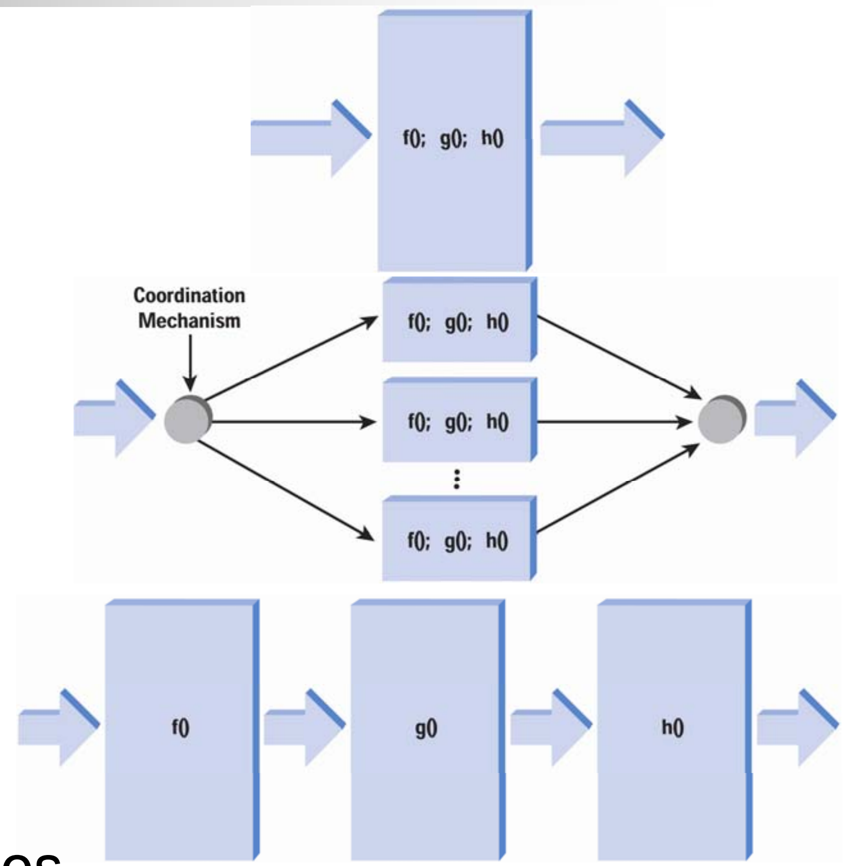
Parallelism

- Single processor
- Run-to-completion

- Packet interdependence
- Parallel processing

- Pipelining

- Typical processor design issues
  - Superscalar/pipelined processors: e.g. P4
  - Higher clock with higher pipelining degree
  - If dependency exists?



From [1]

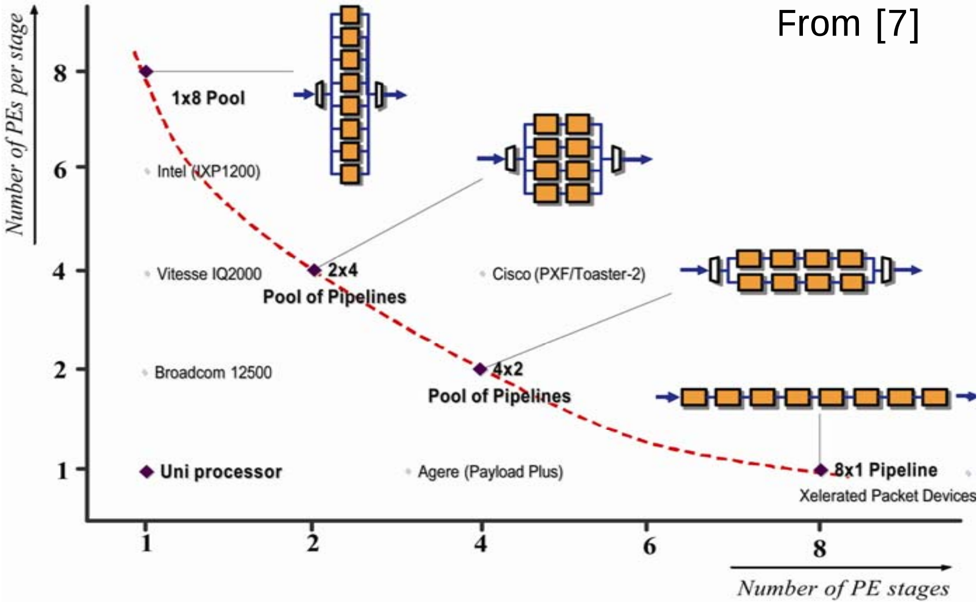
Slide 14

MSOffice9 Simplistic view

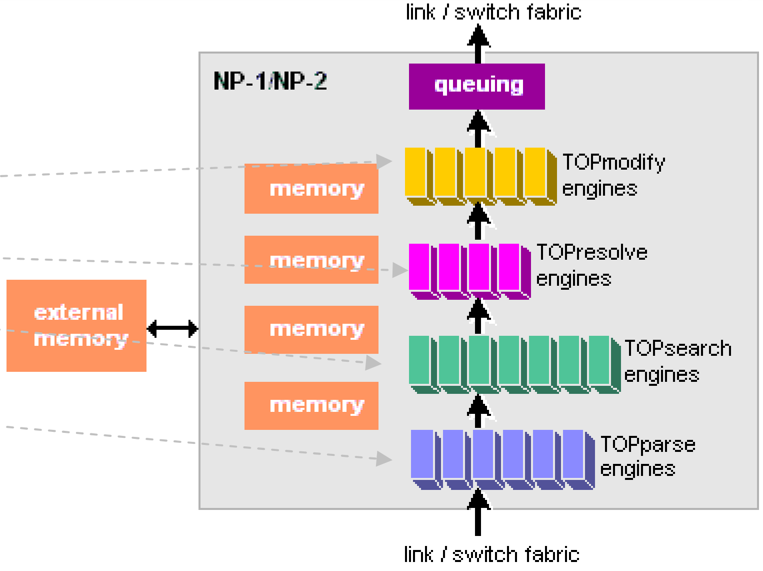
, 3/8/2006

Design space

- Homogenous PEs
  - Fully pipelined
  - Fully pooled
- Heterogeneous PEs
  - EZchip NP-1/2
  - Task Optimized Proc.
    - Packet modification
    - Lookup and classification
    - Forwarding and QoS decision
    - Packet parse



From [7]



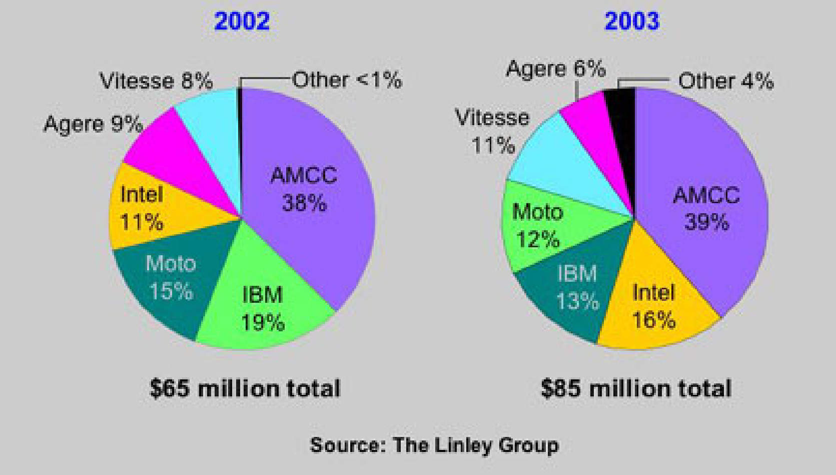
Memory

- Issues
  - Managing 1000s of flows
  - Packet buffering/queuing
  - Complex packet processing: e.g. Encryption
  - Several access to packet
- And all at wire-speed!
- Memory types:
  - SRAM for data structures (memory mgmt. pointers)
  - DRAM for packet buffers
  - Using on-chip cache rather than off-chip memory
  - Inter-PE communication; synchronization
- Specialized memory management/queuing blocks

MSOffice5

Market

- 2005: $174 million   2003 estimate in 2001: $1B
- 1st: AMCC, nP37X0: single-chip OC-48/traffic mgr.
- 2nd: Intel, IPX2800/2400: flexible, software, power
- 3rd: Agere, Hifn, Wintegra
- EZchip: startup, 5% market share, 10Gbps
- Mainstream: OC-48 and up; metro-Ethernet switches
- Niche market: Hifn: PowerNP → security, VPN



Source: The Linley Group

Slide 17

MSOffice5 2.5Gbs/10Gbs/40Gbs OC 48-192-768

Intel: 10Gb/s: OC-48

, 3/5/2006

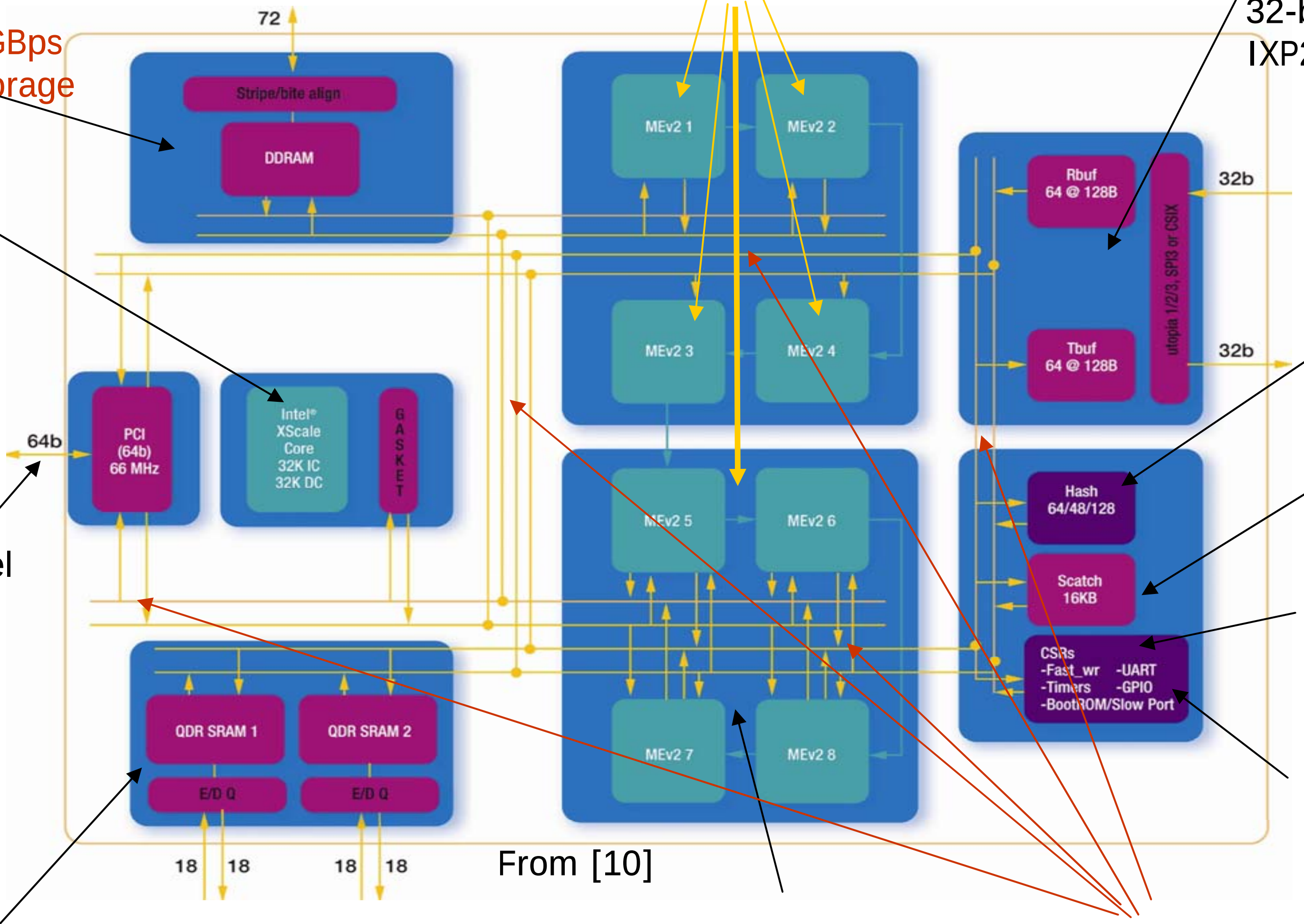
Intel IXP product line

- IXP4xx series
  - For home, small-to-medium enterprise level
    - Wireless access point, router, DSL, VoIP , ...
    - LinkSys, DLink, Netgear routers
- IXP12xx series
  - 1st generation NP
  - OC-12 applications
- IXP2xxx series
  - 2nd generation NP
  - OC-192 applications
  - For high-performance, and scalable network
    - Edge and core applications from T1/E1 to OC-192

IXP2xxx series

- Integrated XScale RISC proc. (ARM-based)
  - For control-plane tasks
- IXP2400
  - OC-48 – 2.5Gbps
  - Single chip packet forwarding/traffic management
  - 8 micro-engines, 4K word inst. Memory @ 600MHz
  - 5.4 Giga op/s
- IXP28xx
  - OC-192 – 10Gbps
  - 16 MEs with 8K inst. Memory @ 1.5GHz
  - 24 Giga op/s
  - More SDRAM/DRAM channels: 4 and 3 vs. 2 and 1 respectively
- IXP2855
  - Specialized cryptography engines (DES, AES, SHA)

IXP2400 building blocks



2 GB
150Mhz, 2.4GBps
Data buff. storage

32-bit ARM
600 Mhz
Chip mgmt

Host proc.
DMA channel
PCI arbiter

16MB/channel,
200Mhz, 1.6GBps
Control info. storage

Soft-pipelining

32-bit multithreaded
600 Mhz, 4K inst.
Data-plane proc.

From [10]

Internal bus

PHY device (framer) or
fabric switch interface
32-bit, 133 MHz
IXP2400/CSIX interface

Hash accel.
(HW assist)

16KB memory
general purpose

Control/Status reg.
inter-ME/XScale
communication

XPI (peripheral
interface: interrupt
ctlr., GPIO, ...)

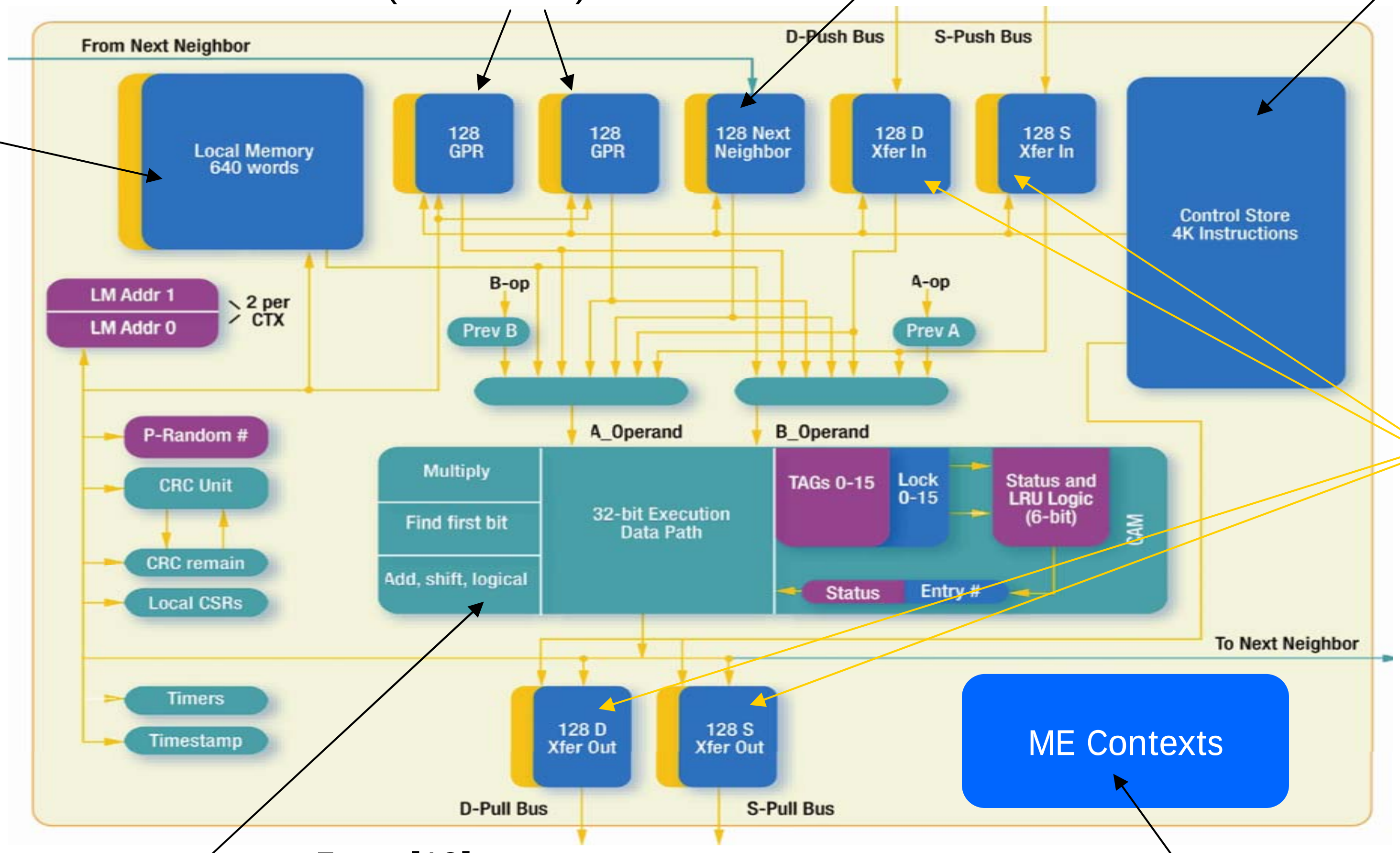
Micro-engine

General purpose reg. (bank A & B)

Next neigh. reg. src/dst in program

Inst. mem. (program) Initialized by XScale

(exclusive to ME)



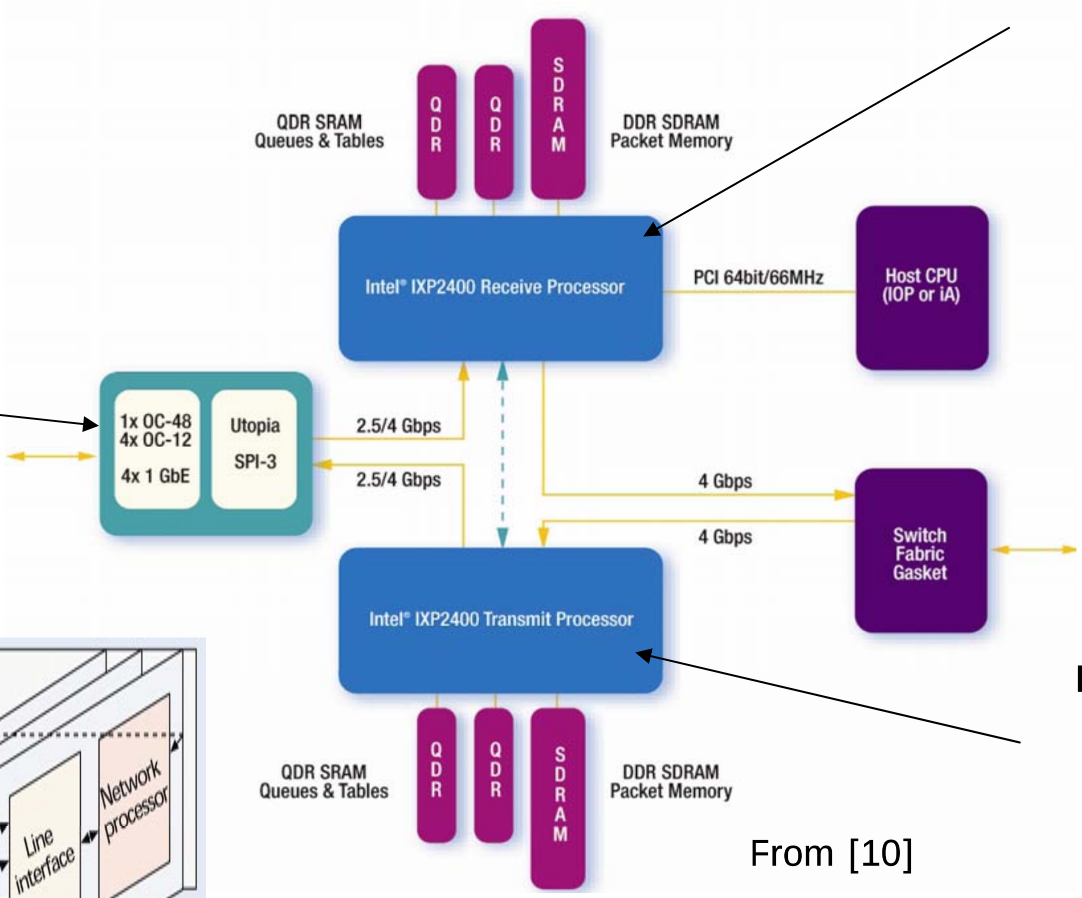
Transfer reg. in/out (RAMs)

From [10]

Execution unit

8-way multithreaded
Own reg. set (PC, …)

IXP2400 Line card

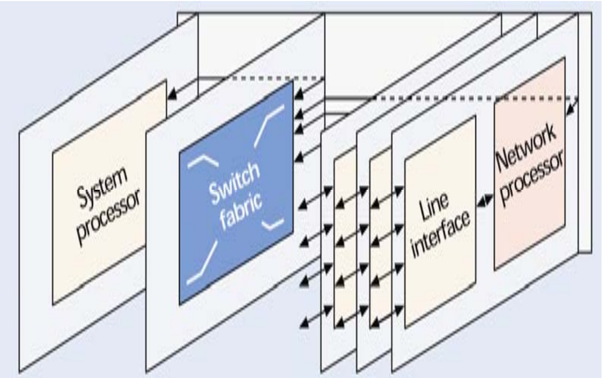


Ingress Processor
SAR'ing
Classification
Metering
Policing
Initial Congestion
Management

eg. Intel IXF6048 framer

Egress Processor
Traffic shaping
DiffServ

From [10]



From [3]

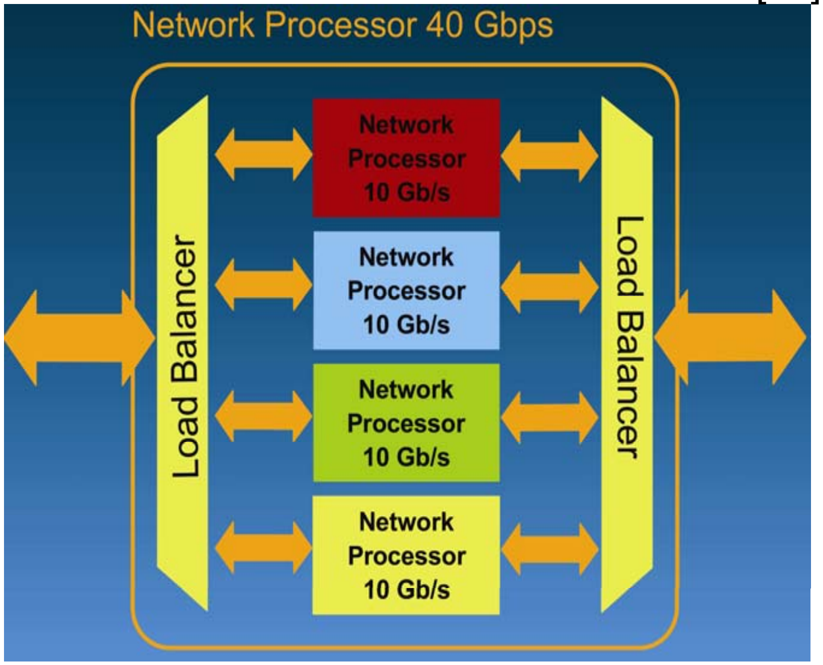
IXP2400 SDK

- Software is critical!
  - NP: all about programmability
- Runtime environment
  - VxWorks, Embedded Linux, QNX Neutrino
- Tools
  - Microcode assembler, Microengine C Compiler and C Runtime Library, cycle-accurate simulator, Architecture Tool, ...
- Data-plane Libraries:
  - Microcode and Microengine C versions of: Hardware Abstraction Library, Protocol Library, Cryptography Library (IXP2850), Utility Library, and Microblock Infrastructure Library

Scalability & future

- OC-768 router
  - Load balancing
  - Concurrent NPs
  - Without inter-proc. comm.
- Future trend
  - Internet is booming
    - OK! Not at the old pace
  - Nodes are more BW hungry
  - New services and applications
    - Not following OSI model
  - More complex & upper-layer task at edge/core
  - ➔ More powerful NP; standard HW/SW interfaces
    - More like CPU trend

From [13]


Network Processor 40 Gbps
Load Balancer
Network Processor 10 Gb/s
Network Processor 10 Gb/s
Network Processor 10 Gb/s
Network Processor 10 Gb/s
Load Balancer

Software examples

- Sample IXP2xxx code for NAT
  - http://www.npbook.cs.purdue.edu/intel/code/NAT_pkt_handler.c.txt
  - http://www.npbook.cs.purdue.edu/intel/code/NAT_microblock.uc.txt
- Intel SDK 4.2

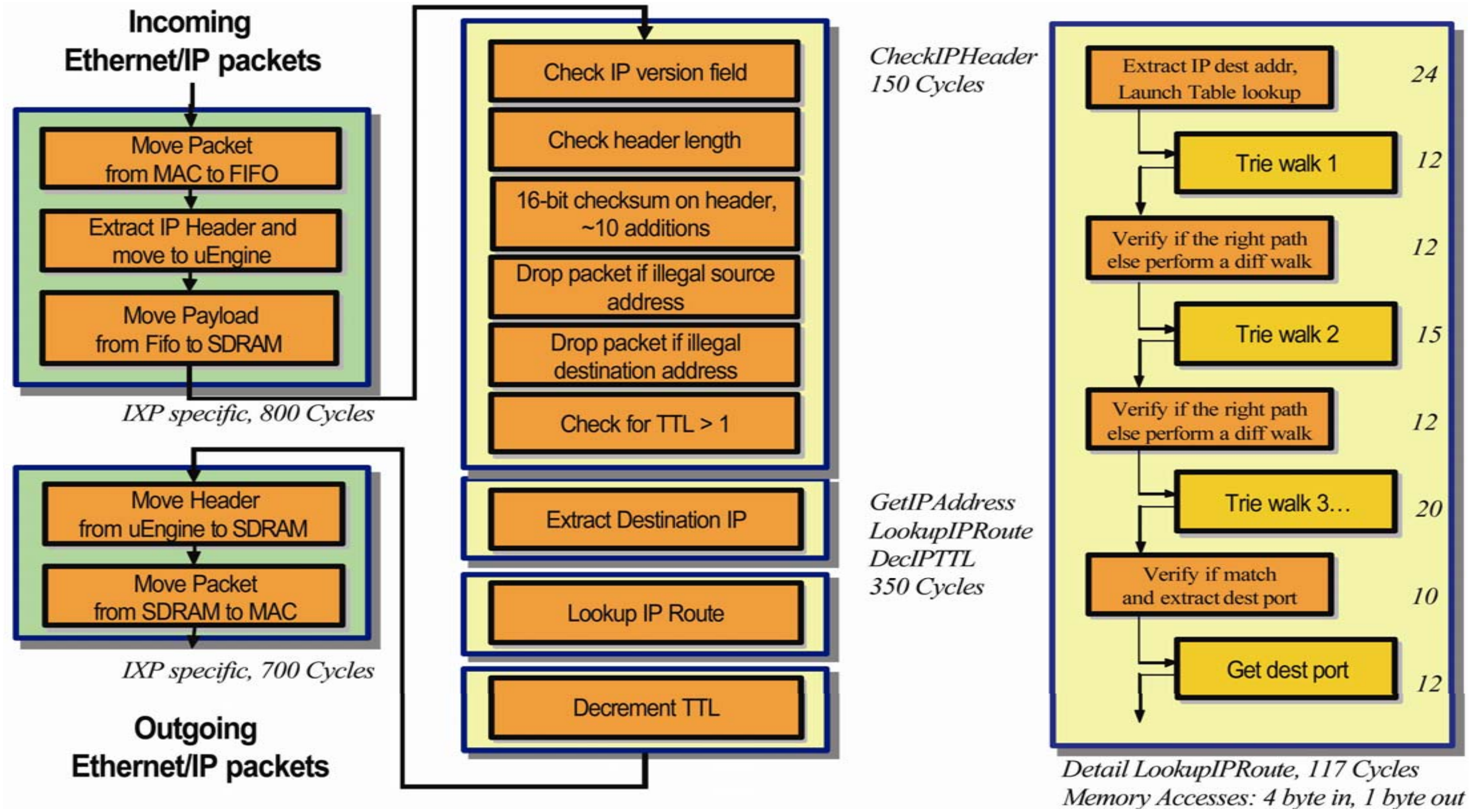
References

- [1] Douglas Comer, “Network Processors: Programmable Technology for Building Network Systems”, The Internet Protocol Journal, Vol. 7, No. 4.
- [2] David Husak, “Network Processors: A Definition and Comparison”, Freescale Semiconductor.
- [3] Werner Bux, et al., “Technologies and Building Blocks for Fast Packet Forwarding”, IEEE Communications Magazine, Jan. 2001.
- [4] Yan Luo’s slides, Network Processor and Its Applications.
- [5] Network Processor Tutorial in Micro 34-Mangione-Smith & Memik.
- [6] Intel Corp., “Next Generation Network Processor Technologies”, 2001.
- [7] Matthias Gries, “Exploring Trade-offs in Performance and Programmability of Processing Element Topologies for Network Processors”, 9th International Symposium on High Performance Computer Architecture (HPCA9), Feb. 2003.
- [8] Intel Corp., “IXP2400 Network Processor Datasheet”.
- [9] Intel Corp., “IXF6048 Multi-Speed SONET Packet Framer Product Brief“.
- [10] Intel Corp. “IXP2855 Product Overview”.
- [11] Andreas Kind , “The Role of Network Processors in Active Networks”, IBM Zurich Research Lab, 2003.
- [12] The Linley Group, “A Guide to Network Processors for Metro Applications”, 7th edition, Dec. 2005. (only its summary is publicly available)
- [13] Patricia Sagmeister, “Scaling Network Processor Performance to 40 Gbps”, IBM Research Zurich.

IPv4 task graph

- Sample packet flow in IXP

From [7]



Incoming Ethernet/IP packets

Move Packet from MAC to FIFO

Extract IP Header and move to uEngine

Move Payload from Fifo to SDRAM

IXP specific, 800 Cycles

Check IP version field

Check header length

16-bit checksum on header, ~10 additions

Drop packet if illegal source address

Drop packet if illegal destination address

Check for TTL > 1

CheckIPHeader 150 Cycles

Extract Destination IP

Lookup IP Route

Decrement TTL

GetIPAddress LookupIPRoute DecIPTTL 350 Cycles

Move Header from uEngine to SDRAM

Move Packet from SDRAM to MAC

IXP specific, 700 Cycles

Outgoing Ethernet/IP packets

Extract IP dest addr, Launch Table lookup *24*

Trie walk 1 *12*

Verify if the right path else perform a diff walk *12*

Trie walk 2 *15*

Verify if the right path else perform a diff walk *12*

Trie walk 3… *20*

Verify if match and extract dest port *10*

Get dest port *12*

Detail LookupIPRoute, 117 Cycles
Memory Accesses: 4 byte in, 1 byte out

MSOffice4

Active networks

- Decouples network service/infrastructure
- Active packets
  - Carry code (reference or directly)
- Active nodes
  - Execution environment like a VM; byte-code (JIT)
  - Access to node resource (link, routing table)
- App-level filtering:
  - Dropping B-frames in multicast tree
- Network management:
  - node params; aggregating several managed nodes

Slide 28

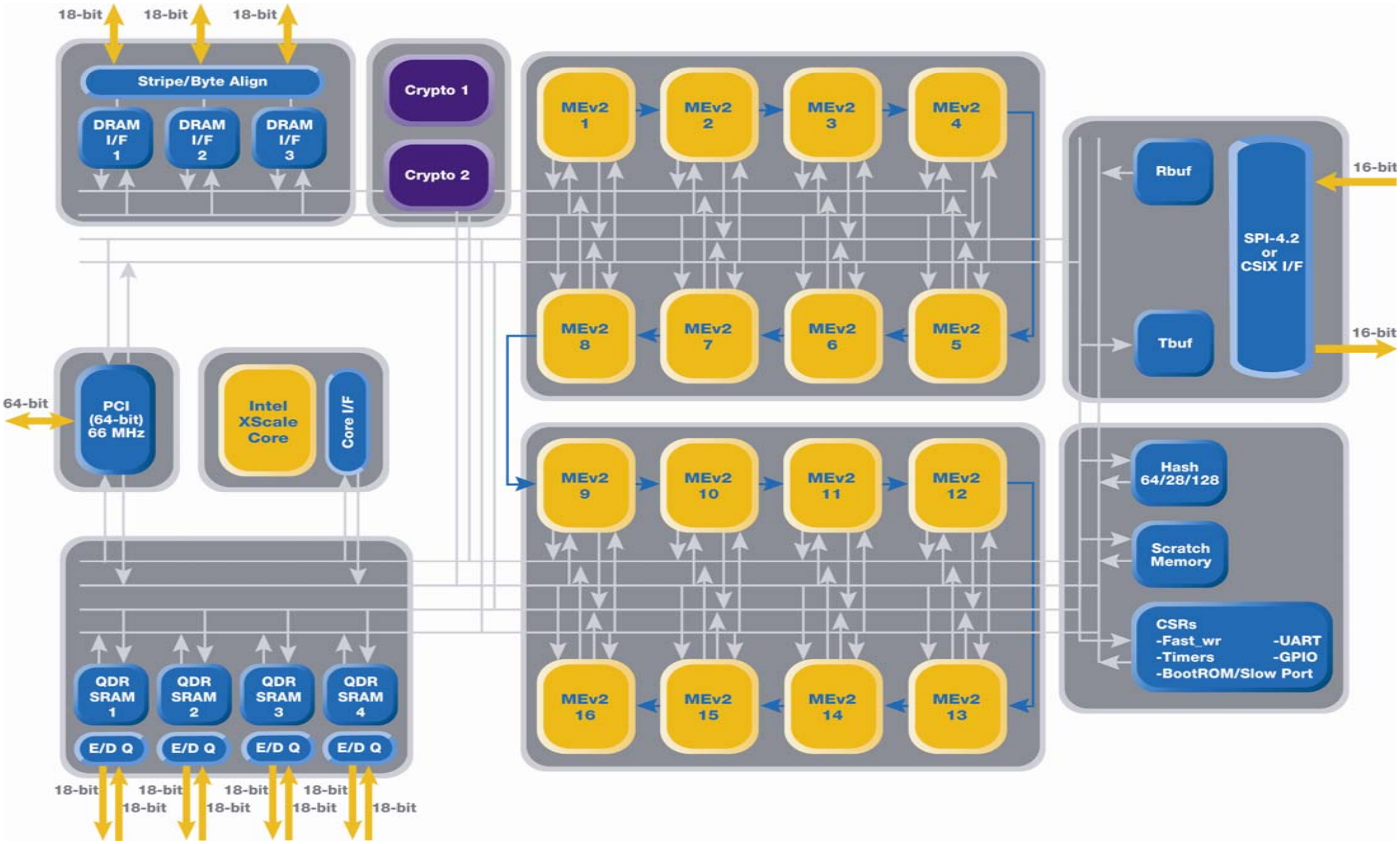
MSOffice4 Active packet source:
 - end-user
 - active gateways
 - network management app

in p2p could sense congestion and adapt
JIT: compiling once: storing in native binary format
 , 3/4/2006

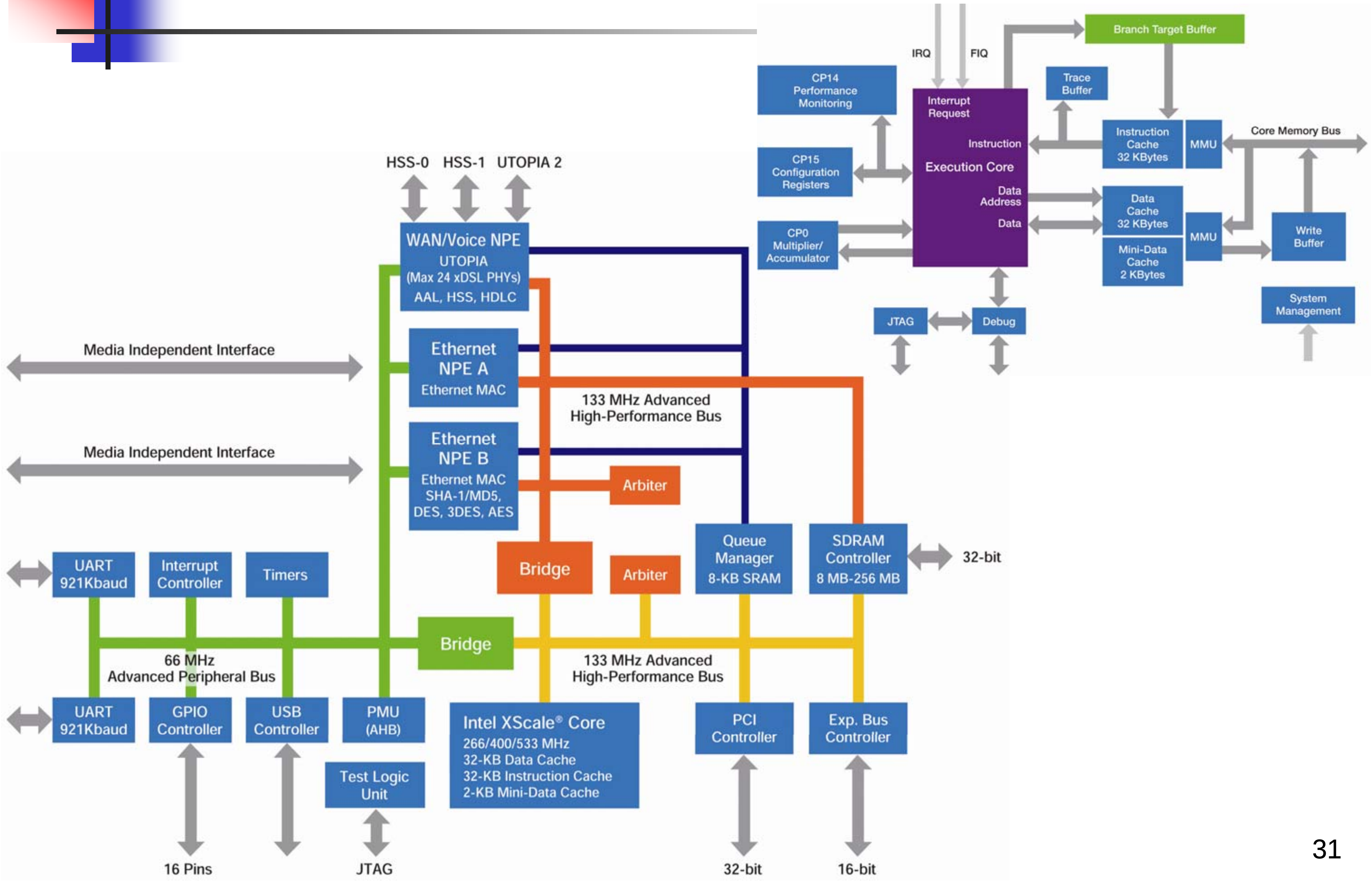
IXP2400 ME hardware assists

- Multiplier:
  - To improve performance and code density for QoS algorithms
- A pseudo random number generator:
  - To accelerate congestion avoidance algorithms like WRED
- Cyclic Redundancy Check (CRC) generator:
  - To automate CRC generation for ATM AAL5, Ethernet, Frame Relay, ...
- 16-entry Content Addressable Memory (CAM):
  - To efficiently share data among ME threads
  - To reduce memory bandwidth consumption
- 64-bit local timer:
  - To enhance traffic scheduling and shaping
- Memory features:
  - To accelerate updates to shared memory locations

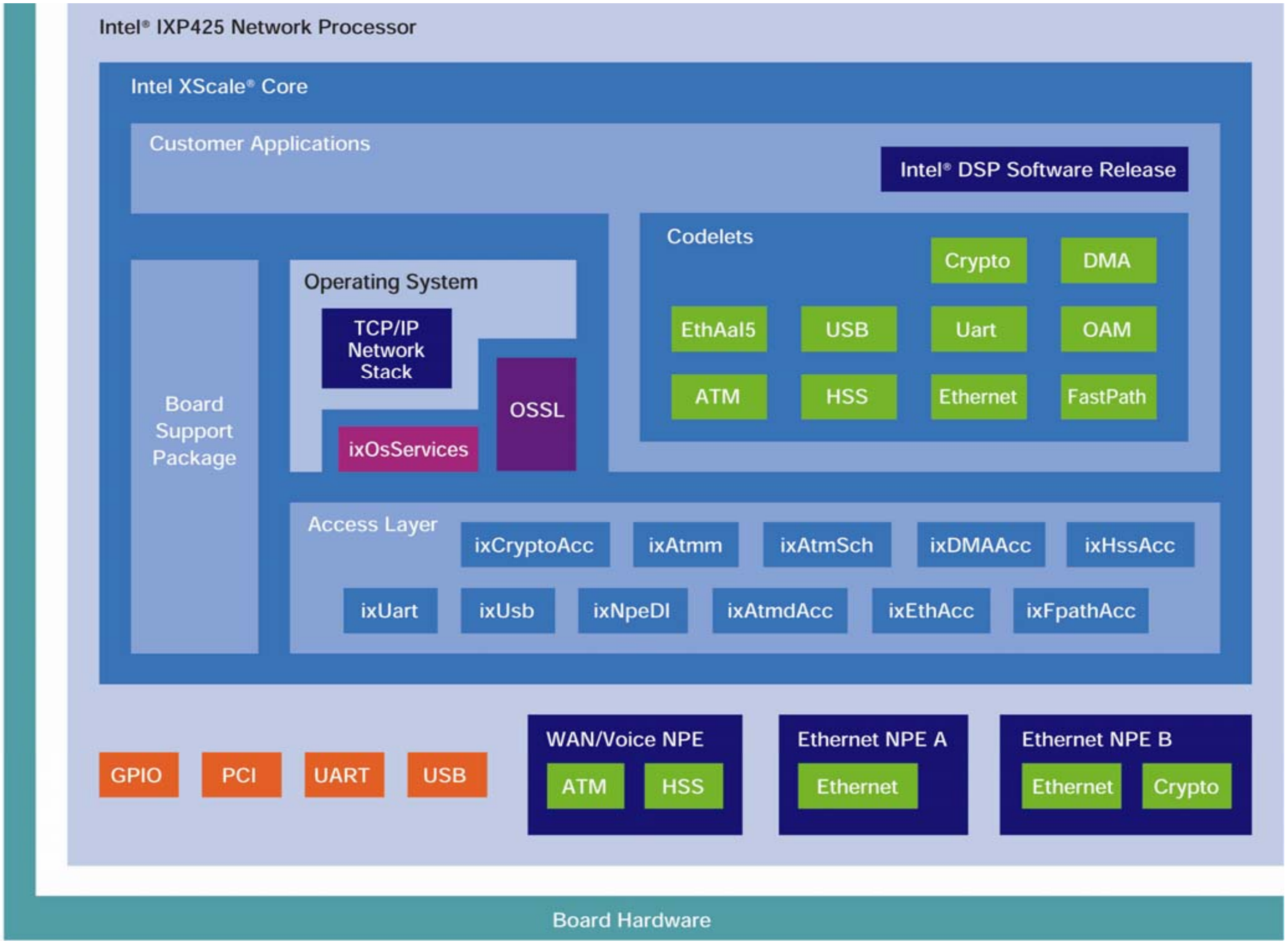
IXP2855


18-bit
18-bit
18-bit
Stripe/Byte Align
DRAM I/F 1
DRAM I/F 2
DRAM I/F 3
Crypto 1
Crypto 2
MEv2 1
MEv2 2
MEv2 3
MEv2 4
MEv2 8
MEv2 7
MEv2 6
MEv2 5
Rbuf
SPI-4.2 or CSIX I/F
Tbuf
16-bit
16-bit
64-bit
PCI (64-bit) 66 MHz
Intel XScale Core
Core I/F
MEv2 9
MEv2 10
MEv2 11
MEv2 12
MEv2 16
MEv2 15
MEv2 14
MEv2 13
Hash 64/28/128
Scratch Memory
CSRs
-Fast_wr
-UART
-Timers
-GPIO
-BootROM/Slow Port
QDR SRAM 1
QDR SRAM 2
QDR SRAM 3
QDR SRAM 4
E/D Q
E/D Q
E/D Q
E/D Q
18-bit
18-bit
18-bit
18-bit
18-bit
18-bit
18-bit
18-bit

IXP425 and XScale

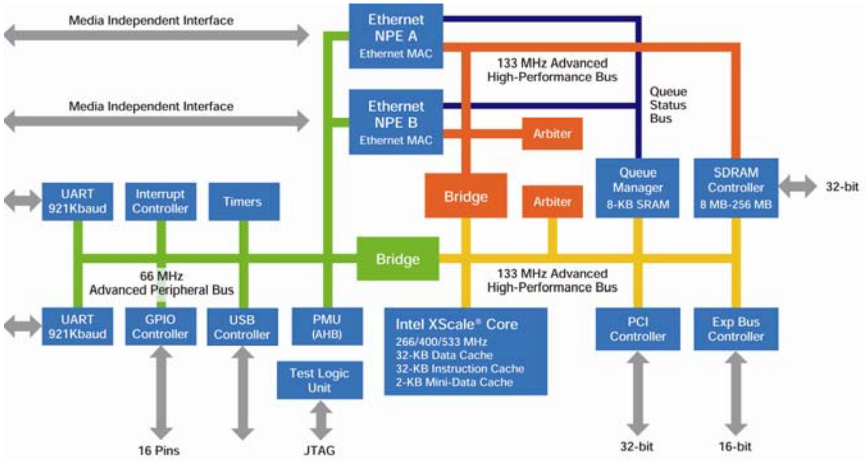


IXP425 software architecture

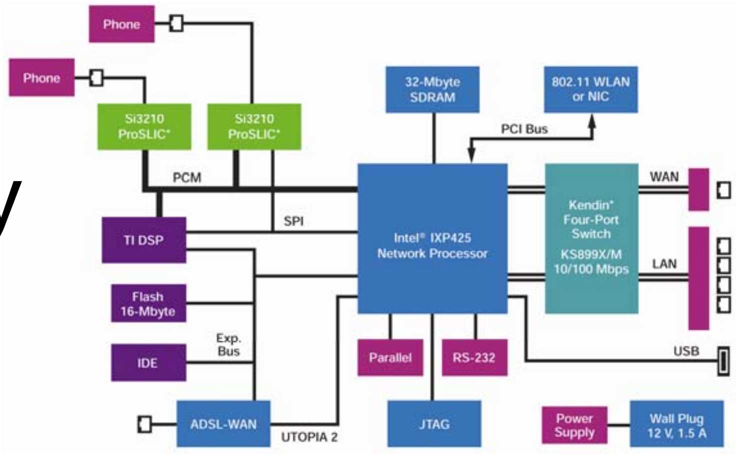


IXC1100 and Res. Gateway

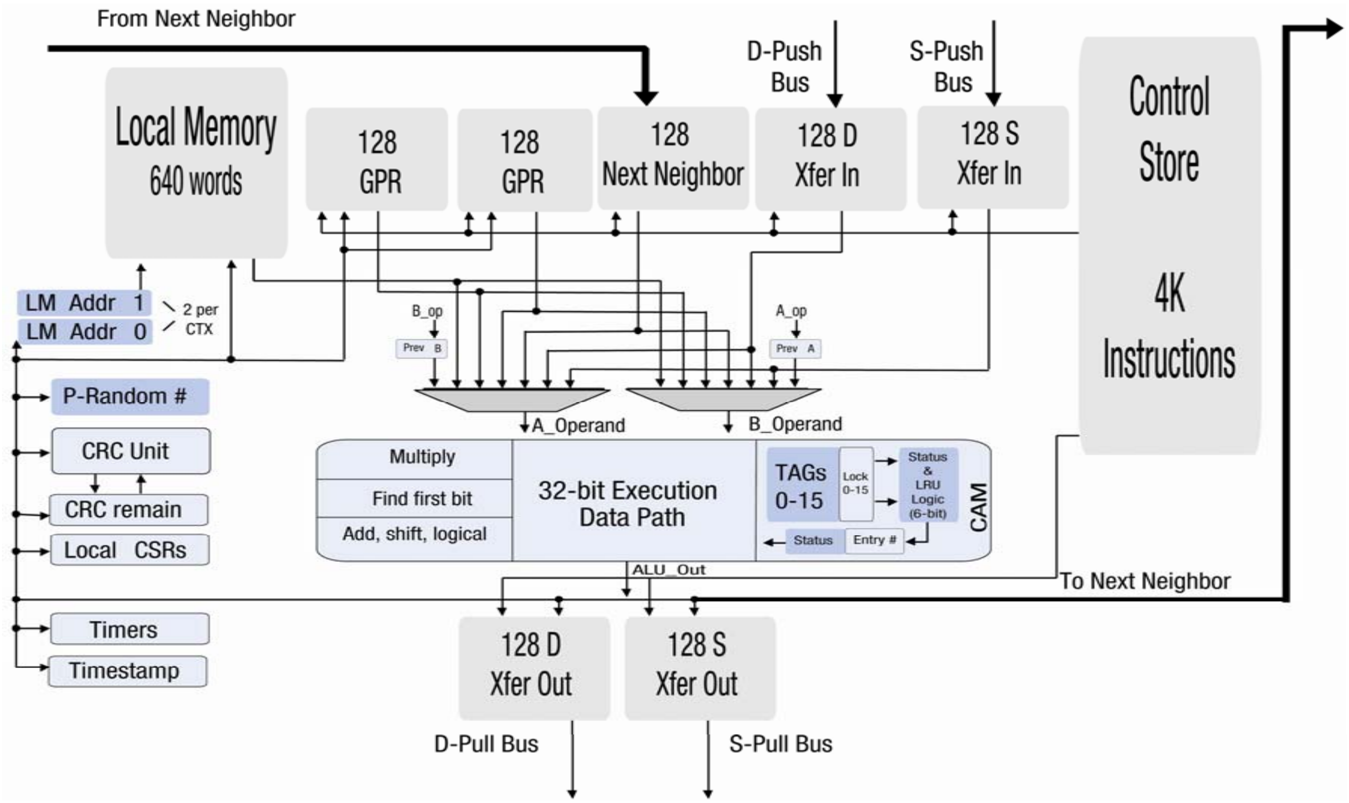
- IXC1100
- Control-plane proc.



- Residential gateway system architecture



IXP425 micro-engine


From Next Neighbor
D-Push Bus
S-Push Bus
Local Memory 640 words
128 GPR
128 GPR
128 Next Neighbor
128 D Xfer In
128 S Xfer In
Control Store
4K Instructions
LM Addr 1
LM Addr 0
2 per CTX
B_op
Prev B
A_op
Prev A
P-Random #
CRC Unit
CRC remain
Local CSRs
A_Operand
B_Operand
Multiply
Find first bit
Add, shift, logical
32-bit Execution Data Path
TAGs 0-15
Lock 0-15
Status & LRU Logic (6-bit)
Status
Entry #
CAM
ALU_Out
To Next Neighbor
Timers
Timestamp
128 D Xfer Out
128 S Xfer Out
D-Pull Bus
S-Pull Bus