One Billion Packet per Second Frame Processing Pipeline

Mike Davies
Director of IC Development
Fulcrum Microsystems

Fulcrum Ethernet Switch Chip Evolution

Tahoe

- 96 serdes
- Up to 24 10G ports
- 200nS latency
- 1MB shared memory
- L2 forwarding

TSMC 130nm

Bali

- 96 serdes
- Up to 24 10G ports
- 300nS latency
- 2MB shared memory
- L3 routing and ACLs at 360 Mpps
- DCB features

TSMC 130nm

Alta

- 96 serdes
- Up to 72 10G ports, 40G
- 400nS latency
- 8MB shared memory
- L3 tunneling and ACLs at 1080 Mpps
- DCB features
- Trill and virtualization

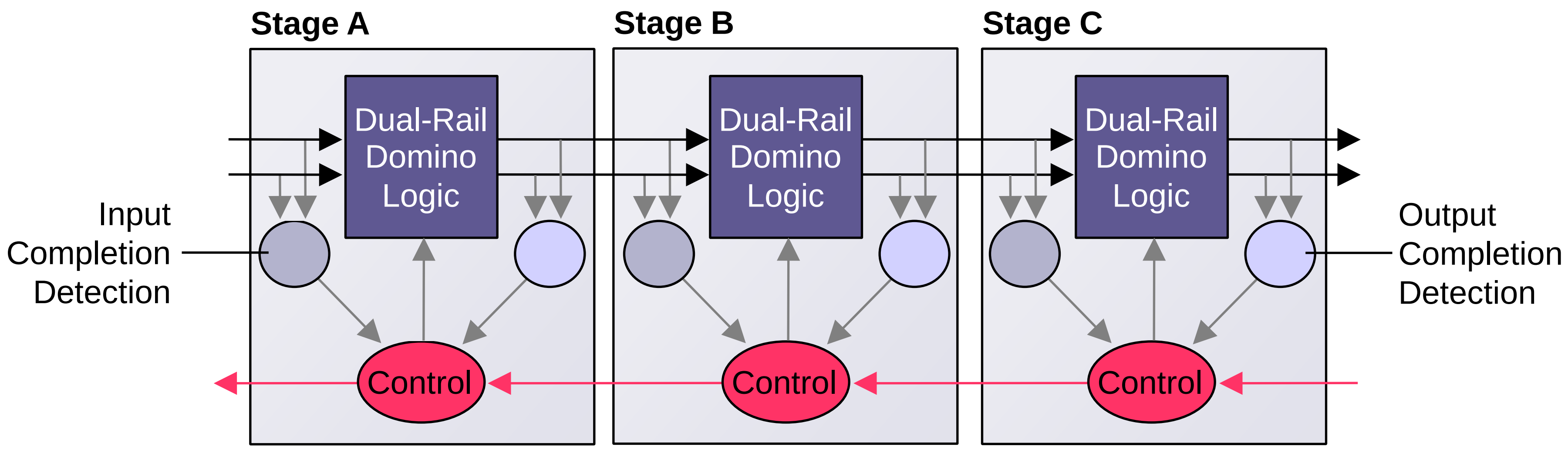
TSMC 65nm GP

Fulcrum's Secret Sauce

- **Quasi Delay Insensitive (QDI) Asynchronous Design**
  - Pioneered at Caltech during 1990s
  - Commercialized and further developed by Fulcrum
  - 30+ Patents on technology, methodology, circuits
  - 100 man-years invested in flow development to date
  - Employed in every Fulcrum chip product to date
- **Advantages**
  - High throughput (1.1-1.25 GHz in TSMC 65nm GP)
  - Low latency (domino logic, 100ps per pipeline stage)
  - Power efficient
  - Robust to manufacturing and operational variability
  - Very well suited for crossbar, TCAM, SRAM circuits
- **Disadvantages**
  - Not area efficient for frequencies lower than 1 GHz
  - Custom flow, unproductive for random logic functions

FULCRUM® microsystems

Self-Timed Domino Logic Pipelining



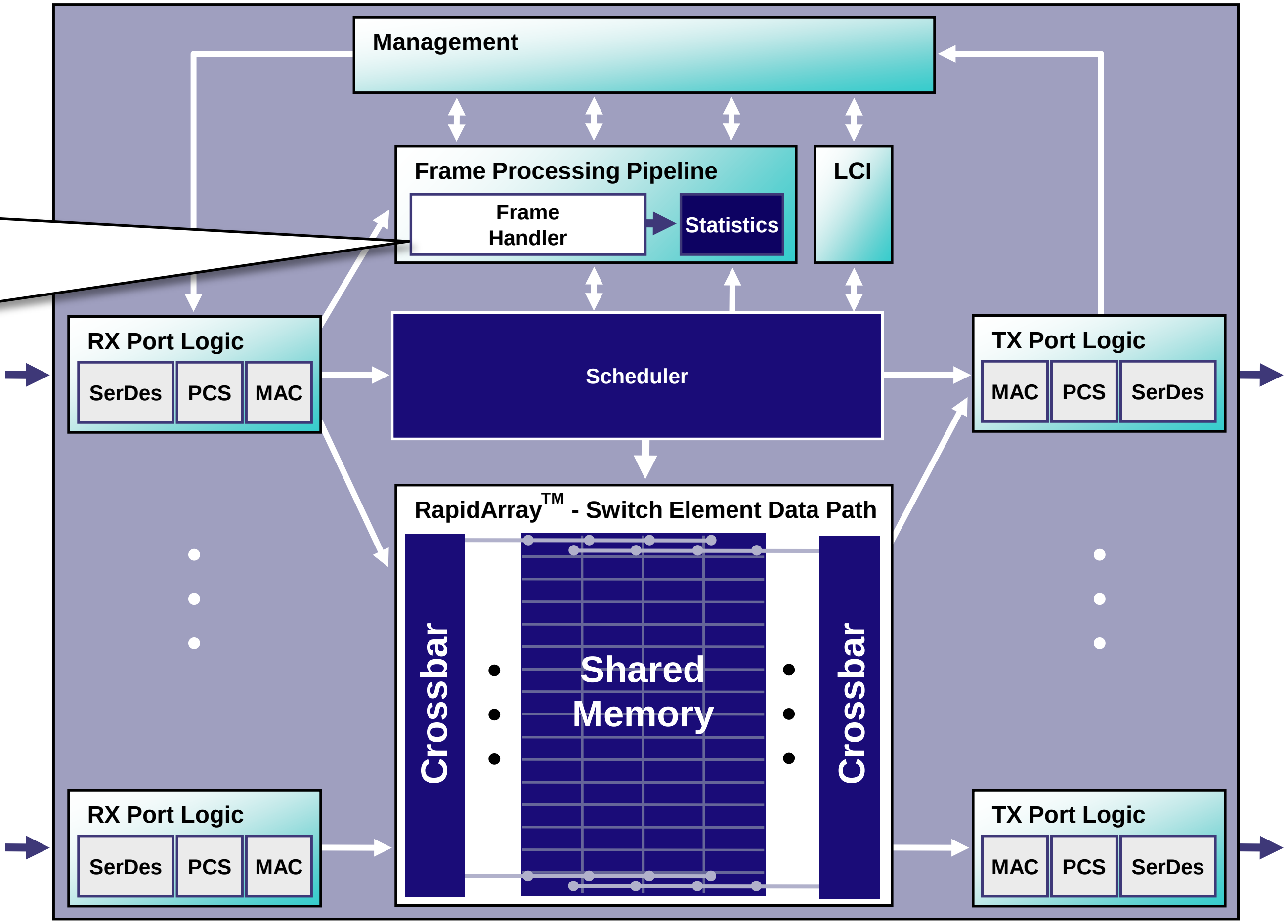
- **Data Completion Detection provides immunity to delay variation**
- **>2x Latency advantage over static logic, flop-based pipelines**
- **Addresses power inefficiency with clockless handshakes**
  - Circuit activity driven exclusively by useful data processing
  - Glitch-free
- **Leverages more efficient NFET transistors**
  - Better for performance , area, power

FULCRUM microsystems

Previous Generation Architecture

Synchronous Frame Processing Pipeline

~10M gates, 80% of Bali development effort



■	Async Logic
□	Sync Logic

Chip-level architecture remains the same in Alta

Frame Processing Wish List

Tahoe (Gen 1) Requirements

- Layer 2 Switching
- RMON/SMON

Bali (Gen 2) Requirements

- IPv4, IPv6 Routing
- IP Multicast
- 5-tuple ACLs
- 802.1Qbb Priority Flow Control
- 802.1Qau Congestion Notification, VCN
- ISL Tagging

Alta (Gen 3) Requirements

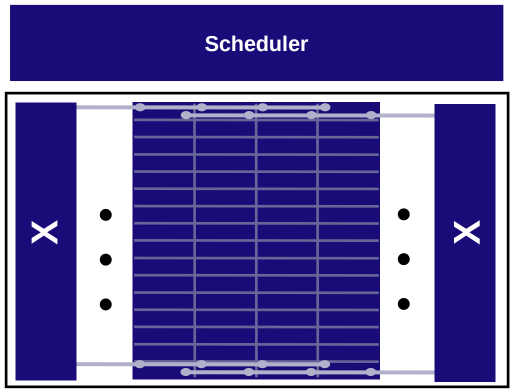
- Multi-Stage ACLs
- Trill/R-Bridge
- MPLS
- Q-in-Q
- SPB, PBB, PBT (MAC-in-MAC)
- 802.1ad Provider Bridging
- Data Center Bridging
- Preamble Tagging
- sFlow
- Advanced Frame Hashing for Load Balancing
- Pseudo-Random Load Balancing

Alta Implementation Challenge

Architectural Innovations Required

- Requirements demanded 3x performance increase
- Process provided 50% frequency increase (750 MHz to 1.1 GHz)

720G Switch/Scheduler (RapidArray)



- 67ns scheduling period over 1000 queues
- Up to 150 MHz per-queue event rate (40G)
- Challenging loop latency requirements
- Heavy use of dual-ported memories
- Very challenging asynchronous design problem, but effectively evolutionary from previous generation

Frame Processing Pipeline

- Initial plan: Dual synchronous pipelines at 550 MHz with RTL reuse
- Final solution required technology & architectural innovation to manage performance and functional complexity:
  1) *Asynchronous Place-and-Route*
  2) *Functional programmability at a fundamental level*

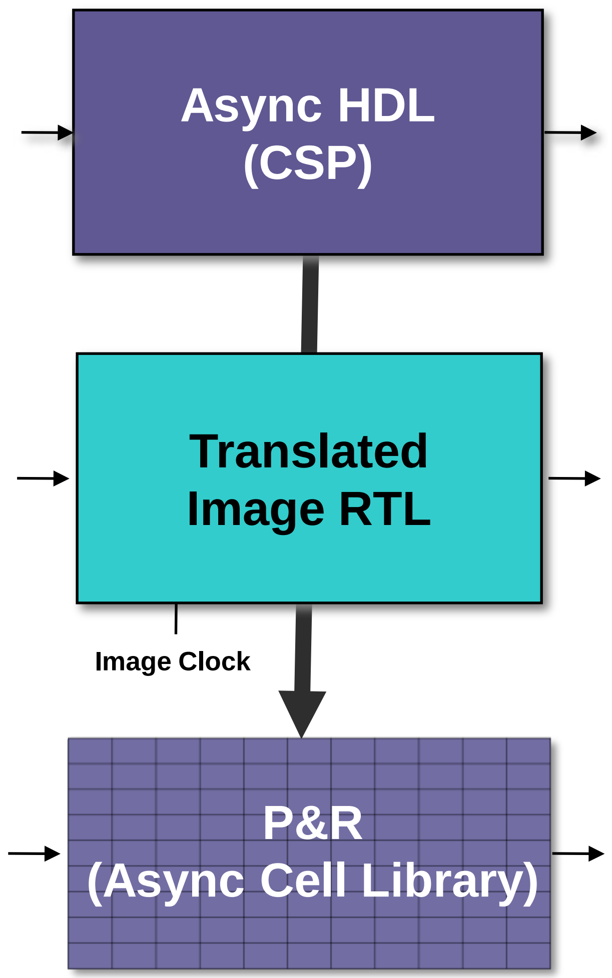
Innovation 1: Async Place-and-Route

“Proteus” flow developed for the job

- 1 GHz Asynchronous Synthesis/P&R flow
- Leverages RTL Compiler for logic synthesis
- Leverages SOC Encounter for P&R

Challenges:

- Implicit pipelining in source async HDL
- Unusual cell library:
  Dual-rail domino logic cells (30%)
  Handshake Control cells (70%)
- Cyclic timing constraints
- Maintaining “slack-matching” during buffer insertion



Alta is the first chip to utilize gigahertz asynchronous place-and-route circuitry

FULCRUM microsystems

Innovation 2: CAM/RAM/MUX Architecture

Observations:

- Common theme in switch silicon's frame processing computations:
  - Pattern matching
  - Simple guarded assignments
  - Muxing
  - Mapping tables
- Bounded iterative data dependencies from ingress to egress
- Bounded aggregate information transfer from ingress to egress

Suggested a streamlined frame processor architecture

- Bottom-up simplification:
  Implement the abstract computations, not hard-coded details
- TCAMs: Excellent for implementing pattern matching & DNF logic
- Crossbars: Excellent for assigning operands and muxing
- SRAMs: Needed for mapping tables and TCAM command mapping

Circuit primitives are all strengths of the async design style

FULCRUM microsystems

TCAM/RAM/MUX Decomposition (1)

```
IF (ValidRoute==1 ∧ Mcast==0 ∧ (EtherType==IPv4 ∨ EtherType==IPv6))
  Look up & Reassign DMAC based on matching IP route
ELSE
  Leave DMAC as-is
```

Sample hard-coded processing rules

```
IF (X1[0]==1 ∧ X1[4]==0 ∧ X3[15:0]==IPv4)
  cmd = $C_1$
ELSE IF (X1[0]==1 ∧ X1[4]==0 ∧ X3[15:0]==IPv6)
  cmd = $C_1$
ELSE
  cmd = $C_m$

Y = Action(cmd)
```

Reduction to generic header variables assigned at an earlier stage.

DNF guard encoding

```
IF ((X1 & 0x11)==1 ∧ (X3 & 0xFF)==0x8000)
  case = 1
ELSE IF ((X1 & 0x11)==1 ∧ (X3 & 0xFF)==0x86DD)
  case = 2
ELSE
  case = N

Cmd = ActionTable[case]
Y   = Action(cmd)
```

Suitable for Ternary CAM implementation

Strict priority encoding case per disjunctive term.

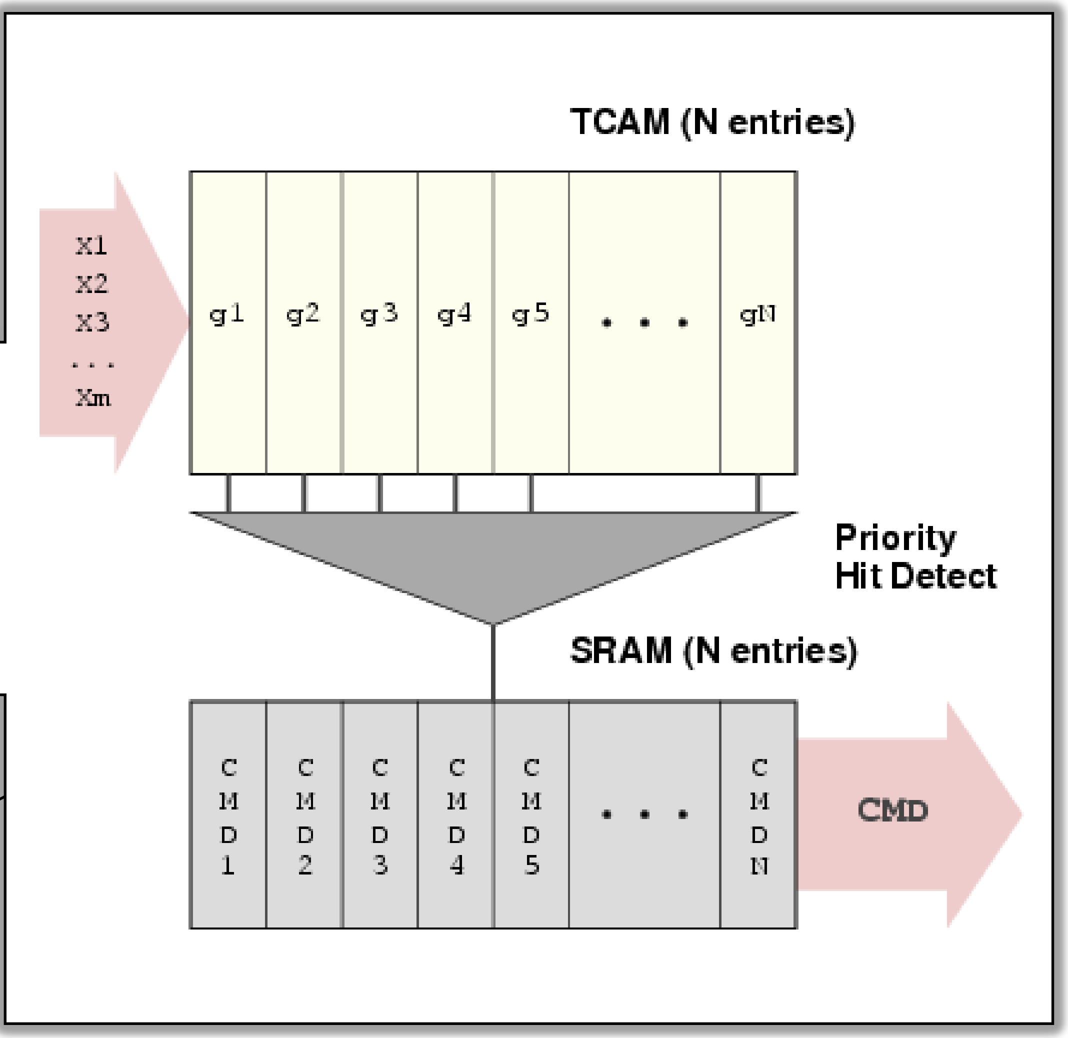
Command mapped per case.

TCAM/RAM/MUX Decomposition (2)

Hardware realization

```
IF ($g_1$($X_1$,$X_2$,…))
  case = 1
ELSE IF ($g_2$($X_1$,$X_2$,…))
  case = 2
…
ELSE
  case = N

cmd := ActionTable[case]
```

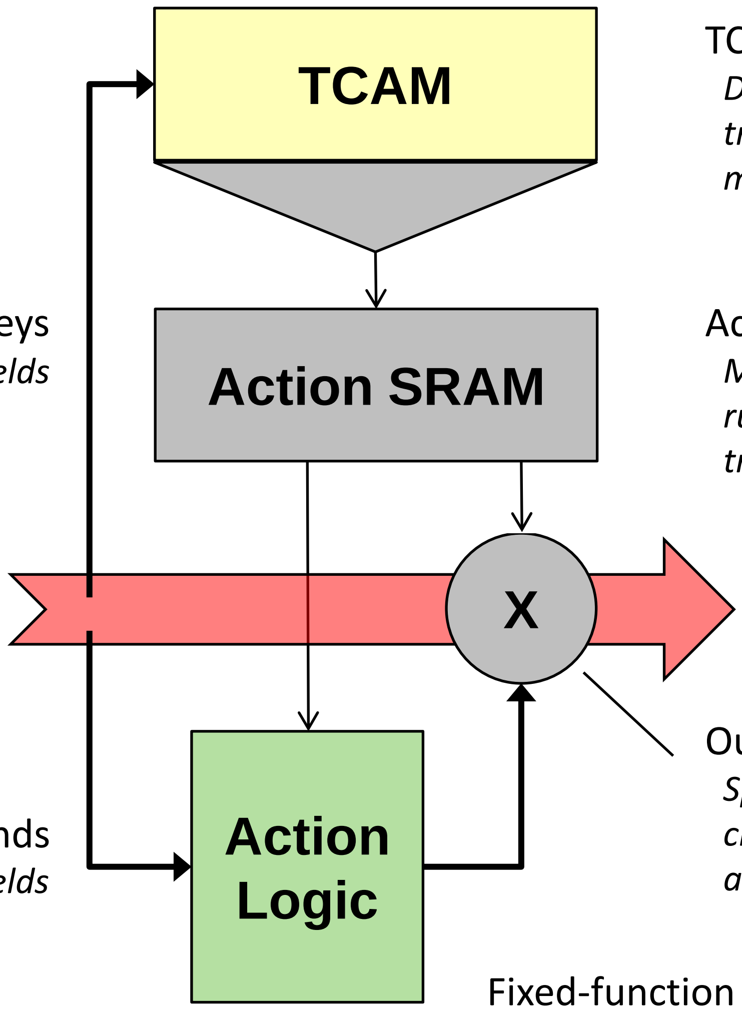


Standard TCAM/RAM structure

RAM maps a transformation command, i.e. a set of "actions", to apply to the frame's header fields at that point in the pipeline

FULCRUM microsystems

Fully General TCAM/RAM/MUX Stage



TCAM Keys
Subset of header fields

Header Field Bus
95 to 125 bytes

Action Operands
Subset of header fields

TCAM
Determines the transfomation rule that matches the frame

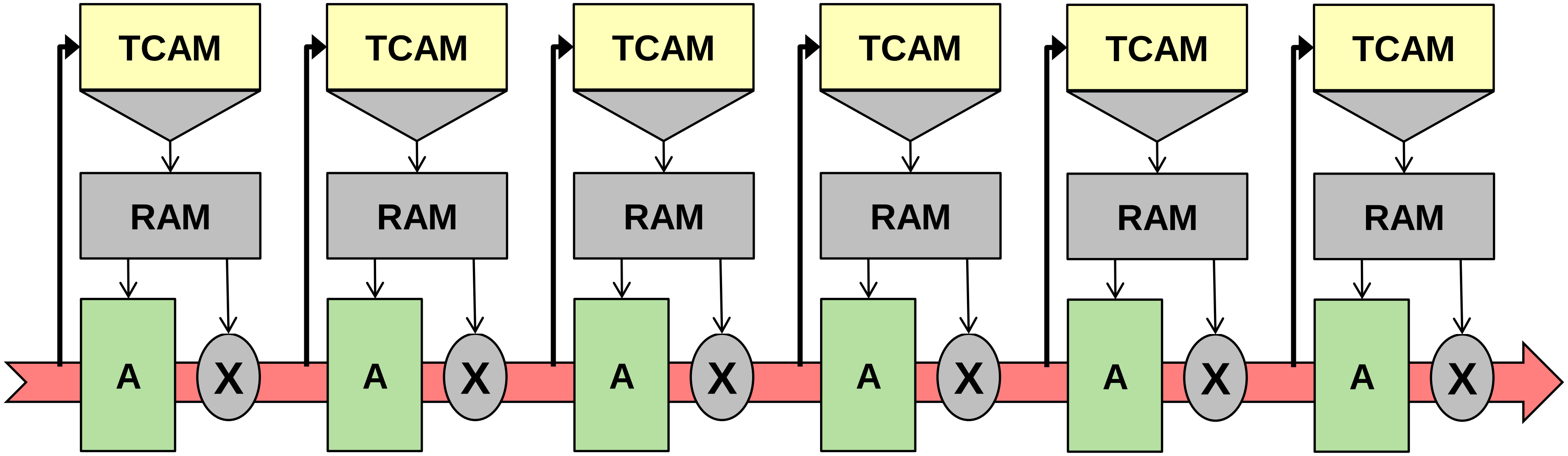
Action RAM
Maps winning TCAM rule number to transformation control

Output Muxing
Sparsely connected crossbar with masked assignment

Fixed-function Action Logic
e.g. Table Lookup, Hash Function – dependent on the type of stage

FULCRUM microsystems

Pipelined Loop Unrolling



Repeating stages provides iterative, fully-pipelined header transformations

- Scope of configurable operation increases exponentially with each stage
- TCAM keys, TCAM/RAM sizes, and Action functions may differ per stage in the pipeline

```
FOR i=1…N {
  IF ($g_{i,1}(X_1,X_2,…)$)
    X = $f_{i,1}(X_1,X_2,…)$
  ELSE IF ($g_{i,2}(X_1,X_2,…)$)
    X = $f_{i,2}(X_1,X_2,…)$
  …
}
```

Example: Configurable Parsing

Next header word from frame

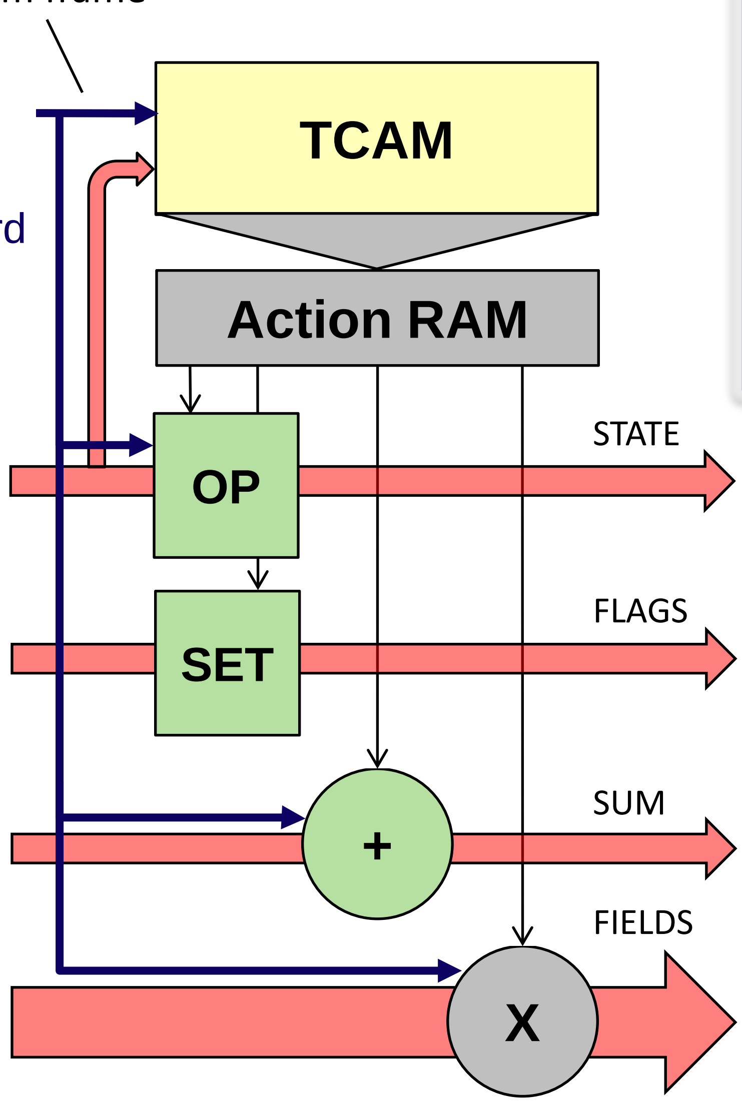
Each stage receives a successive 32b header word from the frame

State Transformation

Parsing state may be transformed by a specified operation. Maintains byte offset counts, current header type, etc.

Checksum

Header fields may be added to propagated header checksum



```
IF (STATE[Header]==L2 &
    STATE[Offset]==12 &
    HdrWord[1]==TYPE_IPv4)
  STATE[Header]  = IPv4
  STATE[Offset]  = 2
  FLAGS[IsIPv4]  = 1
  FIELDS[L2TYPE] = HdrWord[1]
ELSE IF (…)
…
```

Example parsing rule

Flags bus assignment

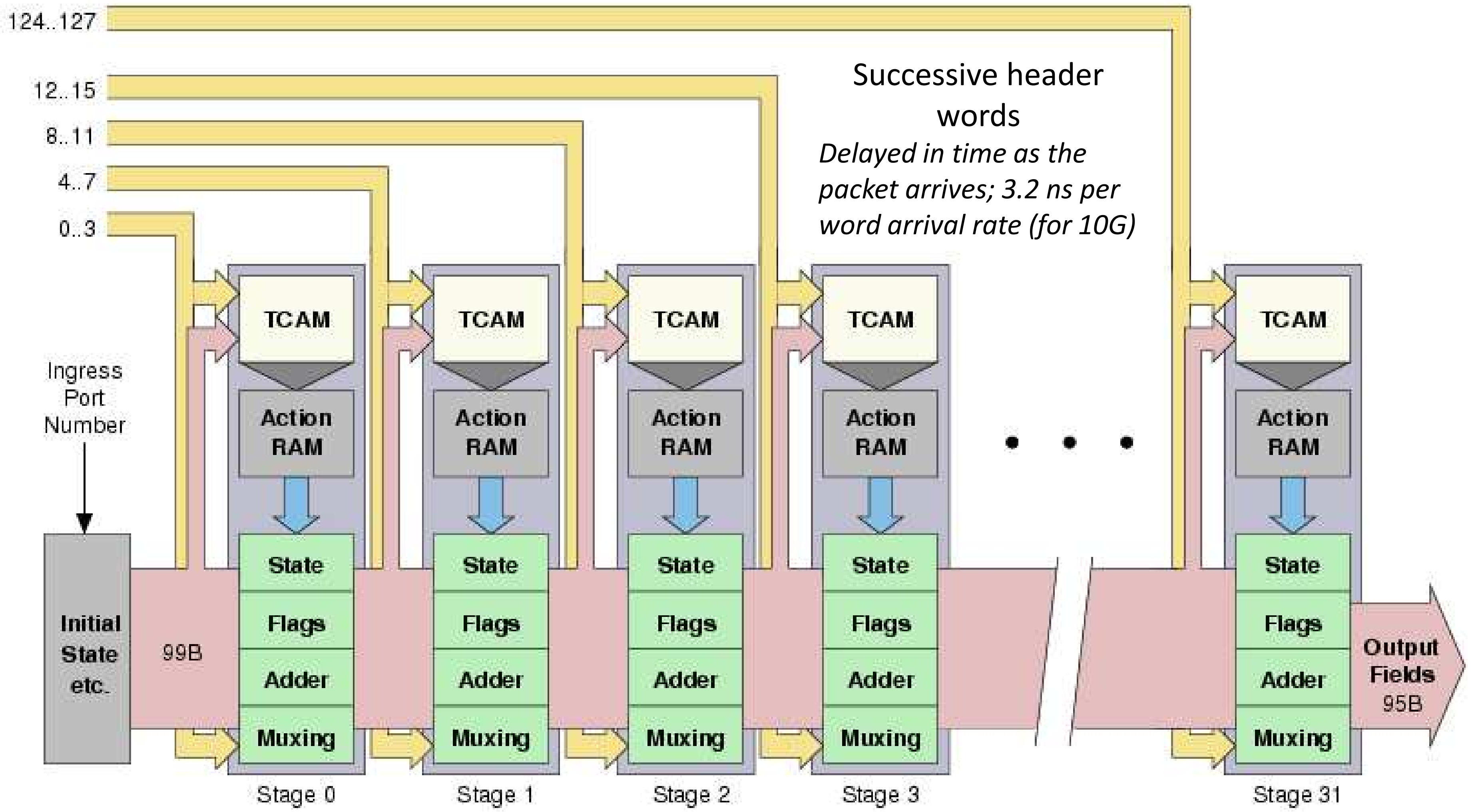
Any of 40 flags bits may be set to propagate frame properties downstream to later stages (e.g. "frame is IPv4")

Output Fields Muxing

Each half-word from the header may be assigned to any of 44 16b fields in the output bus, with an optional bit roll applied

FULCRUM microsystems

Configurable Parsing Pipeline



Successive header words

Delayed in time as the packet arrives; 3.2 ns per word arrival rate (for 10G)

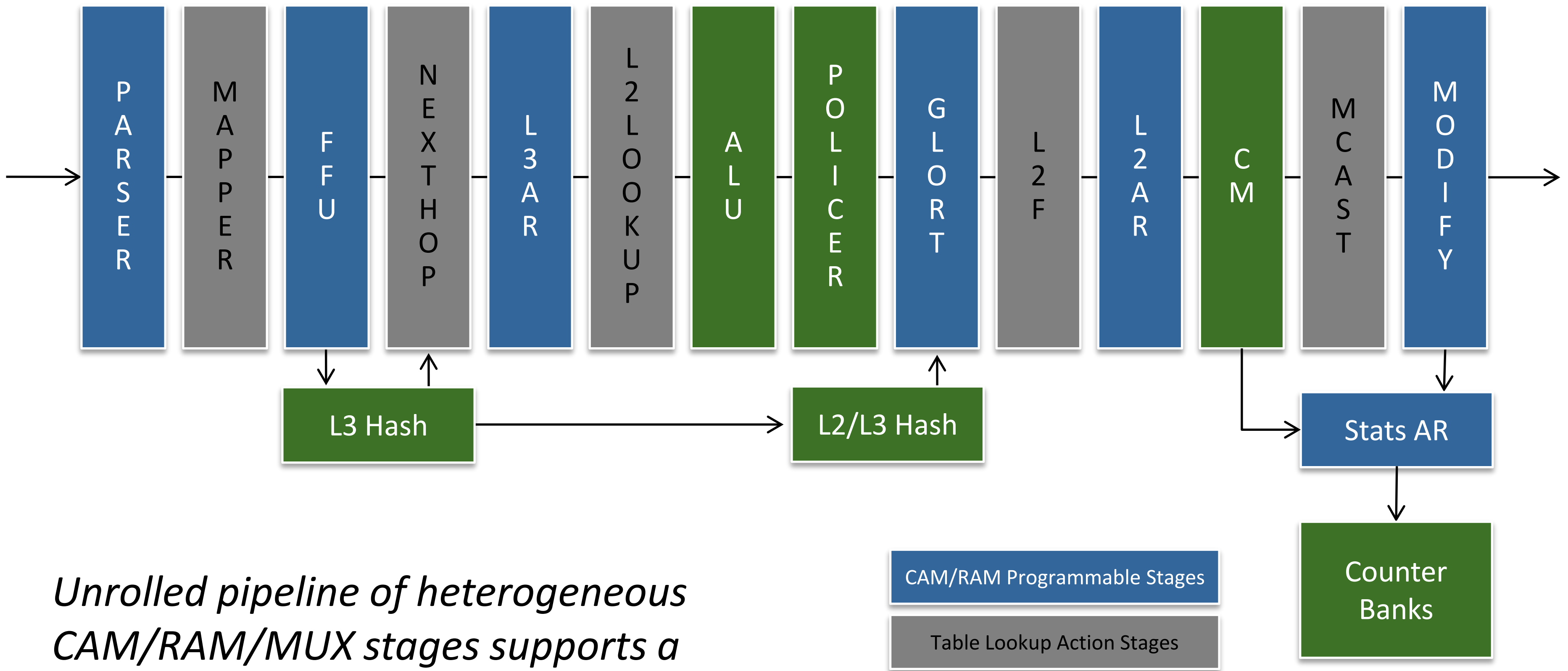
Implements a loop-unrolled, fully pipelined parsing state machine

(fixed max parsing depth of 128B)

FULCRUM microsystems

FlexPipe™ Packet Processing Pipeline

Repeating the theme produces a highly configurable pipeline…



Unrolled pipeline of heterogeneous CAM/RAM/MUX stages supports a superset of many possible frame processing profiles

FULCRUM microsystems

FlexPipe™ Packet Processing Pipeline (2)

Pipeline as a whole is heterogeneous

Different sections of the pipeline are tailored for different functions, and support different fixed-function actions.

CAM/RAM Programmable Stages	Stages optimized for matching and classification with simple, mux-style actions (emphasis on TCAM)	PARSER, FFU, L3AR, L2AR
Table Lookup Action Stages	Stages optimized for directly mapping previously assigned fields to new fields (emphasis on large RAM)	NEXTHOP, L2F, MCAST
Fixed-Function Action Stages	Stages optimized for applying specific fixed-function actions (emphasis on Action logic)	ALU, POLICERS, COUNTERS

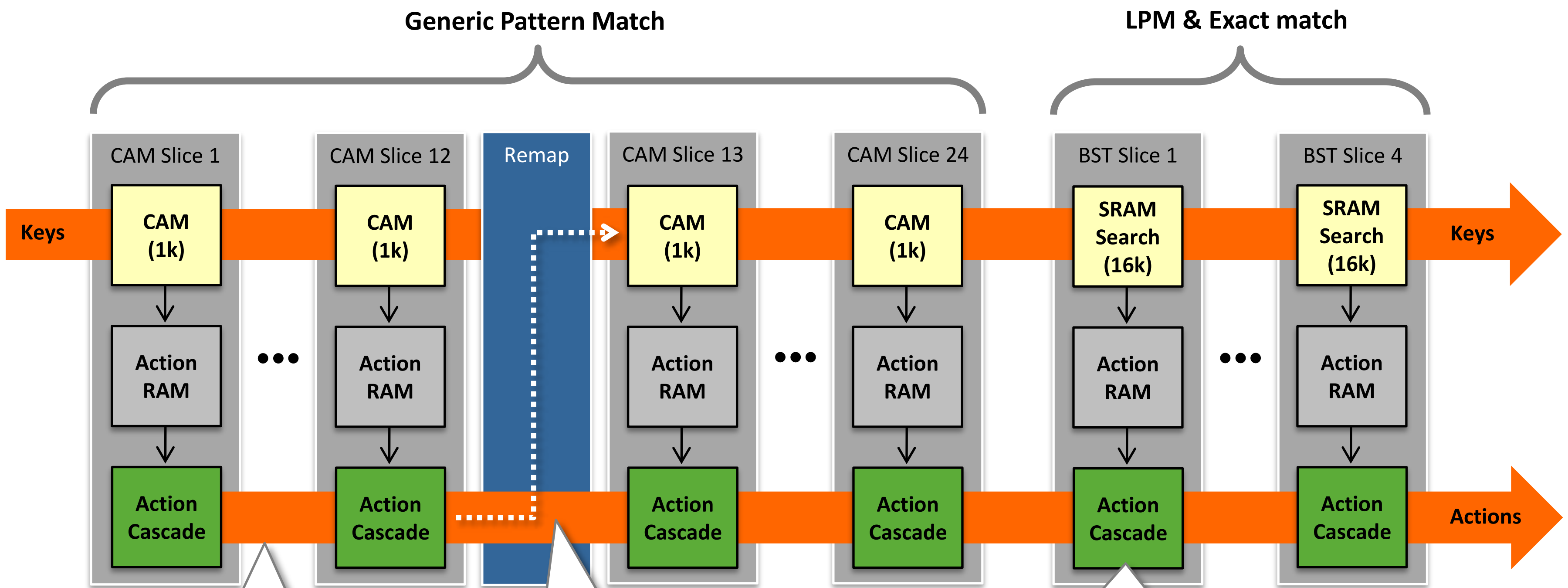
Pipeline structure determined by typical frame processing profiles:

Parse raw frame header to extract fields of interest	PARSER
Classify extracted fields and match against specific address fields	MAPPER, FFU
Lookup a “next hop” destination (either for routing or tunneling)	NEXTHOP
Interpret aggregate output from prior stages, assign post-route L2 fields	L3AR
Perform destination and source address lookups (e.g. DMAC/SMAC)	L2_LOOKUP
Map results of prior lookups to an internal switch destination port mask	GLORT
Filter destination port mask based on classified frame properties (e.g. VLAN IDs)	L2F
Interpret once again the aggregate output from prior stages, finalize forwarding	L2AR
Transform frame header on egress based on earlier classification & mappings	MODIFY

FULCRUM microsystems

Example: Filtering/Forwarding Unit (FFU)

CAM and Binary Search Tree (BST) Stages for Efficient Header Matching



Build up an instruction of actions for future processing, up to one action per stage in a single pass

Output of first 12 slices available as keys to second set. 2nd order iterative matching.

BST table delivers 10x less power and area per bit searched vs. TCAM. Combined, TCAM and BST efficiently offer a large search space (88k rules).

Applications: ACLs, DIP/SIP matching for routing, DMAC/SMAC matching, etc.

FULCRUM microsystems

Example Fixed-Function Action Stages

- **Binary Search Tree (64K entries)**
  - Associative match for keys 32b to 128b wide
  - Keys are configurably generated from header/derived fields
- **Hash Table (64K entries)**
  - 60B key for MAC Address Lookups
  - Supports dual lookups & HW learning
- **ALUs (6 x 16b)**
  - Comparisons between header fields and/or derived fields
  - Table index computation
  - Checksum adjustment
- **Hash Functions (3)**
  - Configurable keys up to 74B of header/derived fields
  - Random mode for load balancing
- **Range Compares**
  - Binning (e.g. TCP port ranges, frame lgnths)
- **Policers/Counters (3 banks)**
  - Token bucket rate limiting (for Tri-Color marking)
  - Generic byte/frame counting
- **Statistics Counters (16+ banks)**
  - 32K(+) x 64b total counters
  - 144-way counter parallelism shared between ingress & egress frames
- **Egress Frame Modification**
  - Per-port Byte Serial Modification Engine
  - 1.25 GHz
  - Command & Operand stream generated by CAM/RAM/MUX stages
  - Supports 20 command & 56 operand bytes
  - Supported commands (vectorized): Insert byte, Delete byte, Overwrite byte, Skip, Decrement, Overwrite Checksum

FULCRUM microsystems

Forwarding & Tunneling Resources

Parser/Mapper

Extracts fields, maps them for downstream FFU classification

Can flag special frames for
mirroring, trapping, log
ging, etc.

NextHop Table

64K entries can hold DMAC-VLAN for routing

32K Cascaded entries can hold up to 4 MPLS labels or 2 with routing

GLORT CAM/RAM (4K-32K)

Maps Destination Port Mask

Key sources:

- Ingress ISL tag (from Parser)
- FFU/BST action RAM
- NextHop Lookup result
- (MAC,VLAN) Lookup result

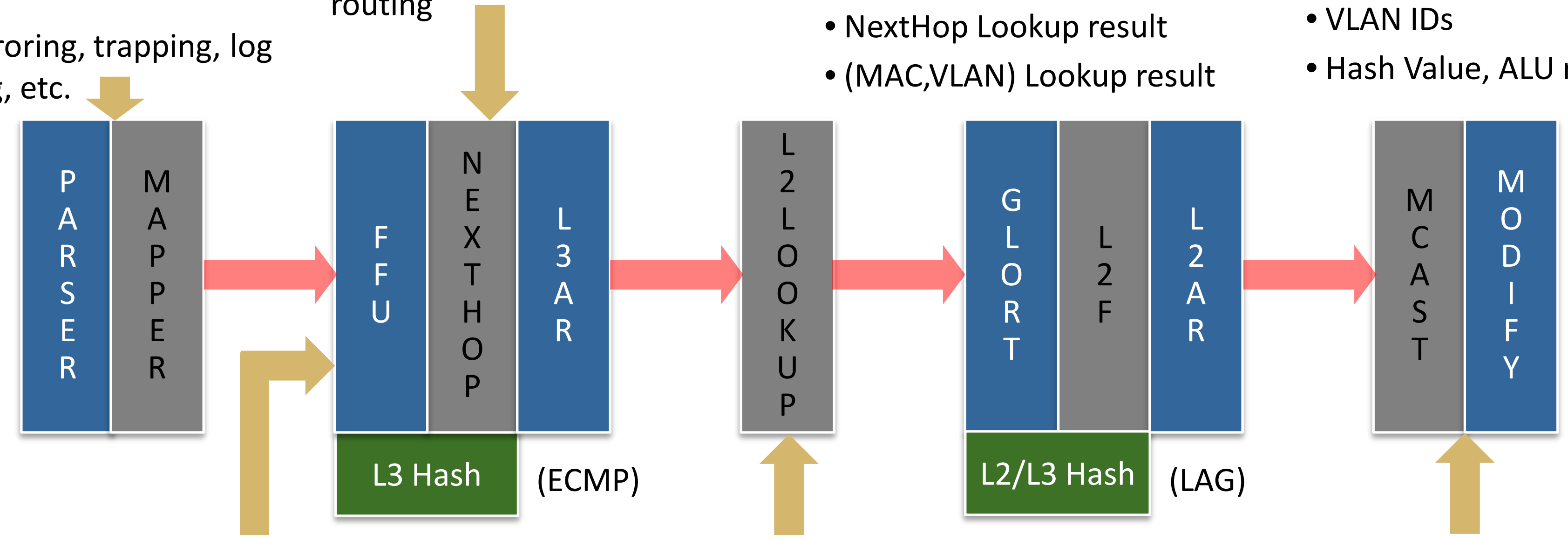
L2F DMASK Tables

8x4K + 4x256 entries

Filters or Maps DMASK by selected index

Example index sources:

- VLAN IDs
- Hash Value, ALU result



FFU/BST Action RAMs

24K TCAM entries for NextHop index or tunnel ID

64K LPM entries for NextHop index or tunnel ID

Tunnel ID can be 12-bit pointer to egress table

Routing rules and ACL rules share TCAM/LPM resources

(MAC,VLAN) table

64K 36-bit entries matched to (DMAC,VLAN)

Provides DGlort and Tunnel ID

MCAST and Egress Modify Tables

4 x 4K 16-bit tables (can hold 4K L3 MCAST VLANs or 4K x 2 labels)

Additional tables for VLAN, QoS updates

FlexPipe™ Capabilities

- **12 Stages of Iterative Header Processing**
  - 12 degrees of match/map separation between ingress & egress header fields
- **TCAM Lookup Resources**
  - 16 architecturally distinct TCAM stages, 175 instances
  - 24K rules in FFU stage optimized for Routing/ACL use
- **Table Lookup Resources**
  - 40 architecturally distinct table lookups, 300+ instances
  - 64K Next Hop Routing Entries, 64K MAC Entries
  - 33K Forwarding Port Masks, 32K Egress Multicast Entries
- **Extensive Parsing/Modification Flexibility**
  - 128B parsing depth, 88B field extraction bandwidth
  - 56B of header can be added/replaced on egress, 160B deleted
- **Configurability supports a microcode programmable usage model**

Fully Provisioned for 1.1 Billion PPS Processing

Frame Processing Wish List (Revisited)

Bonus Features

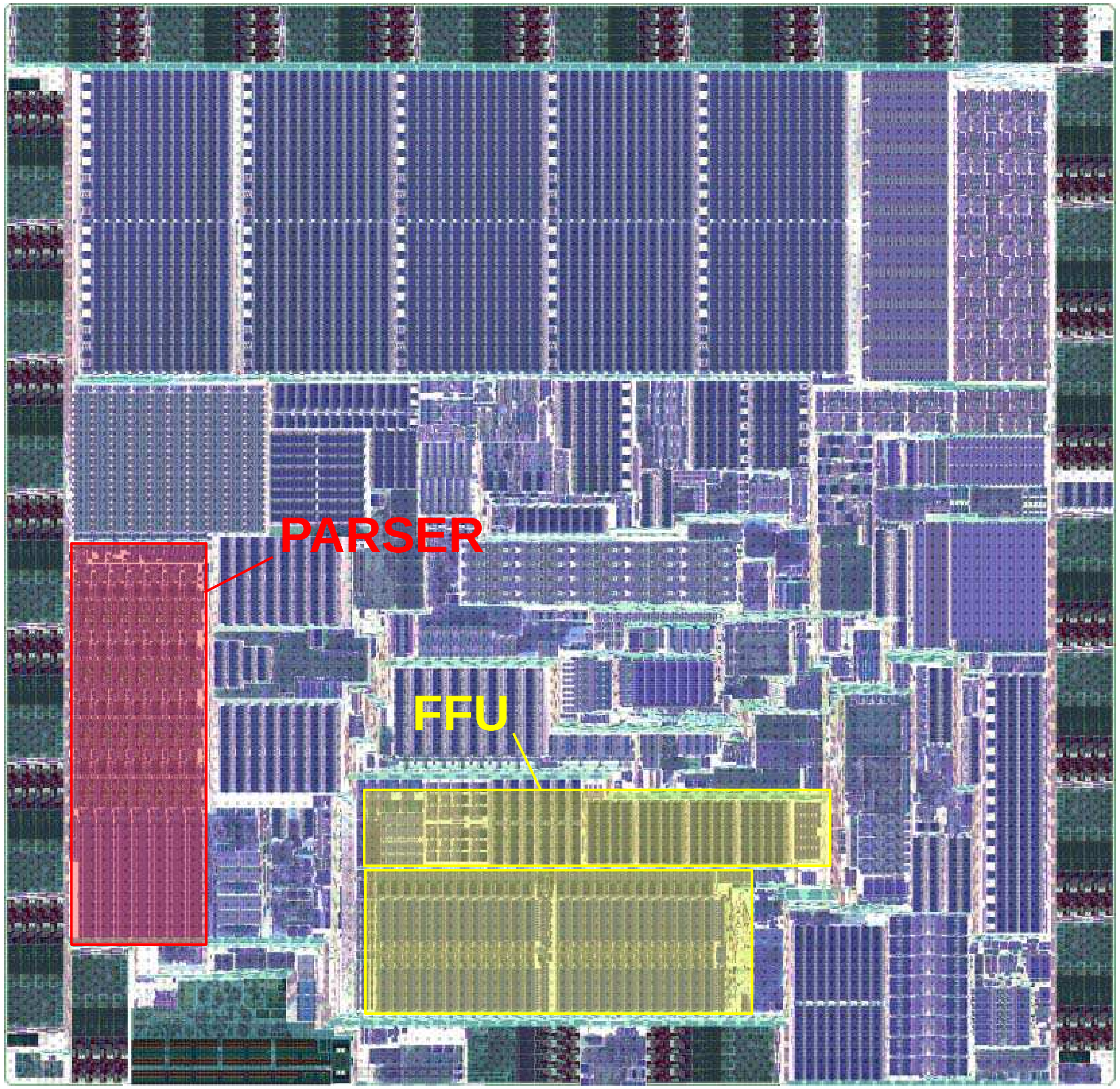
- FCoE, Fibre Channel Forwarder
- Native Fibre Channel parsing
- IP-in-IP
- IP Translation (v4-to-v6, v6-to-v4)
- GRE
- Network Address Translation
- VEPA, VEPA+
- VN-Tag
- Edge Virtual Bridging
- VPWS, VPLS
- OpenFlow v0.9, v1.0, v1.1, …
- LISP (Locator/ID Separation Protocol)

Features determined to be supported by Alta after specification closure, thanks to pipeline configurability

Implementation Challenges

- **A large, complex chip – despite simplifications**
  - 1.2B transistors
  - Chip-wide logic area breakdown:
    10% Synchronous, 13% Proteus, 31% Crossbar, **46% Custom Flow**
- **Small development team**
  - 30 engineers
  - Increased head count to 40 with new hires, interns, and contractors
  - Software/Systems team tasked to help with verification, chip assembly, and low-speed RTL design
- **Proteus flow challenges**
  - Typical case of just-in-time design flow innovation
  - Limited block capacity required more manual design decomposition
  - Large and/or high-latency results required selective custom redesign
- **Result: Area and Schedule expansion**

Alta Chip Plot



- **Status: In Fab**
  Samples Q3 '11
- **72x10G / 18x40G**
- **1W per KR port**
  (72W max power)
- **15 MB total memory**
- **400 ns latency**
- **Microcode Programmable Ethernet Switch**