Department of Information Engineering
University of Pisa
TLCNetGroup

NetFPGA: Reusable Networking Open Architecture

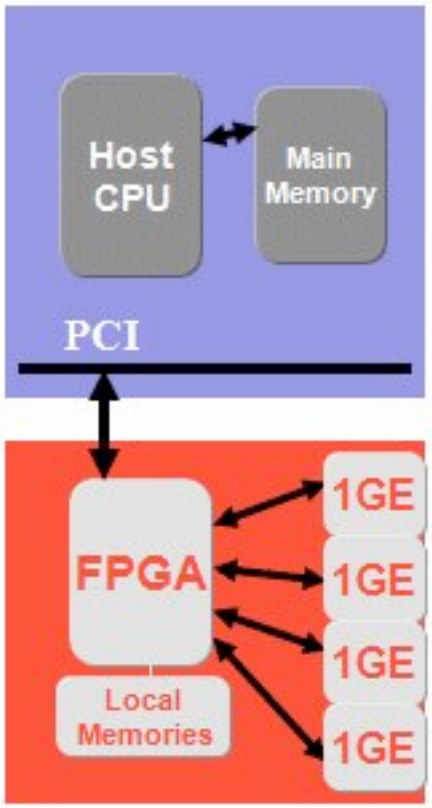
Gianni Antichi

gianni.antichi@iet.unipi.it
http://netgroup.iet.unipi.it

Outline

- ✓ **NetFPGA**
  - *An introduction*
  - *The board*
  - *A platform for research and teaching*
  - *Hardware reference design*
- ✓ **IP Lookup at wire speed**
  - *Bloom Filters*
  - *Blooming Trees for Minimal Perfect Hashing*
  - *Blooming Tree Array*
- ✓ **Ongoing works**
  - *Classification and Pattern Matching using DFAs and Bloom Filters*

NetFPGA: An Introduction



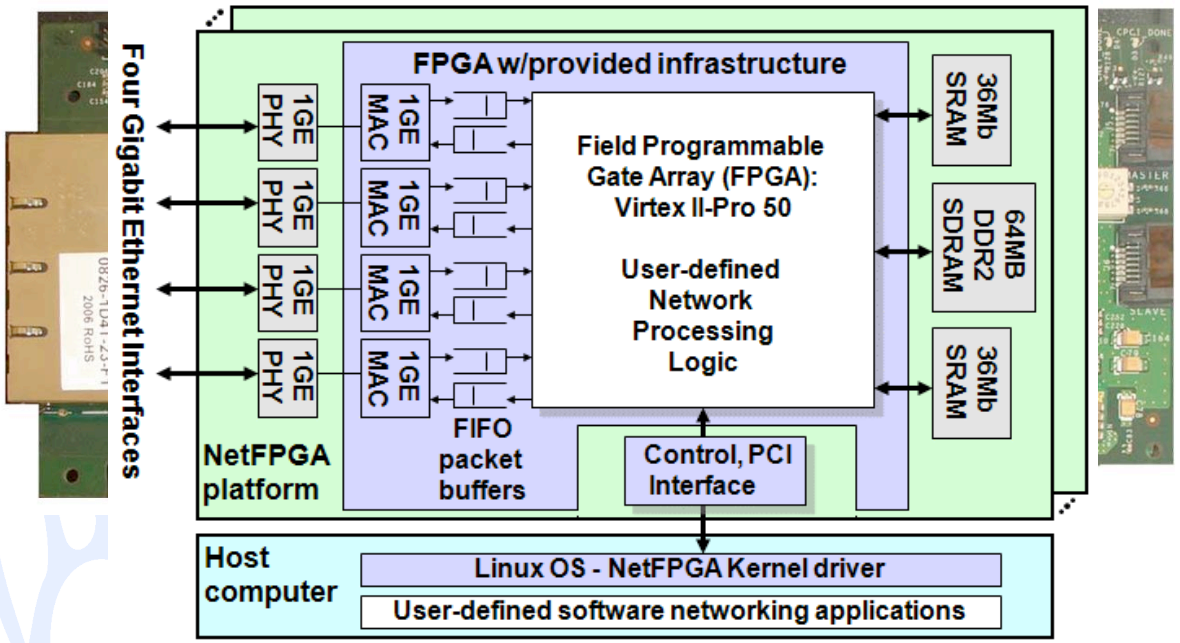
…..from open software to open hardware…..

- Low-cost platform developed by the High Performance Networking Group from Stanford University

- Designed as a tool for teaching networking hardware and router design

- Two different layers:
  - Networking Software running on a standard PC
  - A Hardware accelerator built with FPGA driving Gigabit network links

NetFPGA: The board

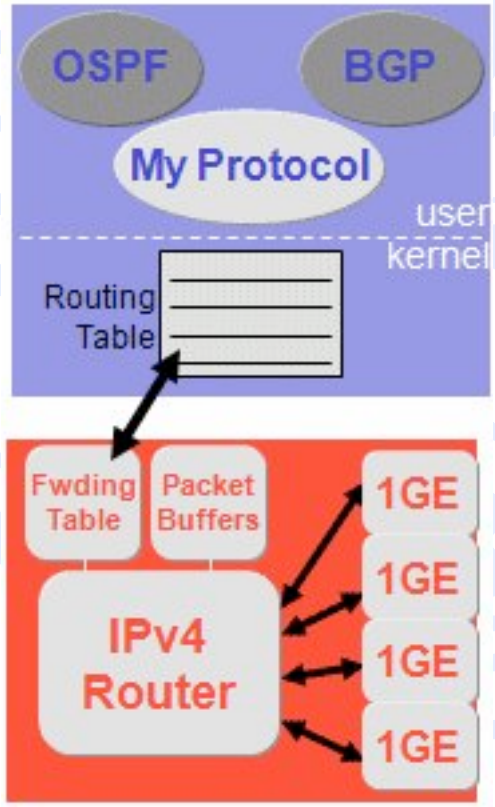


- 2 FPGAs
  - Xilinx Virtex II-Pro 50 (programmed with user defined logic)
  - Xilinx Spartan II (manage the PCI interface)
- 4 Gigabit Ethernet ports
- 9 MB of SRAM equally divided in 2 banks
- 64 MB of DDR2 DRAM

NetFPGA: A platform for research and teaching

- **Who?**
  - Teachers
  - Researchers
- **How?**
  - Build modular designs (IPv4 router, 4-port NIC..)
- **Why?**
  - Better understand the operation of networking systems
  - Prototype new networking systems

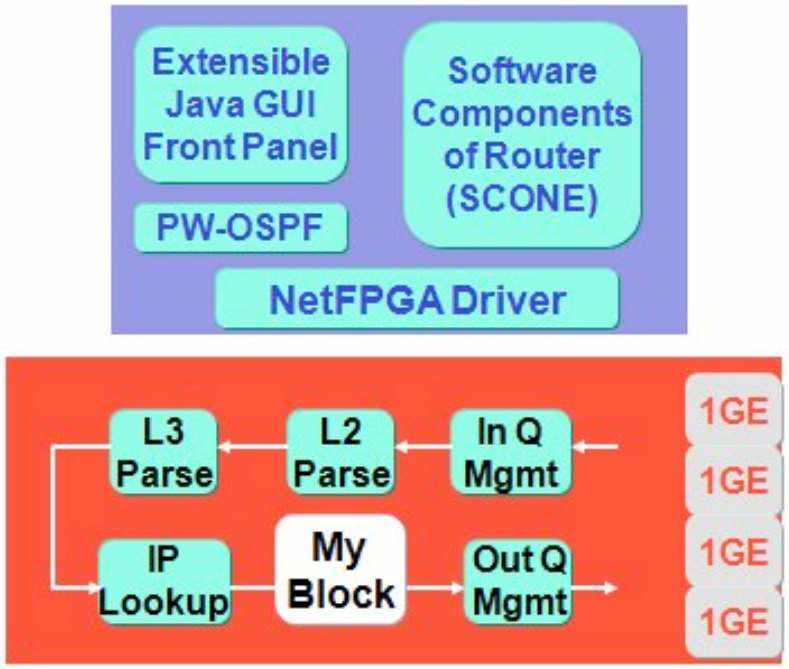
NetFPGA: A platform for research and teaching



A Software approach …

- Write C/C++ code running on the PC that host NetFPGA board
  - Test new routing protocols
  - Implement a control plane for a new hardware project
- Understand the control paradigm of a hardware accelerated networking system

NetFPGA: A platform for research and teaching



A Hardware approach…

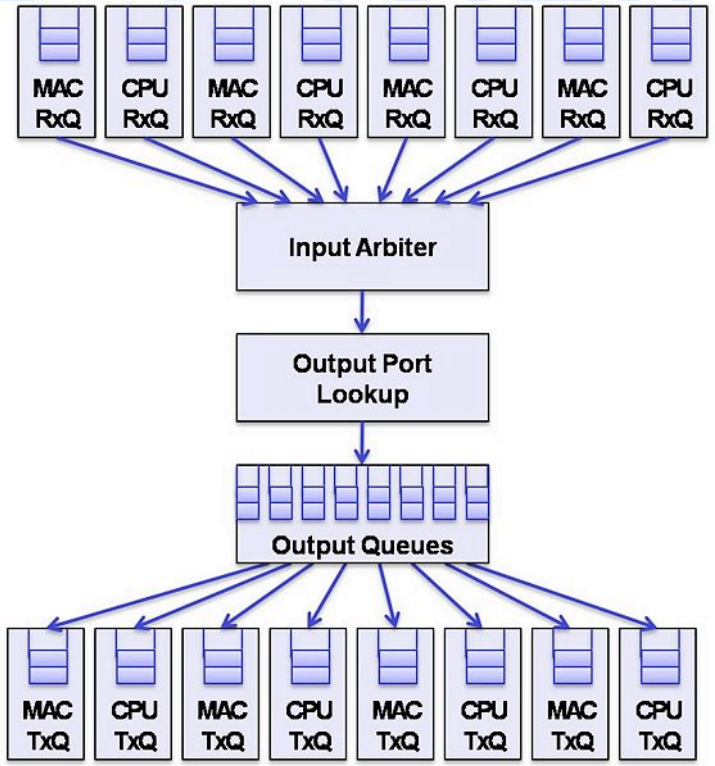
- Write Verilog code running on the Virtex II-Pro FPGA
  - Compile the code using industry standard sesign tools (ISE Foundation)
  - Download the new bitfile in the FPGA through the PCI
- Test new algorithms for hardware accelerated networking systems (Lookup, Classification, Pattern Matching…)

NetFPGA: A platform for research and teaching

…or…a Clean Slate approach

- Implement a new design from scratch
  - Customize software
  - Customize hardware
- Test new and different networking systems

NetFPGA: Hardware reference design



- User Data Path:
  - Input Arbiter (serves the Rx Queues in a round robin fashion)
  - Output Port Lookup (ARP, IP lookup..)
  - Output Queues (stored in SRAM)
- 8 Rx/Tx Queues (4 from/to ethernet ports, 4 from/to Host PC)
- Packets are passed between modules using a fifo-like simple push interface

IP Lookup at wire speed: Bloom Filters

• An m-bits bitmap represent a set of n elements

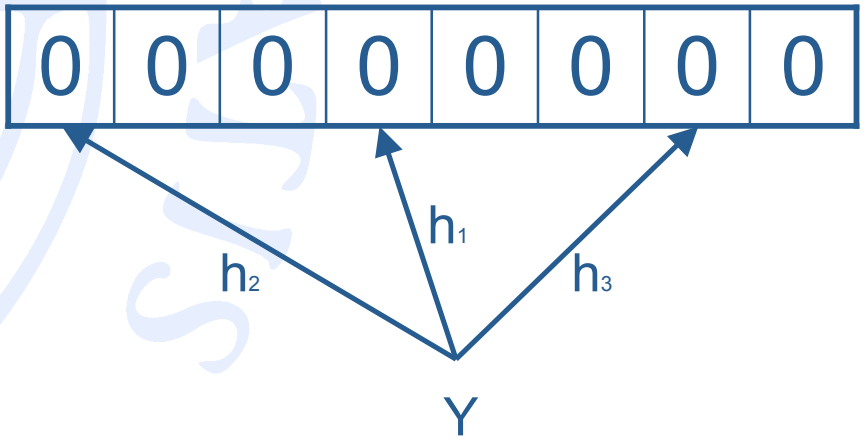
• Easy Set-Membership Lookup by checking k bits addressed by k hash functions: h_1, h_2, …, h_k

• False positives (minimum probability if m = nk/ln2)

•Cannot delete elements

Operations:

Insertion of Y in set X



IP Lookup at wire speed: Bloom Filters

- An m-bits bitmap represent a set of n elements

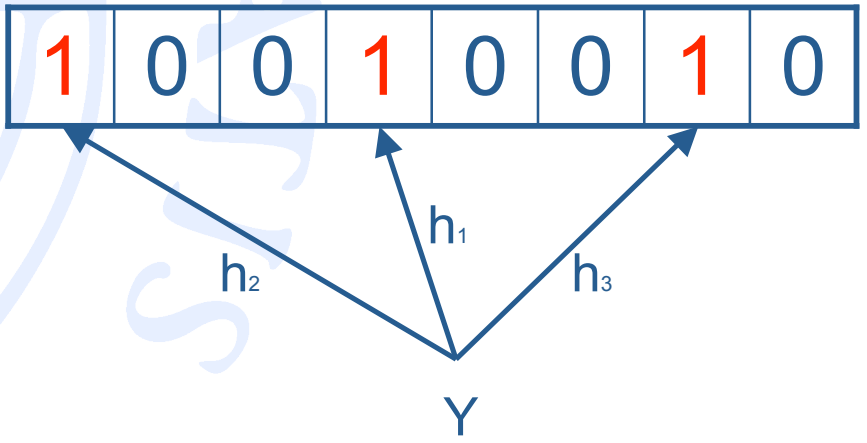
- Easy Set-Membership Lookup by checking k bits addressed by k hash functions: h_1, h_2, …, h_k

- False positives (minimum probability if m = nk/ln2)

- Cannot delete elements

Operations:

Insertion of Y in set X



IP Lookup at wire speed: Bloom Filters

• An m-bits bitmap represent a set of n elements

• Easy Set-Membership Lookup by checking k bits addressed by k hash functions: h_1, h_2, …, h_k

• False positives (minimum probability if m = nk/ln2)

•Cannot delete elements

Operations:

Lookup of s ε X

s

IP Lookup at wire speed: Bloom Filters

• An m-bits bitmap represent a set of n elements

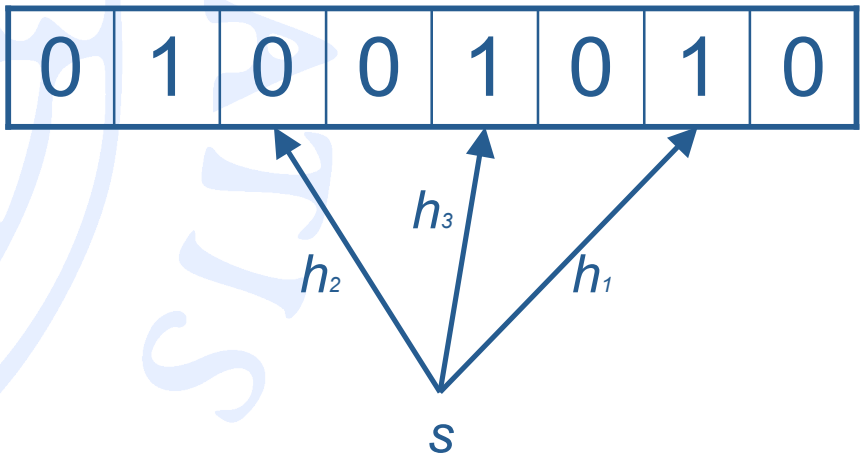
• Easy Set-Membership Lookup by checking k bits addressed by k hash functions: h_1, h_2, …, h_k

• False positives (minimum probability if m = nk/ln2)

•Cannot delete elements

Operations:

Lookup of s ε X

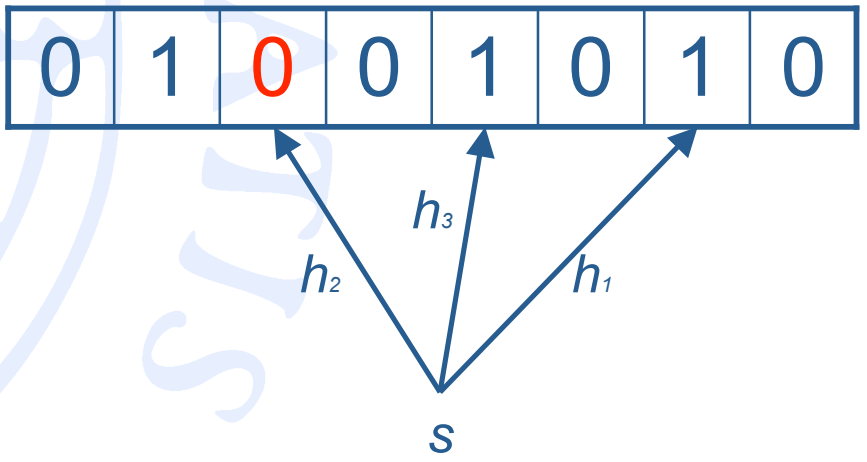


IP Lookup at wire speed: Bloom Filters

- An m-bits bitmap represent a set of n elements
- Easy Set-Membership Lookup by checking k bits addressed by k hash functions: h_1, h_2, …, h_k
- False positives (minimum probability if m = nk/ln2)
- Cannot delete elements

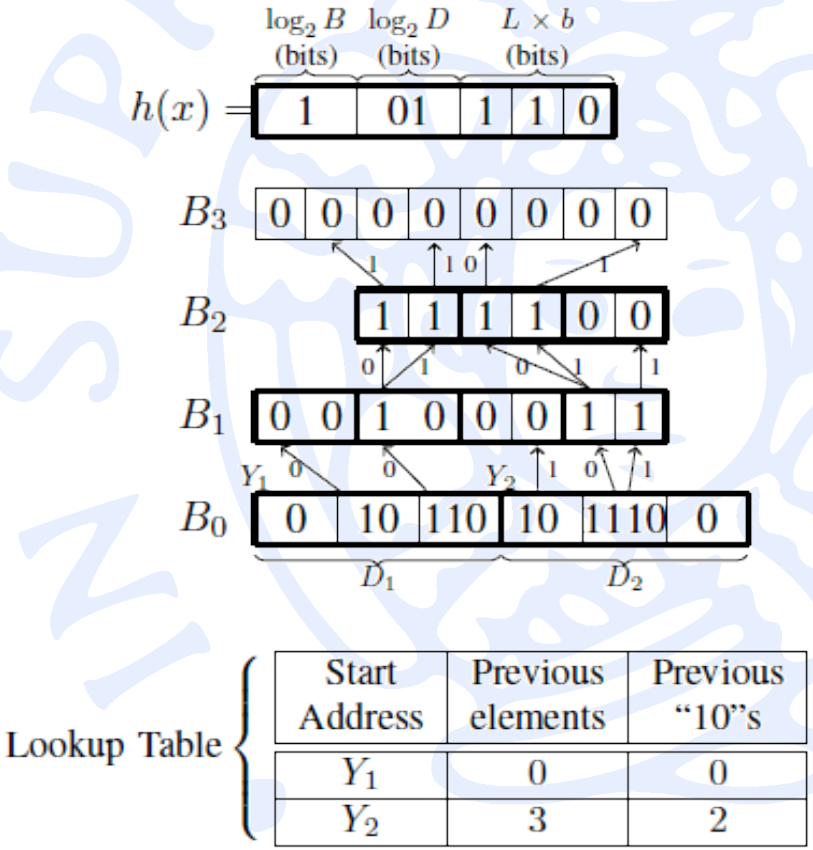
Operations:

Lookup of s ε X



s ε X

IP Lookup at wire speed: Blooming Trees for Minimal Perfect Hashing



Lookup Table

Start Address	Previous elements	Previous "10"s
Y_1	0	0
Y_2	3	2

- A new scheme to construct a minimal perfect hash function based on an alternative to Bloom Filters
- The idea is constructing a binary tree upon each element of a plain BF, thus creating a multilayered structure
  - Low false positive probability
  - Low memory requirements

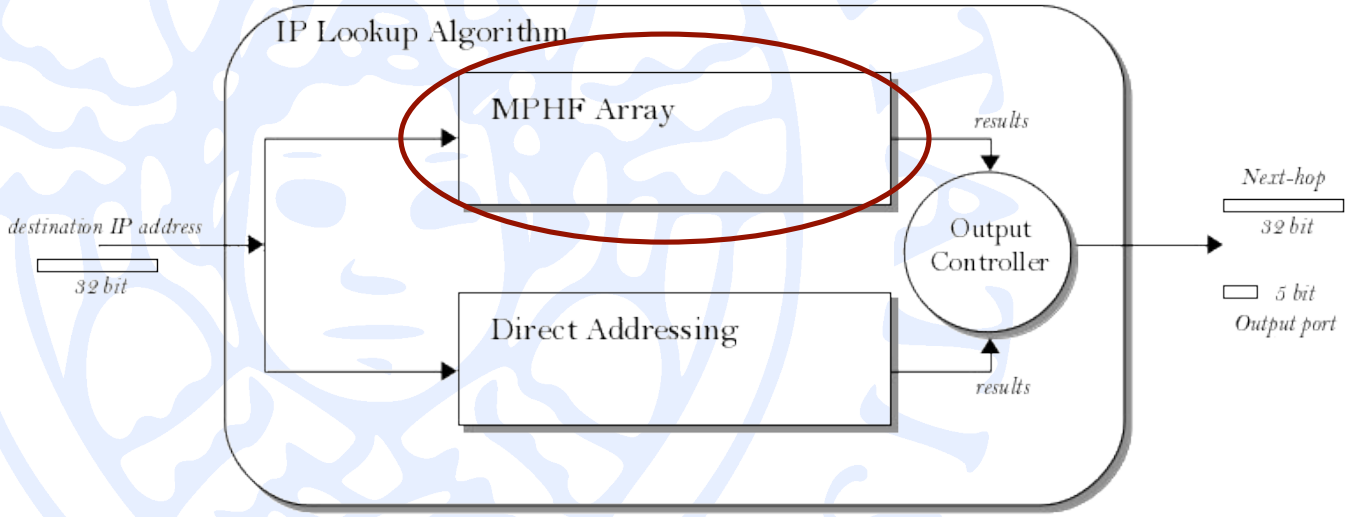
IP Lookup at wire speed: Blooming Tree Array

- An analysis of many forwarding tables of BGP routers has taken out an interesting statistical property

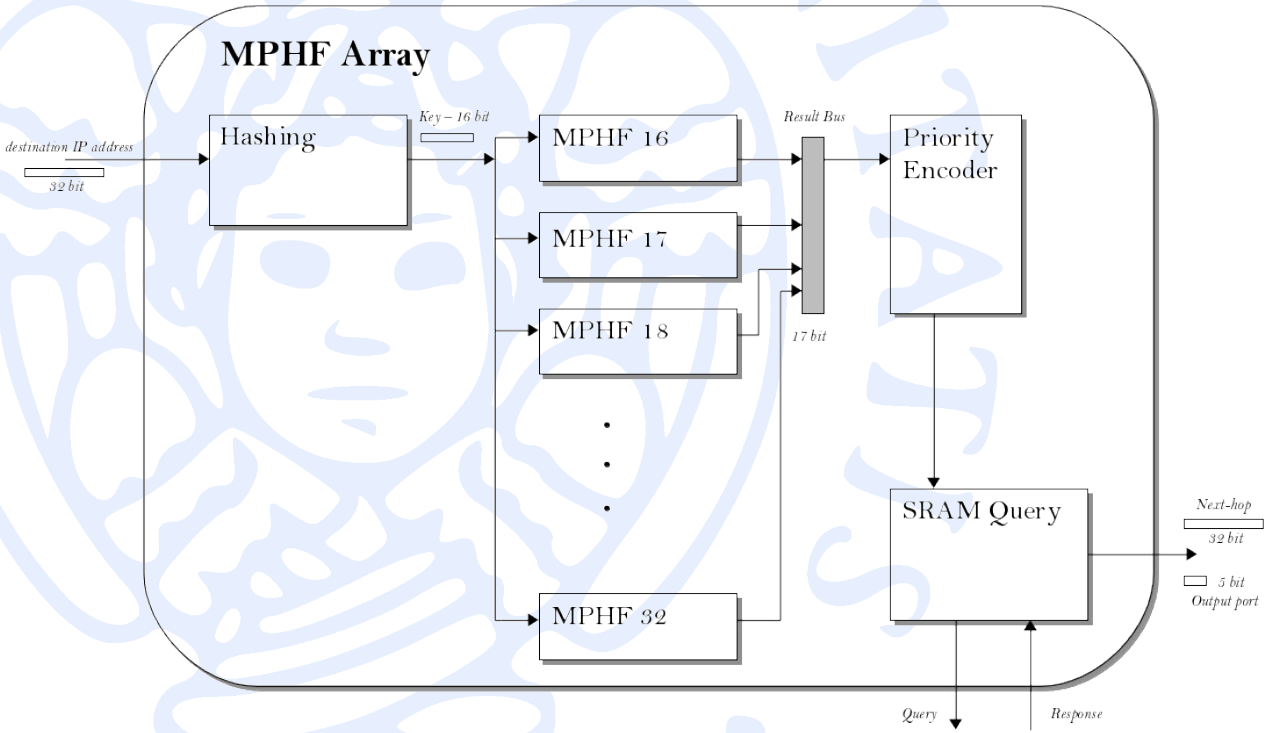
- Presence of very few prefixes with less than 16 bits
  - Address the lookup problem for these entries as an exact lookup
  - The lookup rules with prefixes shorter than 15 bits can be normalized
  - No significant increase of memory footprint

- We could use two different approach in order to take into account this property
  - We use a MPHF for each prefix length greater than 15
  - We use a single Direct Addressing for other lengths
  - When packet arrives, checks all the MPHFs and DA at the same time

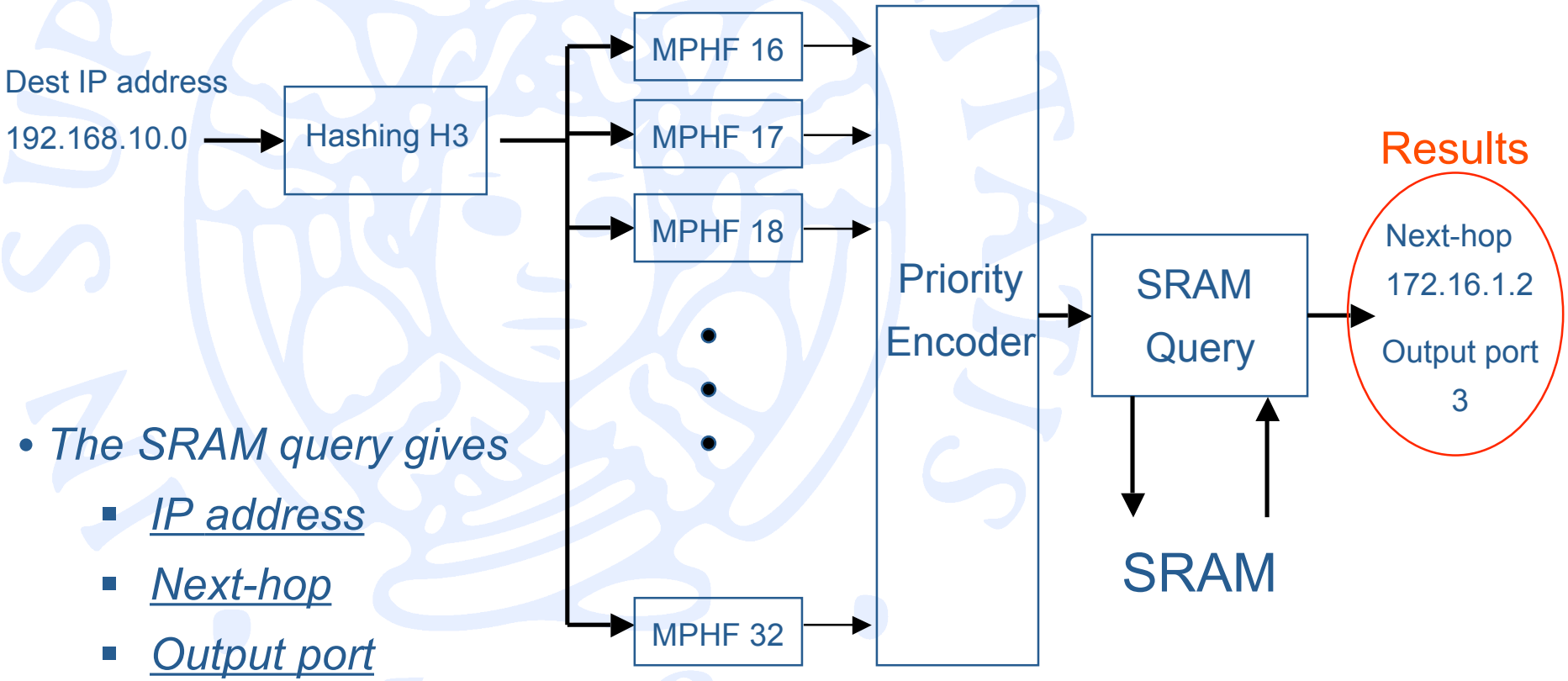
IP Lookup at wire speed: Blooming Tree Array



IP Lookup at wire speed: Blooming Tree Array



IP Lookup at wire speed: Blooming Tree Array



- *The SRAM query gives*
  - *IP address*
  - *Next-hop*
  - *Output port*

Ongoing works: Classification and Pattern Matching using DFAs

- ✓ **Packet Classification at wire speed**
  - *Use Deterministic Finite Automatas (DFAs)*
  - *New solutions to improve the memory occupancy of DFAs*
  - *New solutions to improve the lookup time of DFAs*
- ✓ **Pattern Matching and Anti Evasion at wire speed**
  - *Use DFAs or Counting Bloom Filters*
  - *New solutions to better exploit the memory hierarchy of Network Processors / NetFPGA*
  - New ideas to detect attacks at high speed without reassembly