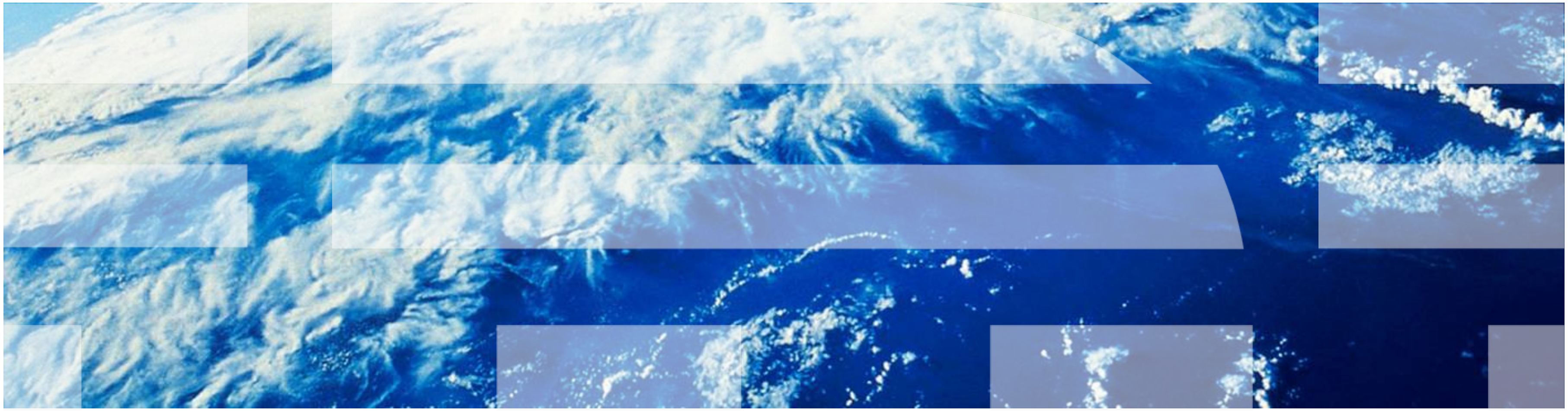
IEEE Hot Interconnects 20, Santa Clara, CA, Aug. 22-23, 2012

Rx Stack Accelerator for 10 GbE Integrated NIC

F. Abel, C. Hagleitner
IBM Research – Zurich
Switzerland

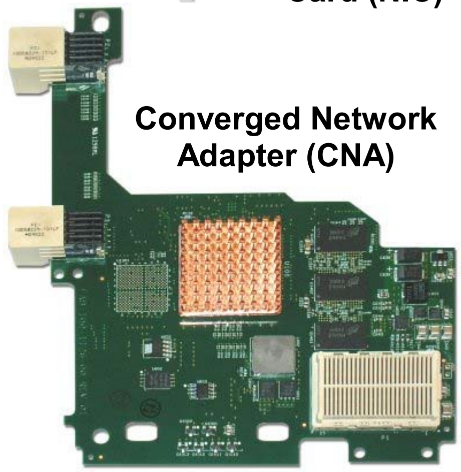
F. Verplanken
IBM Systems & Technology Group
La Gaude – France



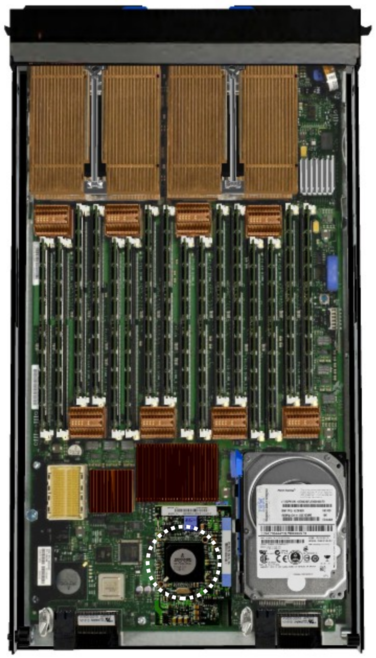
Discrete vs Integrated Network Interface Controller

- **Discrete NIC (dNIC)**
  - Peripheral ASIC device
  - Marketed as:
    - Ethernet Controller
    - Converged Controller

Network Interface Card (NIC)



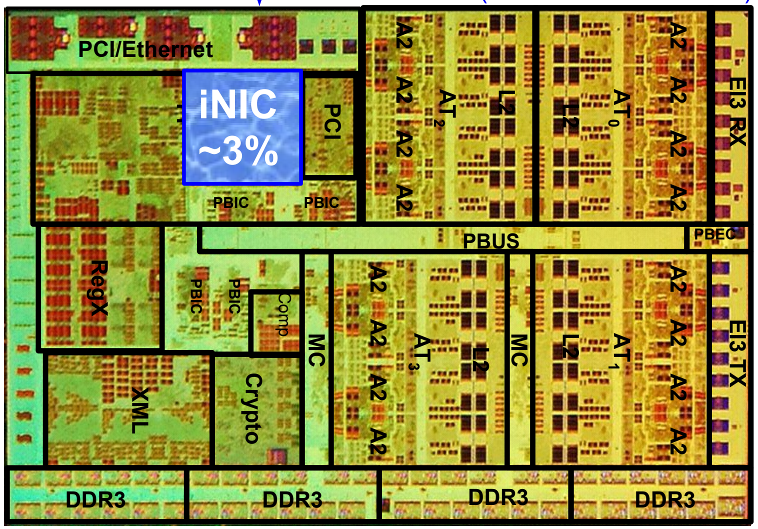
Converged Network Adapter (CNA)



LAN On Motherboard (LOM)

- **Integrated NIC (iNIC)**
  - Sun Niagara 2
  - Freescale → QorIQ™ family
  - IBM → **PowerEN™**

410 mm^2
(1.43 billion transistors)



☑ Higher performance, lower latency

☑ Lower power consumption

☑ **Significant cost reduction**

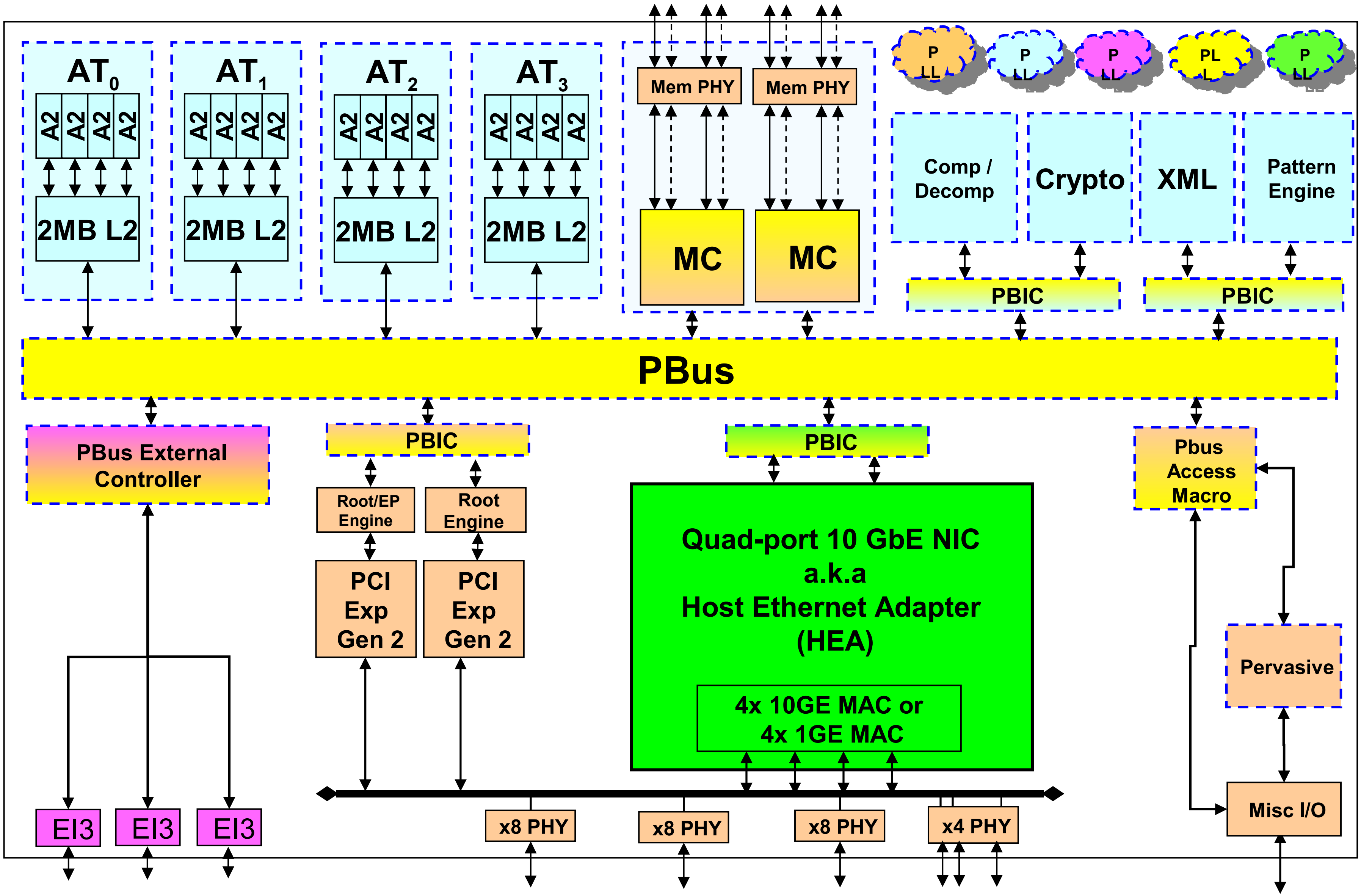
☒ Alter the general-purpose nature of the computer complex

Outline

- **Hardware context of this work**
  - PowerEN^{TM} / Host Ethernet Adapter
- **Functional requirements**
- **Architecture of the Rx Stack Accelerator**
  - Data path
  - Packet parser
  - Packet handler
- **Results**
  - Implementation
  - Performance
- **Summary and conclusions**

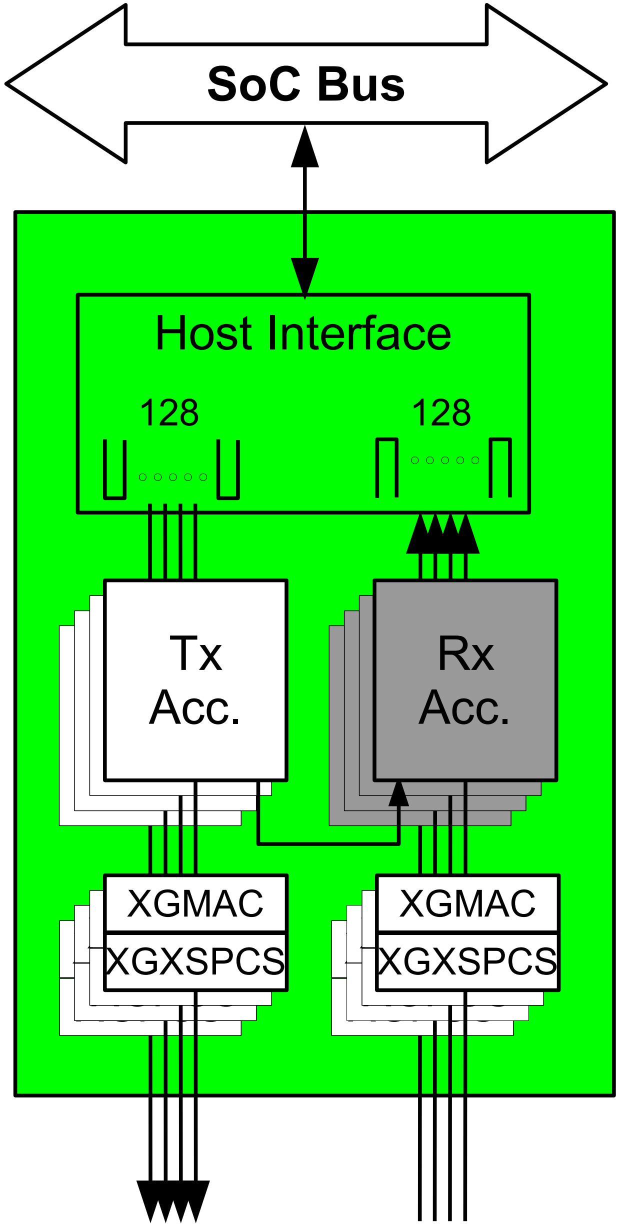
Hardware Context

Context: PowerEN™ / Host Ethernet Adapter (HEA)



Overview of the Host Ethernet Adapter (HEA)

- **4×10 GbE state-of-art Ethernet controller featuring:**
  - I/O virtualization support
    - Through 128 queue pairs
    - Internal layer-2 switch for partition-to-partition data traffic
  - Flexible queue selection and scheduling assist
  - Rx and Tx protocol acceleration
  - Low-latency through direct processor bus attachment and cache injection
  - Multi-core scaling,
  - Interrupt and receive coalescing assist,
  - 9 KB Jumbo frame support,
  - Memory address protection
  - . . .
- **Equivalent functionality and performance levels as modern discrete NICs**
- **Small footprint (13 mm^2) and low power (2.6 W)**

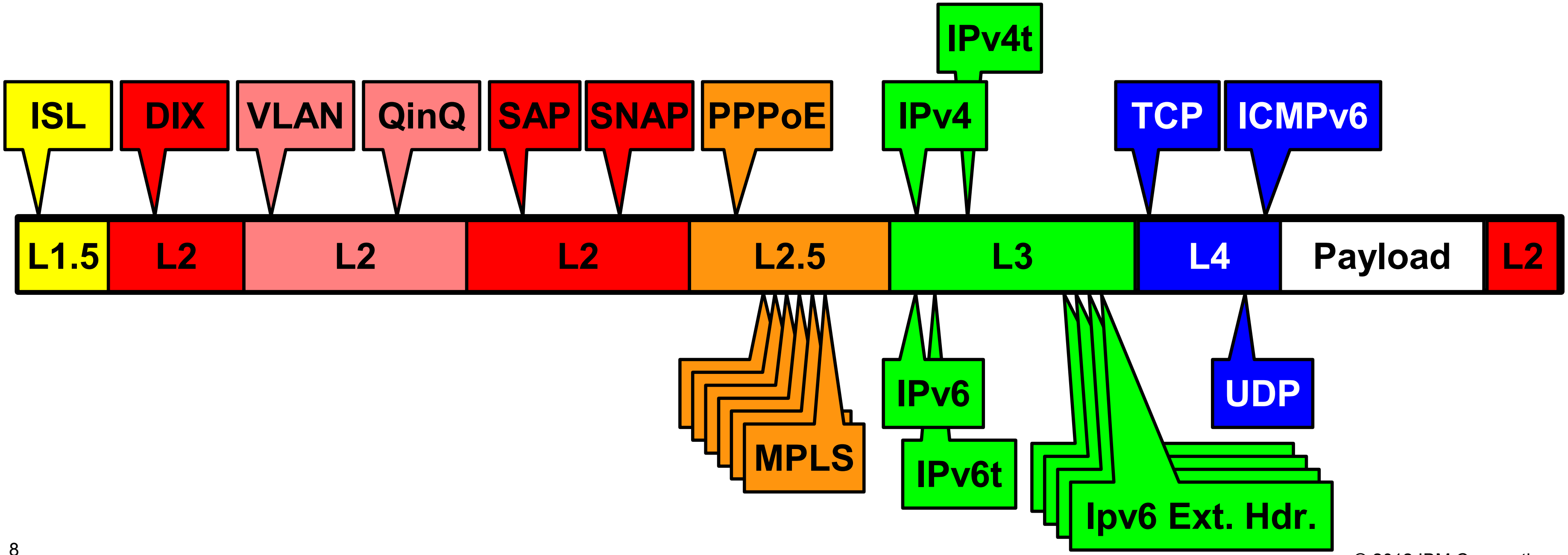


RXACC Functional Requirements

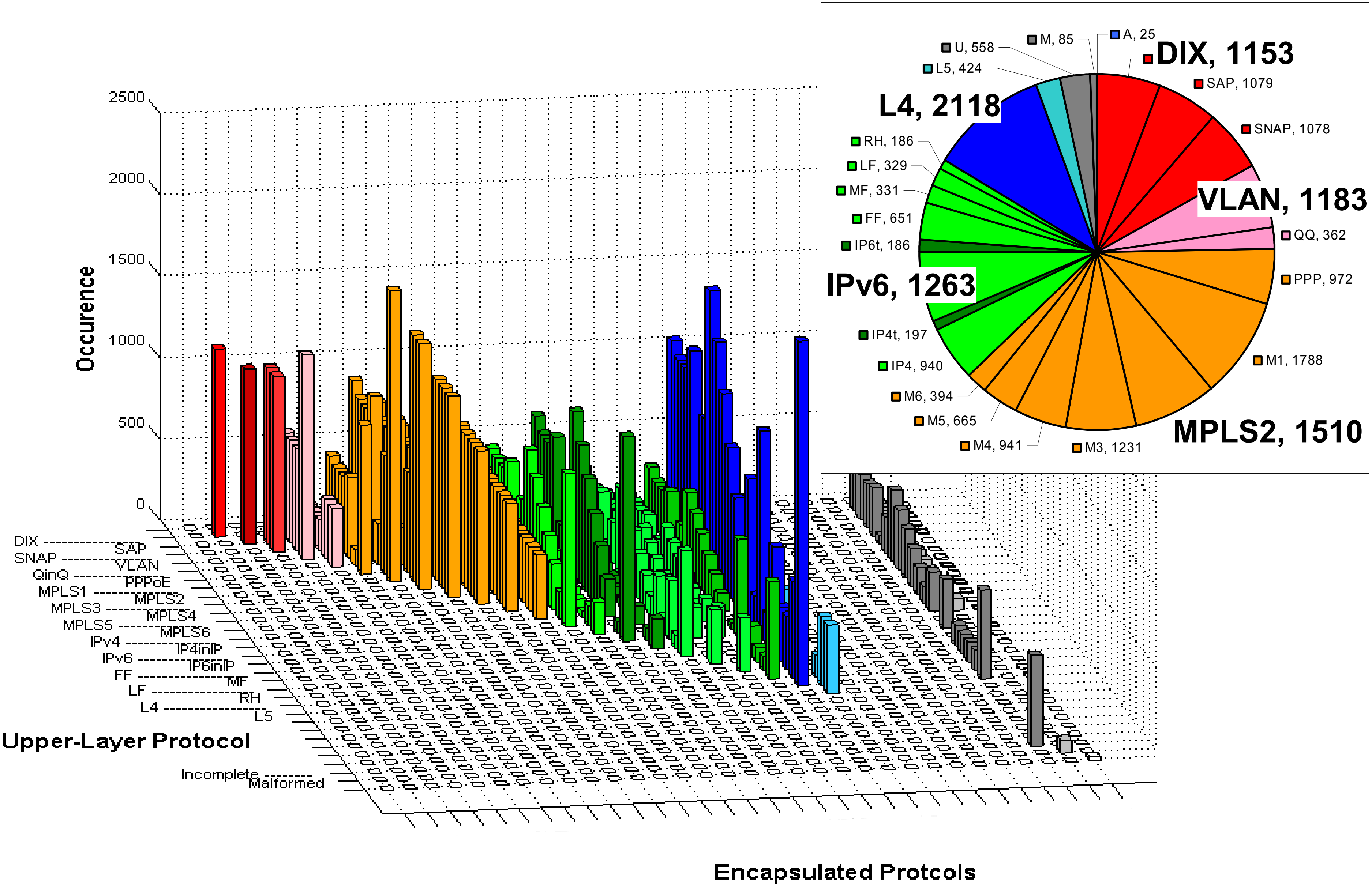
Target Domain Requirements

- **PowerEN^{TM} targets network-facing applications**
  - Must cope with a large spectrum of protocols (13+), traffic characteristics and policies
    - E.g. routers, firewalls, intrusion-prevention systems and network analytics
  - Must be able to adapt to changes
    - Handle other existing or emerging protocols
  - Virtualized I/Os → Must sustain 16 Gb/s (internal layer-2 switch)

- **Ethernet: A trucking service for application data**
  - Protocol layered architecture (i.e., encapsulation) → 2000+ permutations

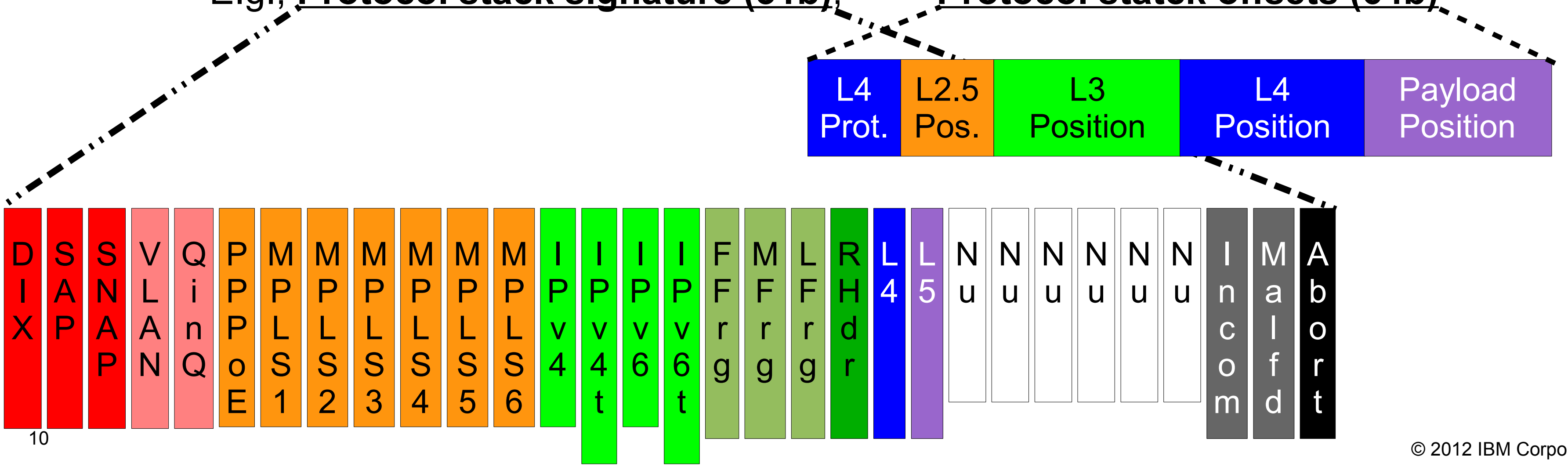


Distribution of the Protocol Stacks



Offloaded Service Requirements

- **WHY**
  - TCP Rx+Tx processing → ~3000 instructions (assuming zero-copy and checksum offload)
  - 10 GbE → Occurrence of 64 B packets → 67.2 ns (14.8 Mfps)
  - A generally accepted rule of thumb → 1 GHz / 1 Gb/s
- **WHAT (business as usual)**
  - *'per-byte'* operations
    - check-summing
  - *'per-packet'* operations
    - header processing: VLAN, MAC, flow identification, QoS determination, discard, errors, traffic steering
- **HOW (different)**
  - Flexible way → Programmable parsing and processing (rule-based)
  - Meta-data descriptors → Save 300-400 instructions per frame
    - E.g., **Protocol stack signature (31b)**, **Protocol statck offsets (64b)**



L4 Prot. | L2.5 Pos. | L3 Position | L4 Position | Payload Position

DIX | SAP | SNAP | VLAN | QinQ | PPPoE | MPLS1 | MPLS2 | MPLS3 | MPLS4 | MPLS5 | MPLS6 | IPv4 | IPv4t | IPv6 | IPv6t | FFrg | MFrg | LFrg | RHdr | L4 | L5 | Nu | Nu | Nu | Nu | Nu | Nu | Incom | Malfd | Abort

Rx Stack Accelerator

Rx Stack Processing

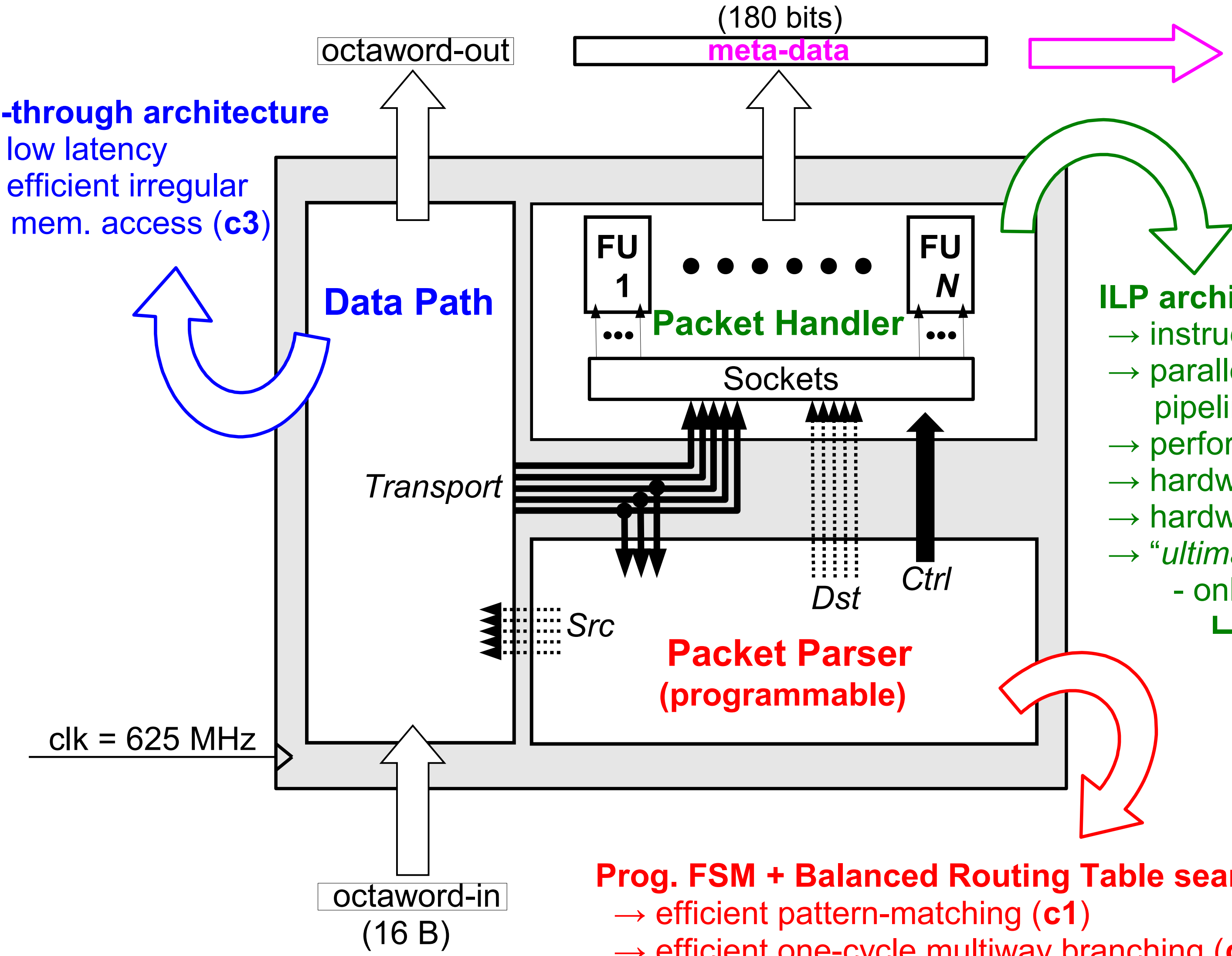
- **Can be decomposed in three major tasks:**
  - (**t1**) Parsing
    - Identify protocol stack + position of protocol fields
  - (**t2**) Data extraction
    - Locate and retrieve data to be processed
  - (**t3**) Processing
    - Execute instructions based on identified rules
      - Filtering (MAC, VLAN), VLAN extraction,
      - Rx queue assignment, checksum verification,
      - flow determination, discard, counters increments, …

- **Main characteristics exhibited by protocol processing applications [Jantsch1998]**
  - (**c1**) Intensive use of pattern matching
    - especially on headers
  - (**c2**) Complex and control dominated flow
    - many nested if-then-else and case structures
  - (**c3**) Intensive use of irregular memory accesses
    - various sizes and patterns

RXACC Architecture

A Transport Triggered Architecture (TTA) w/ 5 transport buses (1 byte/bus)

Cut-through architecture
→ low latency
→ efficient irregular mem. access (**c3**)



Frame "signature"
→ saves 300-400 instructions (avg.)

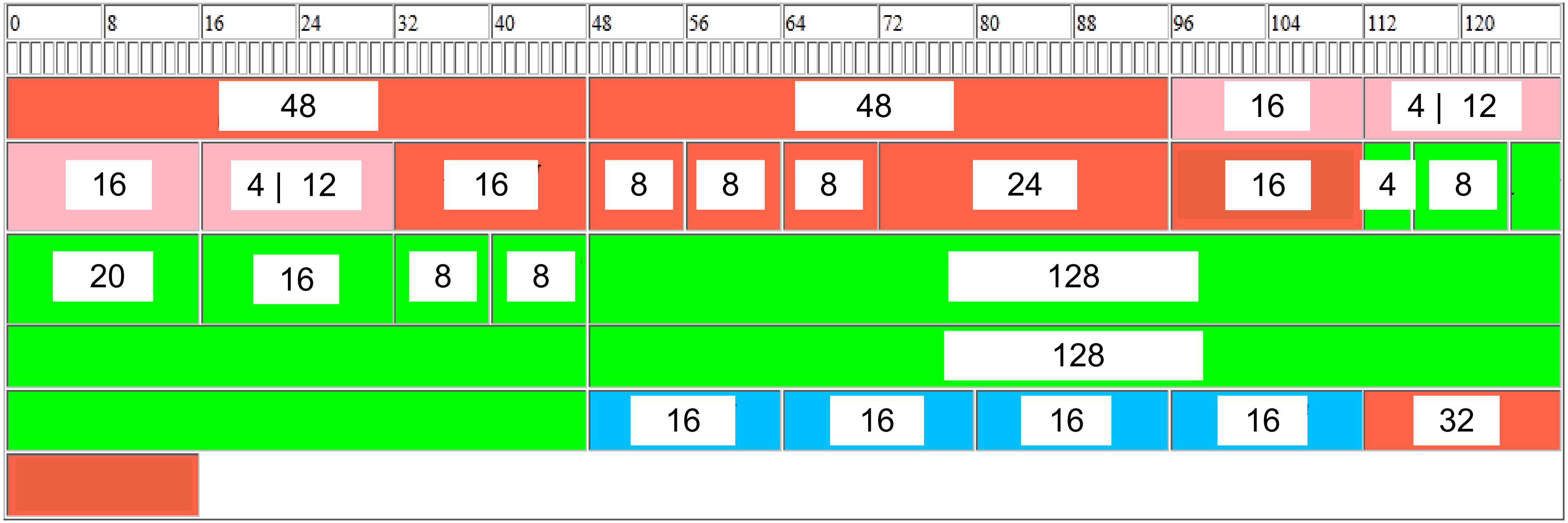
ILP architecture
→ instruction-level parallelism
→ parallel, independent, and pipelined coprocessors (FUs)
→ performance scalability
→ hardware efficiency
→ hardware modularity
→ "*ultimate*" RISC architecture
- only 1 instruction
↳ "*move data*"

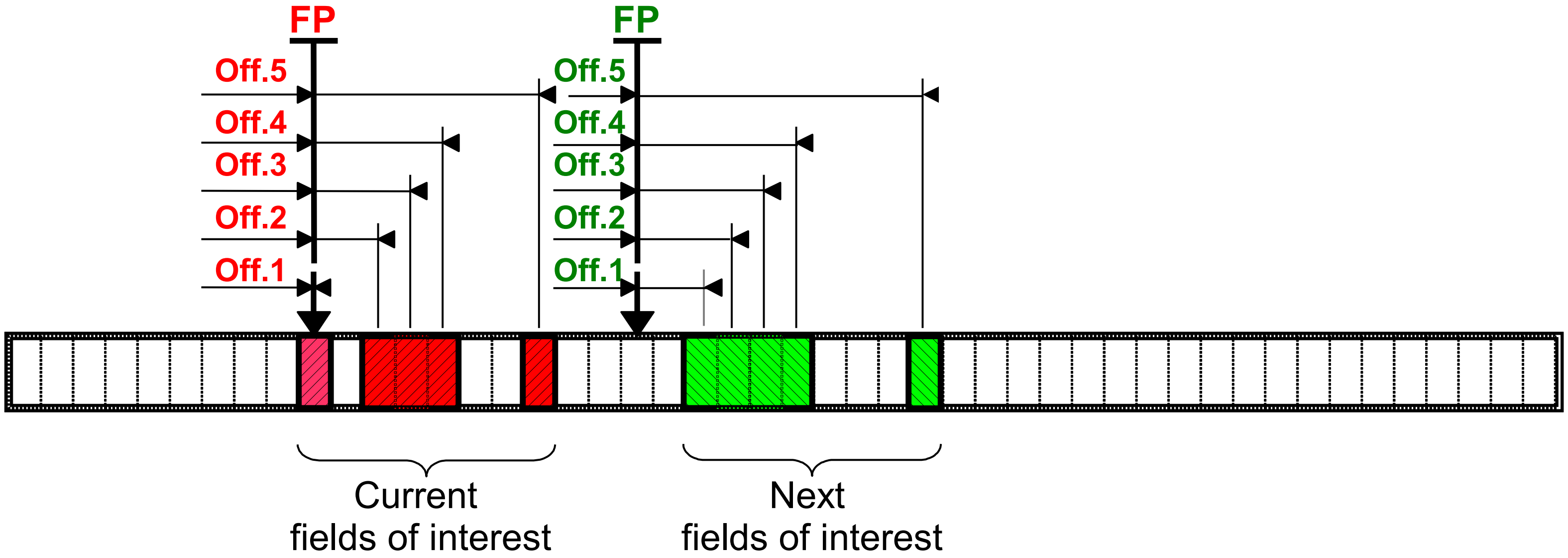
Prog. FSM + Balanced Routing Table search algorithm (B-FSM)
→ efficient pattern-matching (**c1**)
→ efficient one-cycle multiway branching (**c2**)

Data Path
(DP)

Multi-field Packet Inspection & Data Extraction

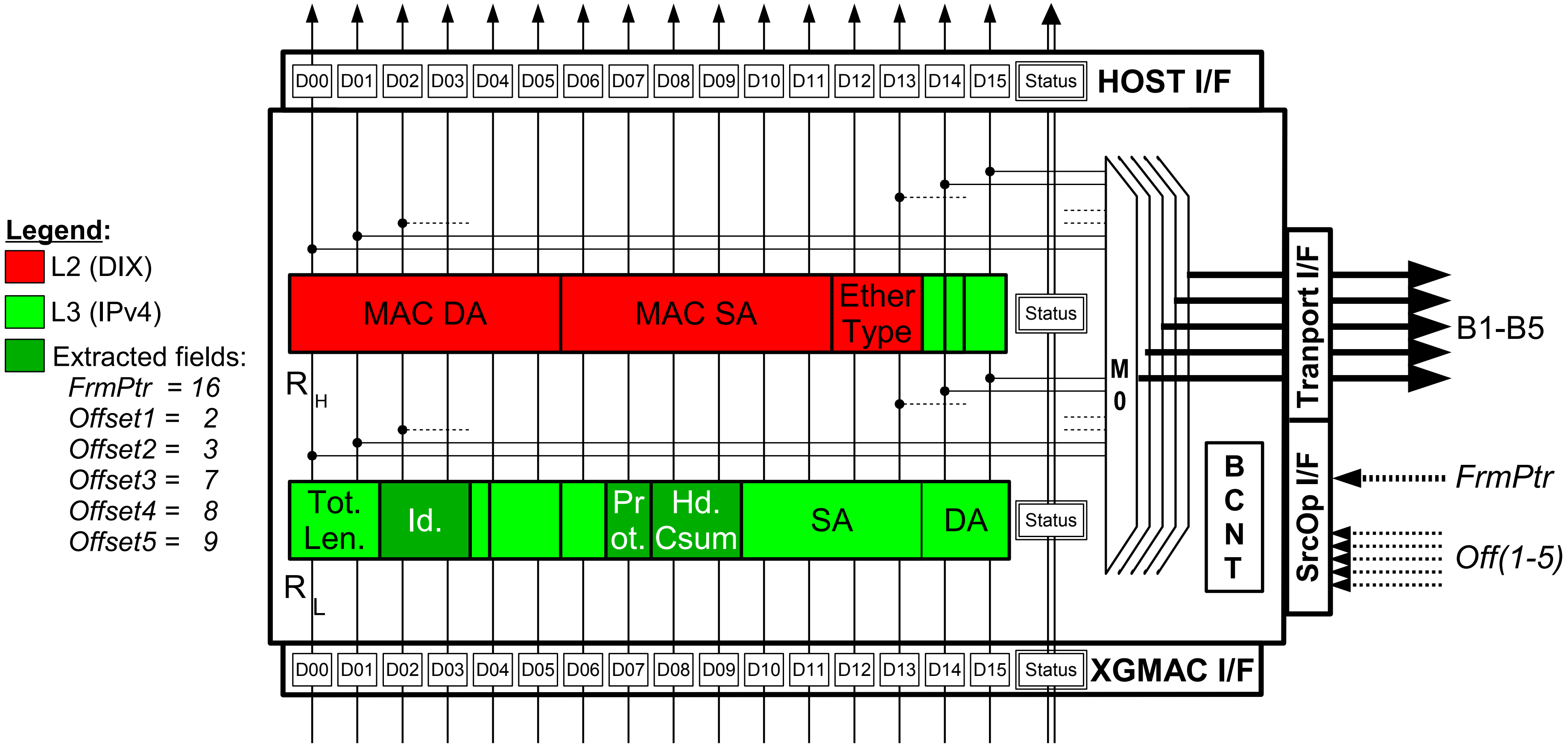
QQ/VLAN/SAP/SNAP/IPv6/UDP





Data Path Architecture

- **Cut-through architecture**
  - Relaxed buffering (2 × 16B) + Low latency
  - Flexible data extraction (any 5 bytes within the 32B window)
  - Entire packet is exposed to the parser → Can inspect and match any data of the frame

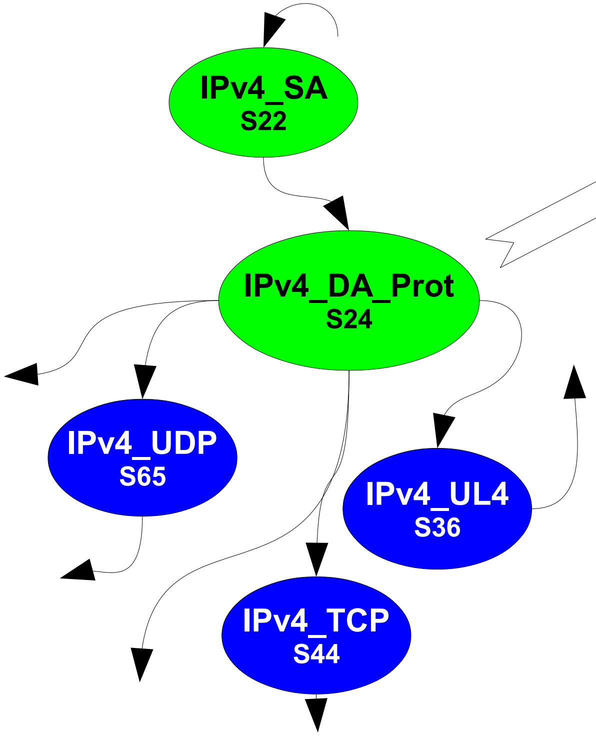


Packet Parser (PP)

Packet Parser

Design space:

- ☒ micro-coded → limited branch capabilities → multi-GHz operation → power
- ☒ finite state machine (FSM) → efficient but inflexible
- ☑ programmable finite state machine (pFSM) → high performance and energy efficient



State transition diagram

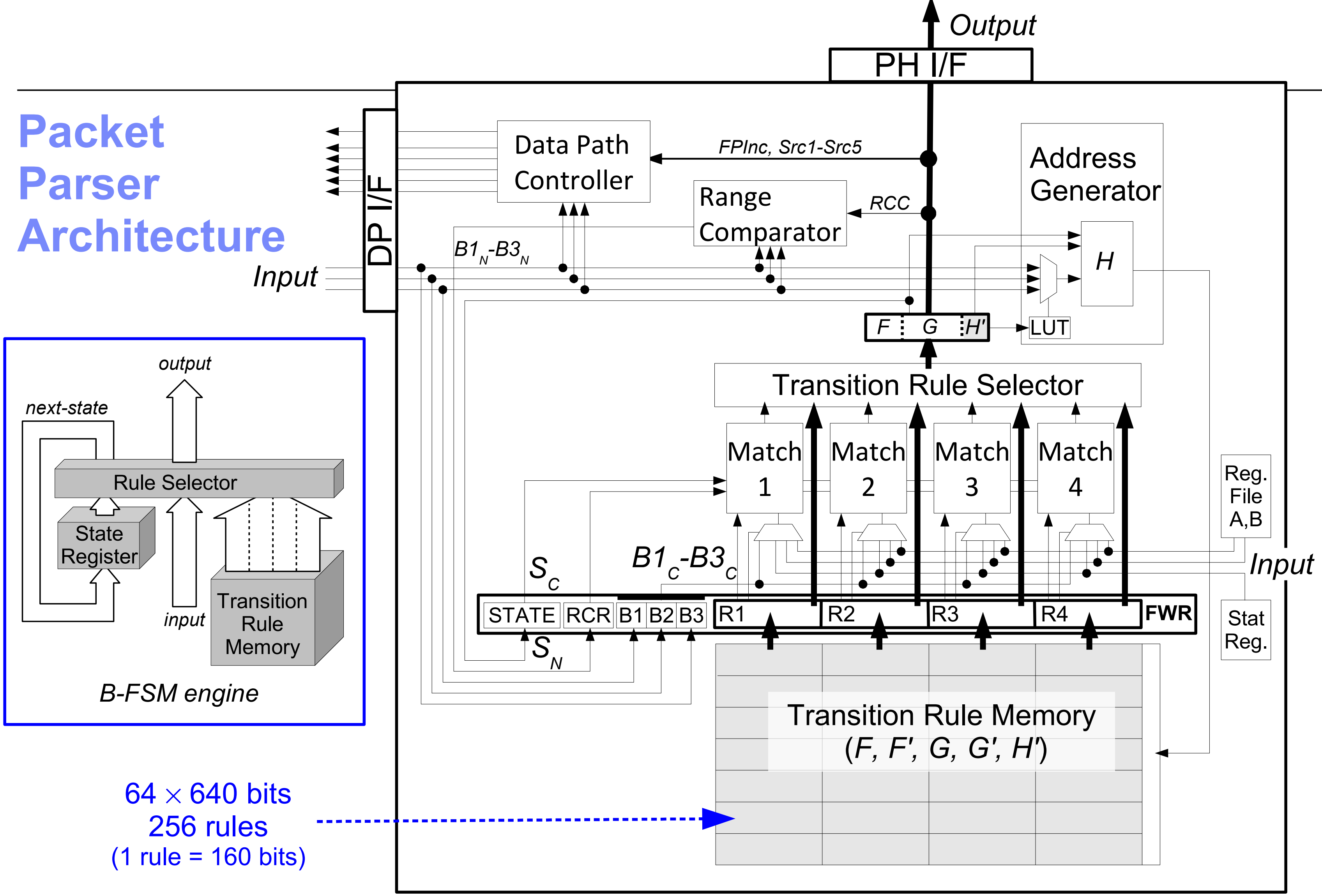
Rule	Current State	Input symbols			Next State	Priority
		Input 1	Input 2	Input 3		
. . .	…..	…	. . .	. . .	. . .	. . .
R44	S22	XXXX_XXXXb	XXXX_XXXXb	XXXX_XXXXb	S24	0
R52	S24	XXXX_0101b	0001_0001b	X00X_XXXXb	S65	0
R53	S24	XXXX_XXXXb	0001_0001b	X00X_XXXXb	S82	1
R61	S24	XXXX_XXXXb	0010_1001b	X00X_XXXXb	S36	2
R112	S24	XXXX_0101b	0000_0110b	X00X_XXXXb	S44	3

State transition rules
('X' symbol represents a “don't care” bit)

HW Engine

We use the B-FSM architecture [van Lunteren2006]
'B' stands for Balanced Routing Table search algorithm (BaRT)
Originally designed for longest matching prefix searches
(i.e. routing table lookups)

Packet Parser Architecture



64 × 640 bits
256 rules
(1 rule = 160 bits)

Rule Structure

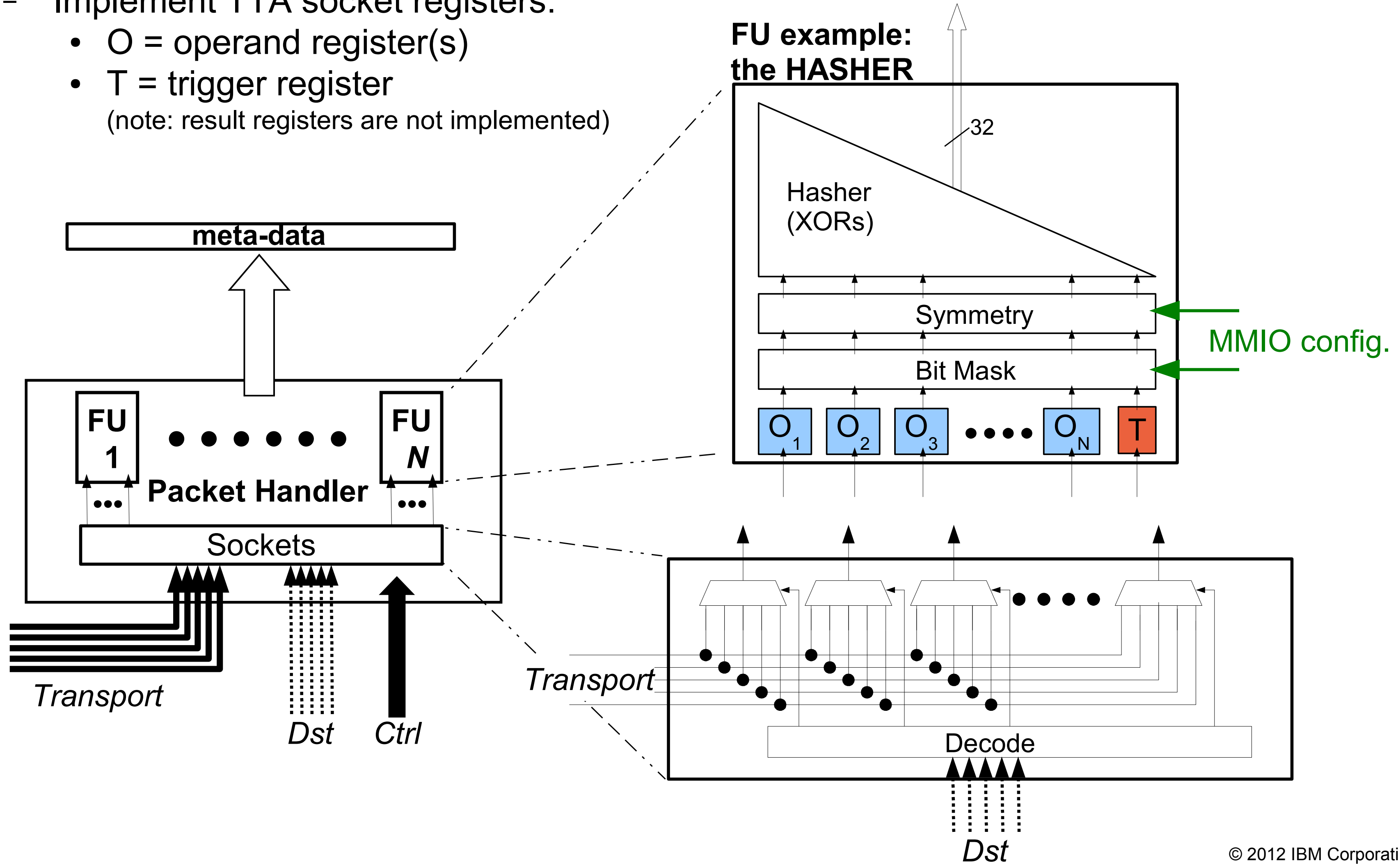
Test part		Result part		H part
S_C	Input symbols	S_N	Output symbols	H sym.

Packet Handler (PH)

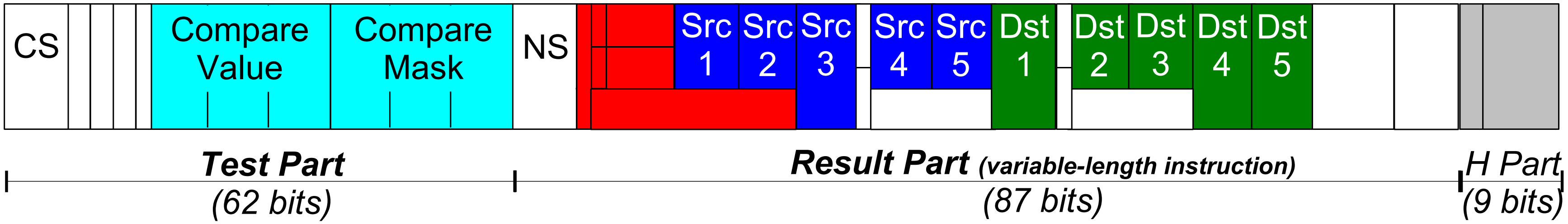
Packet Handler Architecture

- **Functional Units (FU)**
  - Independent coprocessors (IP-blocks) → Concurrent operation → ILP → Performance
  - Configurable through MMIO
  - Implement TTA socket registers:
    - O = operand register(s)
    - T = trigger register

    (note: result registers are not implemented)



Rule Format → 160 bits



🟥 Frame pointer increment (Long/Short – Fixed/Variable)

🟦 Source of the operands (i.e. offsets w.r.t the frame pointer)

🟩 Destination of the operands (i.e. functional unit registers)

- **Simplified Instruction Set**
  - Frame pointer instructions
    - e.g. ffpinc(*4*); // fixed increment
    - e.g. vfpinc(*1*); // variable increment
  - Move instruction
    - e.g.: move(*DP(4)*, *CSI.O1*); // unicast destination
    - e.g.: move(*DP(12)*, *HSH.O1 & CSI.O1 & CST.O9*); // multicast destination (socket write sharing)

Performance and Implementation Results

Implementation Results

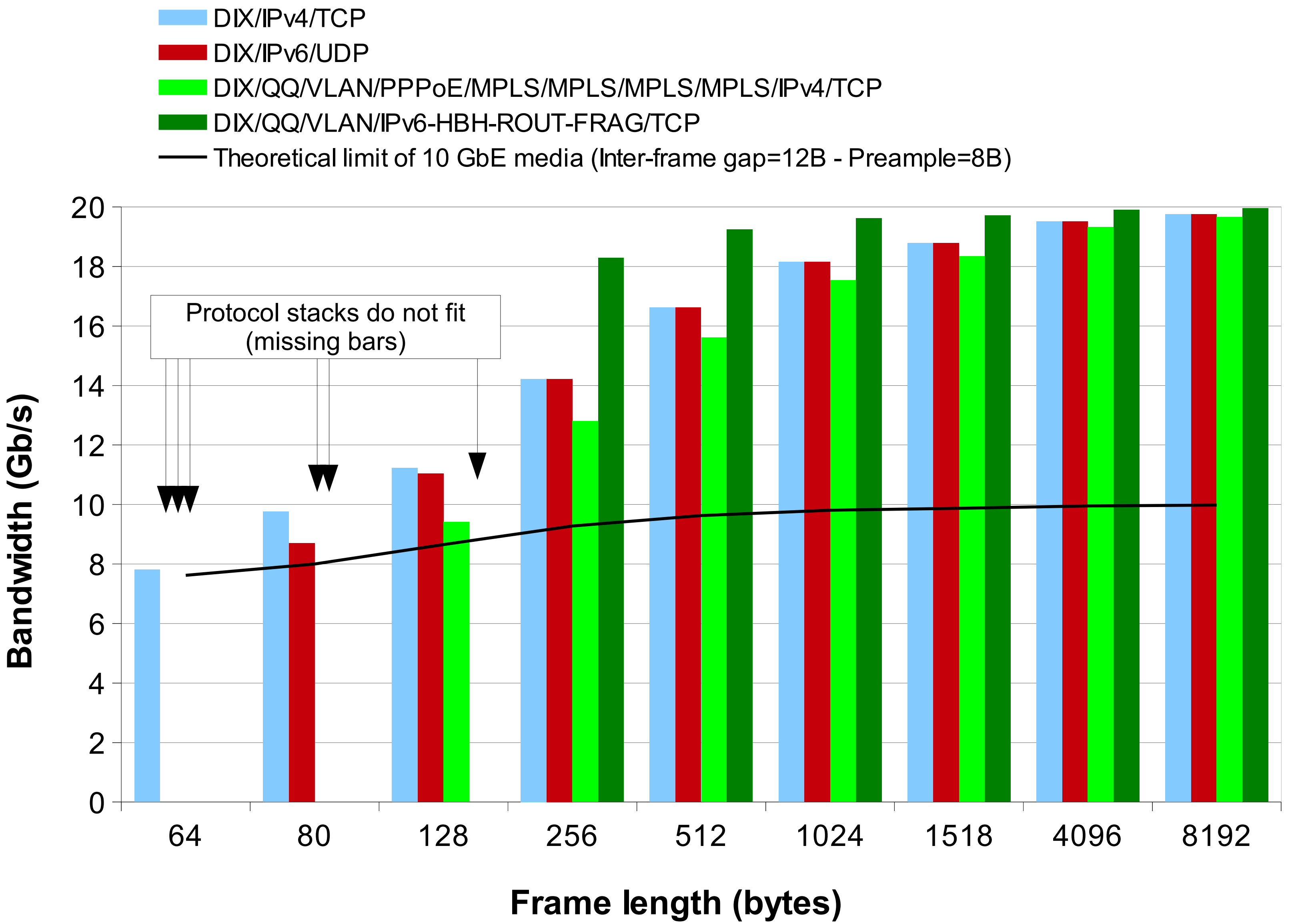
- **Area (in 45 nm SOI technology)**
  - 1 × RXACC = 0.7 mm^2
  - 4 × RXACC = 21.5 % of HEA (entire HEA = 13 mm^2)

- **Clock Frequency**
  - 625 MHz (27% of core frequency)

- **Power (estimate)**
  - RXACC ≈ 0.15 W (entire HEA = 2.6 W)

- **Transition Rule Memory Utilization**
  - 256 rules → 64 x (4 x 160) bits = 40 kbits
    - 68 states out of 128 (53 %)
    - 221 rules out of 256 (86 %)
  - Transition rule memory + ECC logic = 15 % of RXACC

RXACC Bandwidth



Summary & Conclusions

- **Processor compute complex + integrated network I/O complex**
  - **IFF** high-computation performance (1) + high-chip density (2) + low-power (3) + flexibility (4)
    - ➯ RXACC delivers (1) + (2) + (3) + (4)

- **(1) Performance**
  - 15 Mfps, 20 Gb/s (at "*relaxed*" clock frequency → 625 MHz)
  - Saves hundreds of CPU cycles per frame

- **(2-3) Area and power efficiency**
  - 0.7 mm^2 (45 nm SOI), 0.15 W

- **(4) Flexibility**
  - 2000+ protocol permutations
  - Programmable parsing and processing → Rule-based
    - Can parse new emerging standards (e.g. SDN)
    - Can inspect header and payload
    - Pragmatic approach w/ one code-set per application

- **RXACC = TTA + pFSM = novel Application Specific Processor**
  - ➯ Key enabler for integrated NICs
  - ➯ The architecture has headroom to scale towards 40-100 GbE

Rx Stack Accelerator for 10 GbE Integrated NIC

Thank you

IEEE Hot Interconnects 20, Santa Clara, CA, Aug. 22-23, 2012