

**A Configurable Classification Engine
for Polymorphous Computing Architectures**

**Meeta Yadav, Patrick Hamilton, Rick Sears
Yannis Viniotis, Thomas Conte, Paul Franzon**

**Electrical and Computer Engineering
North Carolina State University**

Outline

- > **Motivation**
- > **Polymorphous Chip Architecture.**
- > **Classification Engine**
 - ◆ Features
 - ◆ Sample Rules
 - ◆ Multi field Classifier Scheme
 - ◆ Architecture
 - ◆ Array of Tries
 - ◆ Memory Structure
- > **Performance analysis**
 - ◆ Throughput
 - ◆ Update Rate
 - ◆ Area
 - ◆ Memory Size
- > **Related Work**
- > **Conclusions**

Motivation

> Core objective:

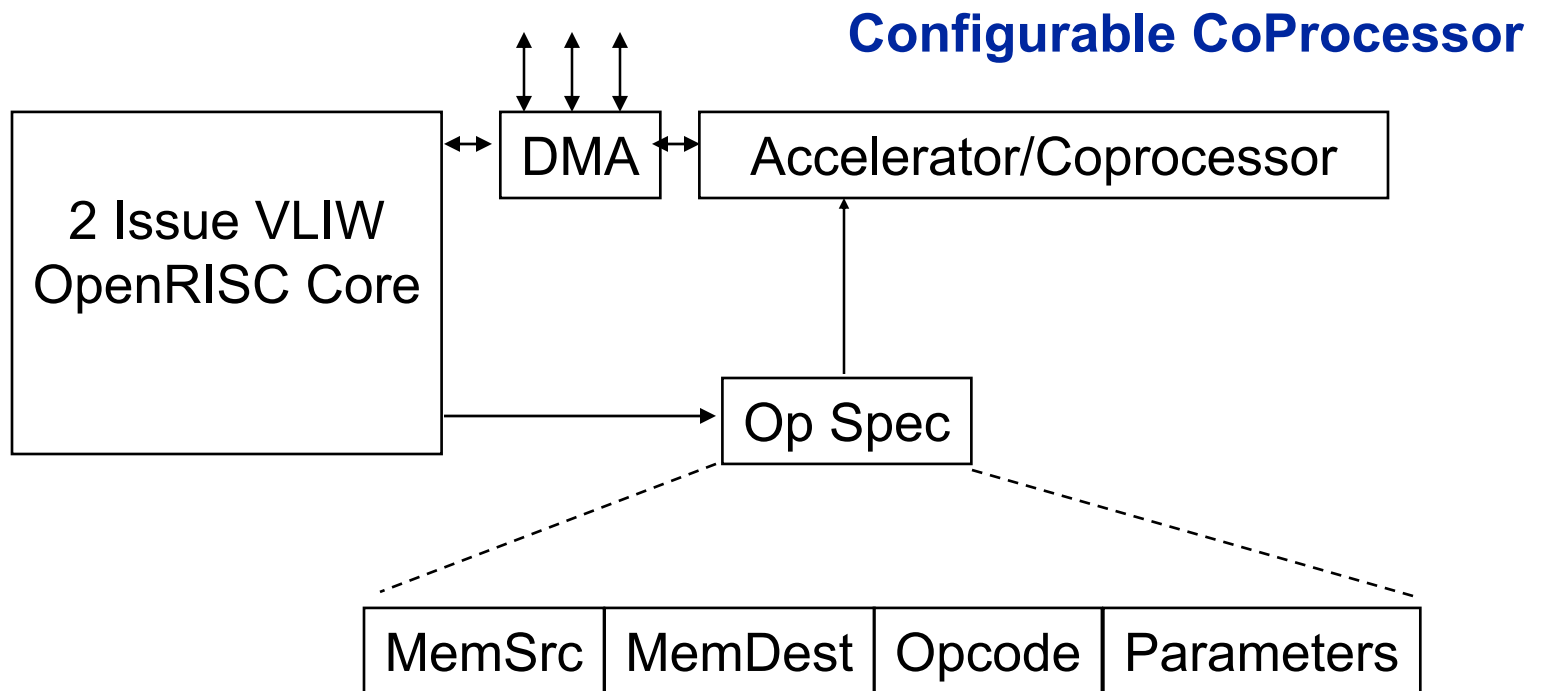
- ◆ Defining Instruction controlled next generation of Network Processors which are:
 - ◇ Feature-rich.
 - ◇ Configurable.
 - ◇ Customizable.

> Specific Tasks:

- ◆ Design a co-processor for IPv4 and IPv6
 - ◇ One common classifier engine
 - ◇ Configurable
 - ◇ Fast lookup and updates
- ◆ Performance focus on lookup rate and update rate.

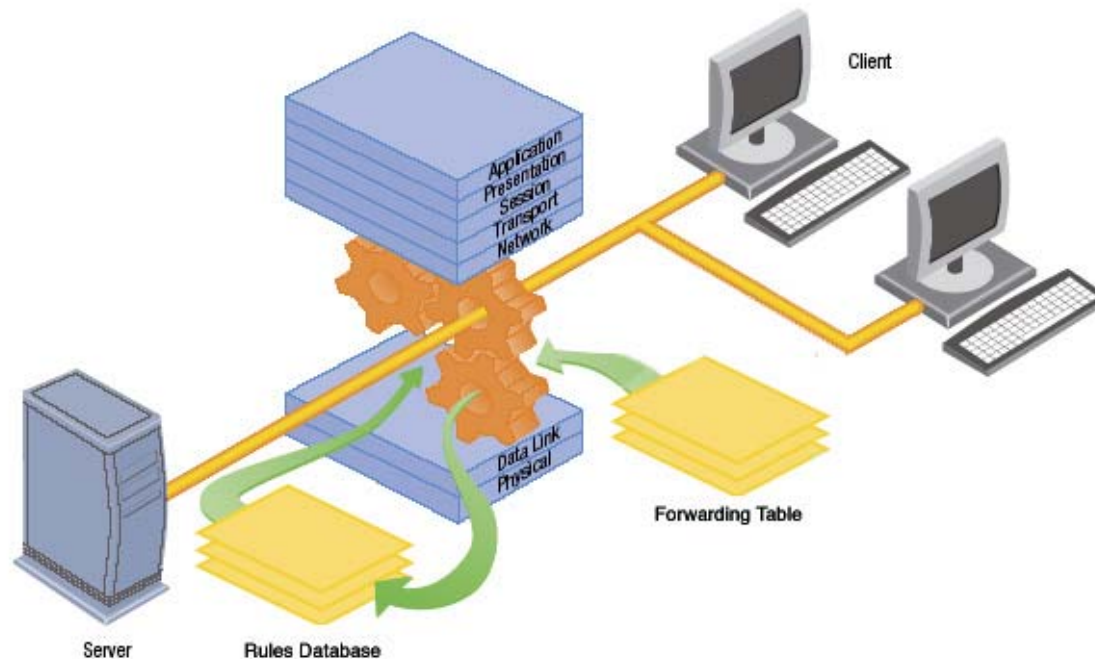
Polymorphous Chip Architecture

- ◆ Configurable coprocessor to execute networking-specific instructions



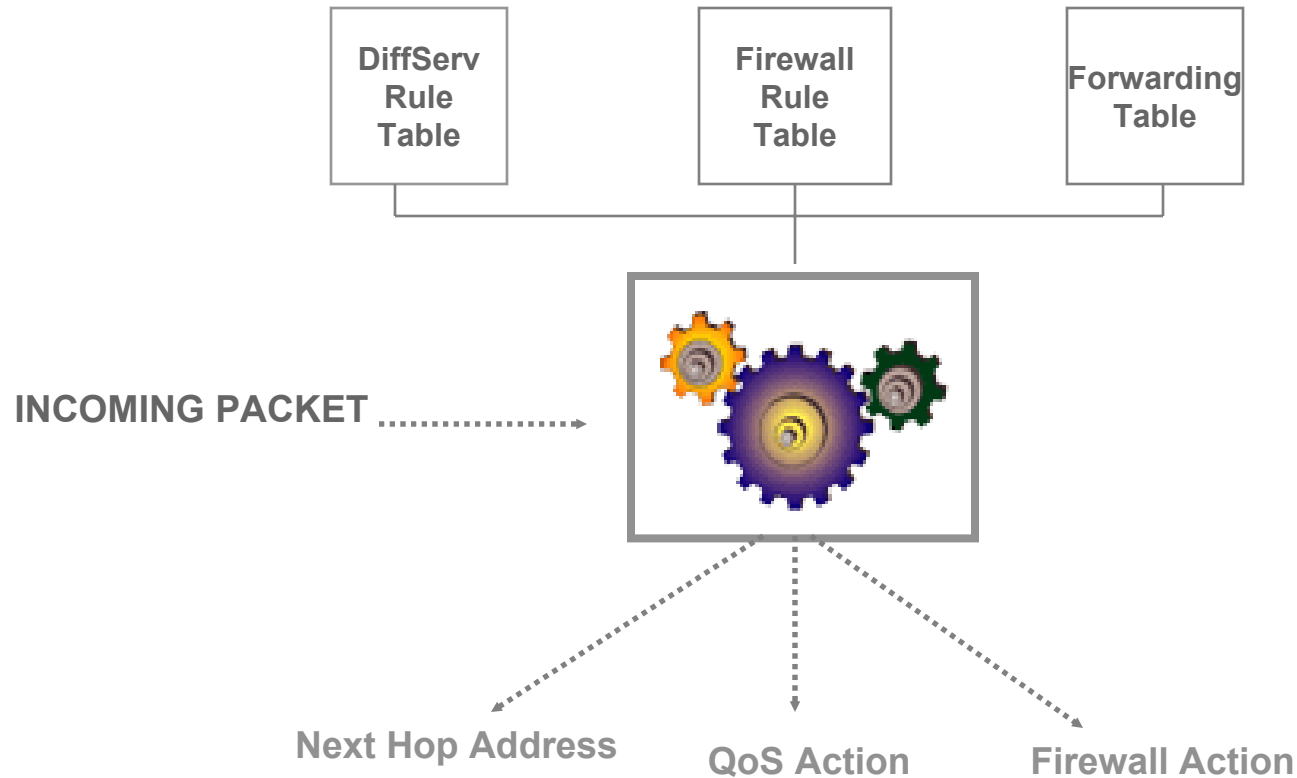
Classification Engine Features

- > To build a Classification engine that is:
 - ◆ Configurable.
 - ◆ Allows for fast updates.
 - ◆ Can handle IPv4 as well as IPv6.



Classification Engine Functions

- ◆ Forwarding
- ◆ Firewall
- ◆ Differentiated Services



Rules

- ◆ **Firewall Rule**

<Source IP> <Destination IP> <Port> <Protocol><TOS><TTL>

- ◆ **DiffServ Rule**

<Source IP> <Destination IP> <Port> <Protocol><TOS><TTL>

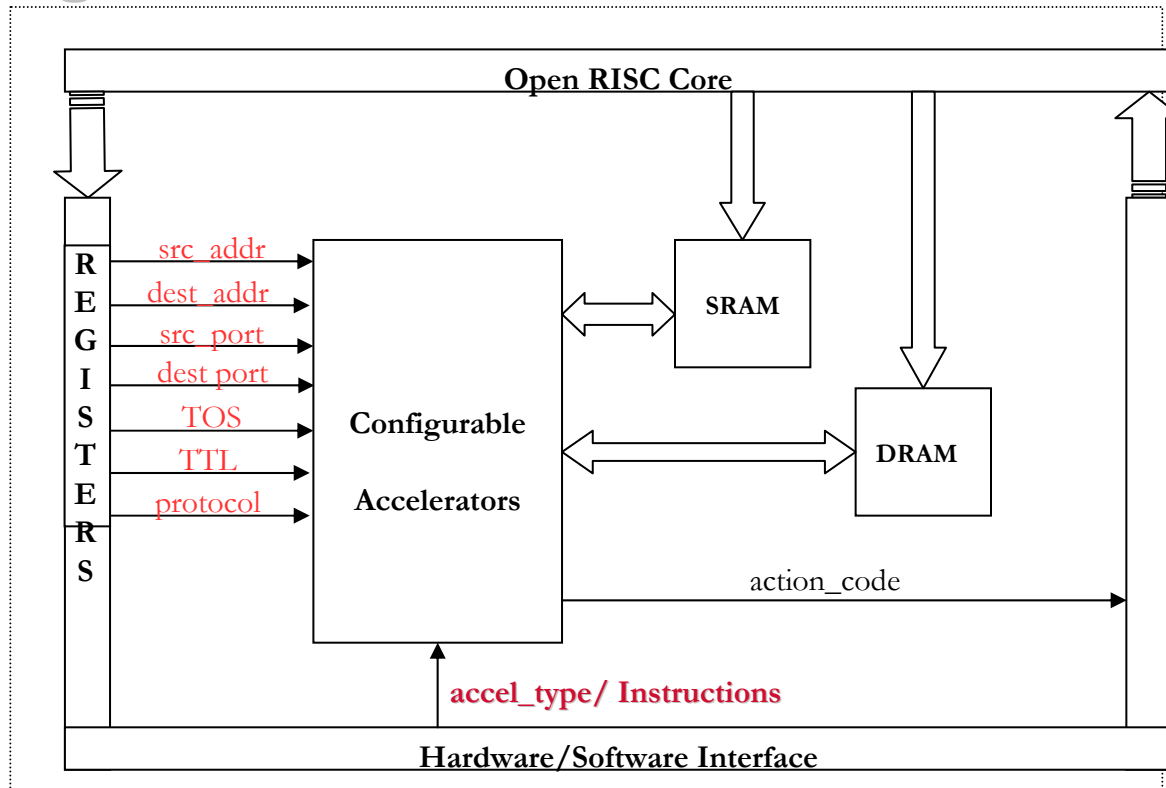
- ◆ **Forwarding Table**

<Destination IP>

Rules

- ◆ **Various fields to be matched**
 - ◇ Source and Destination IP address
 - ◇ Source and Destination Port Numbers
 - ◇ TTL
 - ◇ TOS
 - ◇ Protocol
- ◆ **Final Action**
 - ◇ Firewall - Permit or Deny
 - ◇ Diffserv - Class of Service
 - ◇ Forwarding - Next Hop Address

Configurable Accelerator Architecture



- ◆ Accelerators can be configured as Forwarding/DiffServ/Firewall Engines
- ◆ Accelerators can be controlled by instructions
- ◆ Communication to OpenRISC using Hardware/Software Interface
- ◆ Memories populated by software

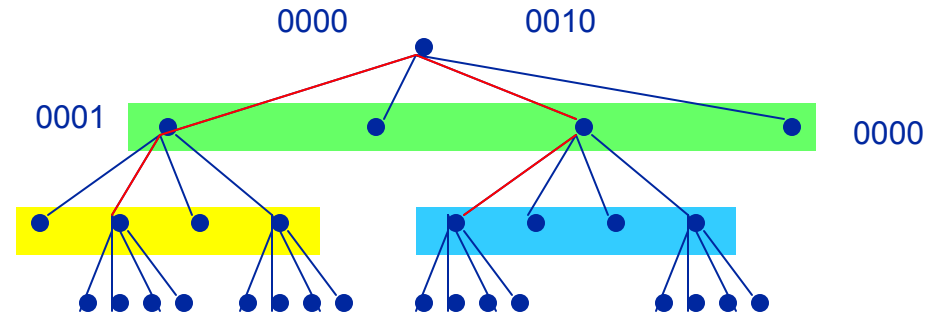
Multi-Field Classifier Scheme

> Implementation

- ◆ We use “ Array of Tries” on Source/Destination pairs to narrow down the search process and speed up the lookup.
 - ◇ The First Tri resolves the source Address and points to second tri
 - ◇ The Second Tri resolves the limited number of Destination addresses
 - ◇ There is a Direct comparison of the remaining fields.

Mehrotra et al Forwarding Scheme

- ◆ Tries store information.
- ◆ The design is static and does not allow for dynamic updates.
- ◆ It is a forwarding engine and not a classification engine.



0000	0001
0010	0000

Compact Trie node/path information:

- 1 → node has child
- 0 → leaf node

“Array of Tries” Architecture

Source IP	Destination IP
x	a
x	b
x	c
x	d
y	a
y	b
y	e
z	l
z	m
z	n
z	o
z	a
z	y

- ◆ Sort the entries based on IP addresses
- ◆ Create Tries of all the unique source IP addresses.
- ◆ Create Tries of the sorted source IP addresses that belong to the same prefix source IP.

Eg

Source IP contains tries of IP addresses x, y, z

Destination IP contains tries of IP addresses

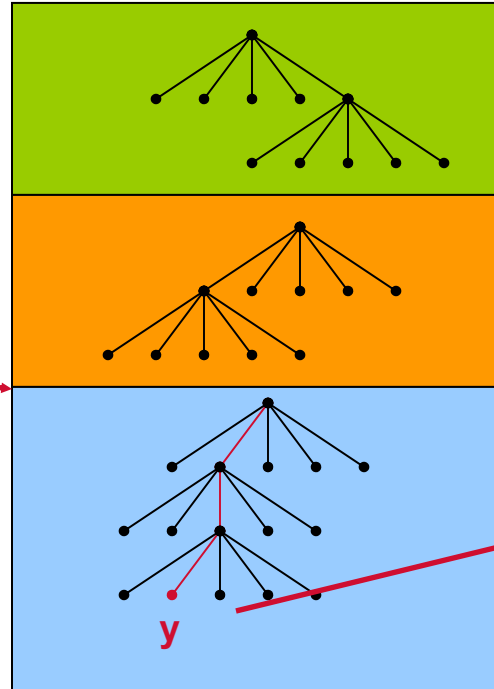
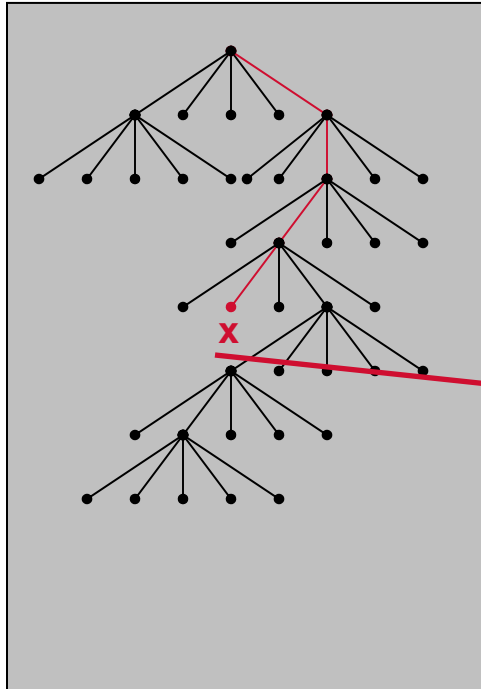
- a. a, b, c, d - In location 1
- b. a,b,c - In location 2
- c. l,m,n,o,a,e - In location 3

Engine Lookup

X <Source IP> Y <Destination IP> PORT <Port> PROTOCOL <Protocol> TOS <TOS> TTL <TTL>

Source IP

Destination IP



In the event of multiple rules
The priority is handled by the
longest prefix IP address match.
The Memory containing the other
Fields also contains a pointer
To the next matching set of data.

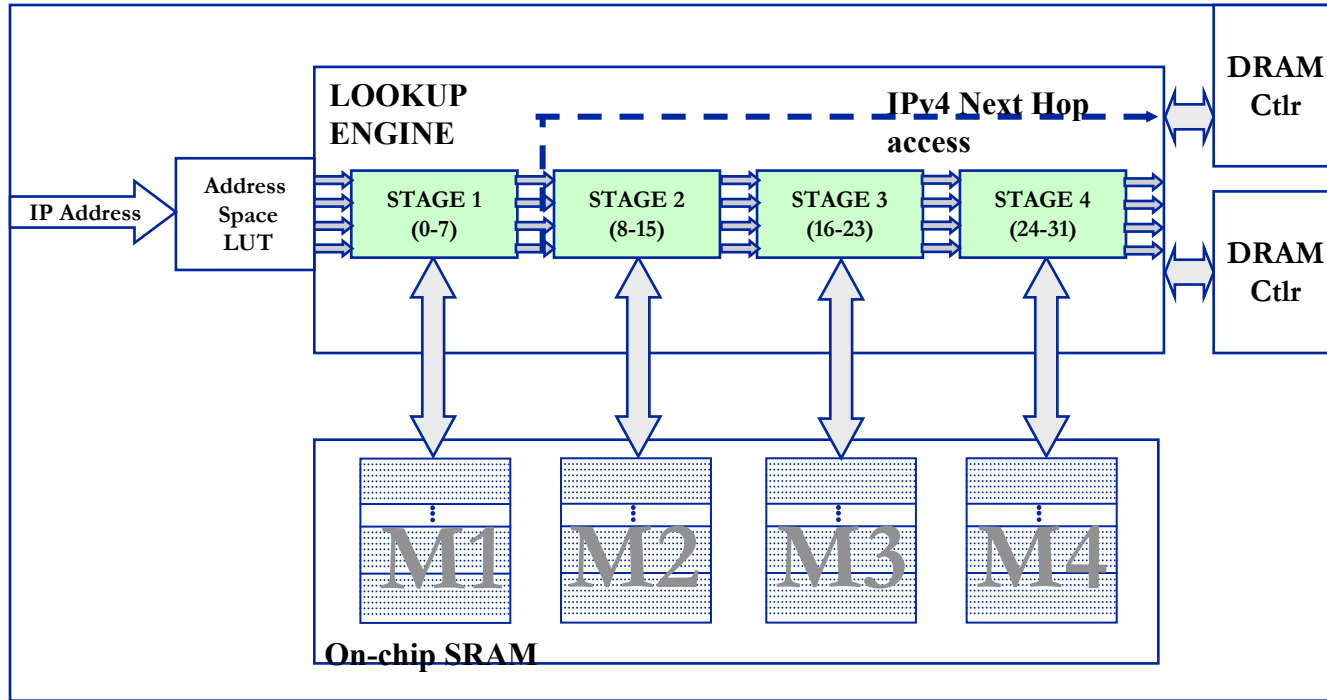
TTL	TOS	PORT	PROTOCOL	PTR
TTL	TOS	PORT	PROTOCOL	PTR
TTL	TOS	PORT	PROTOCOL	

Other Fields

DRAM

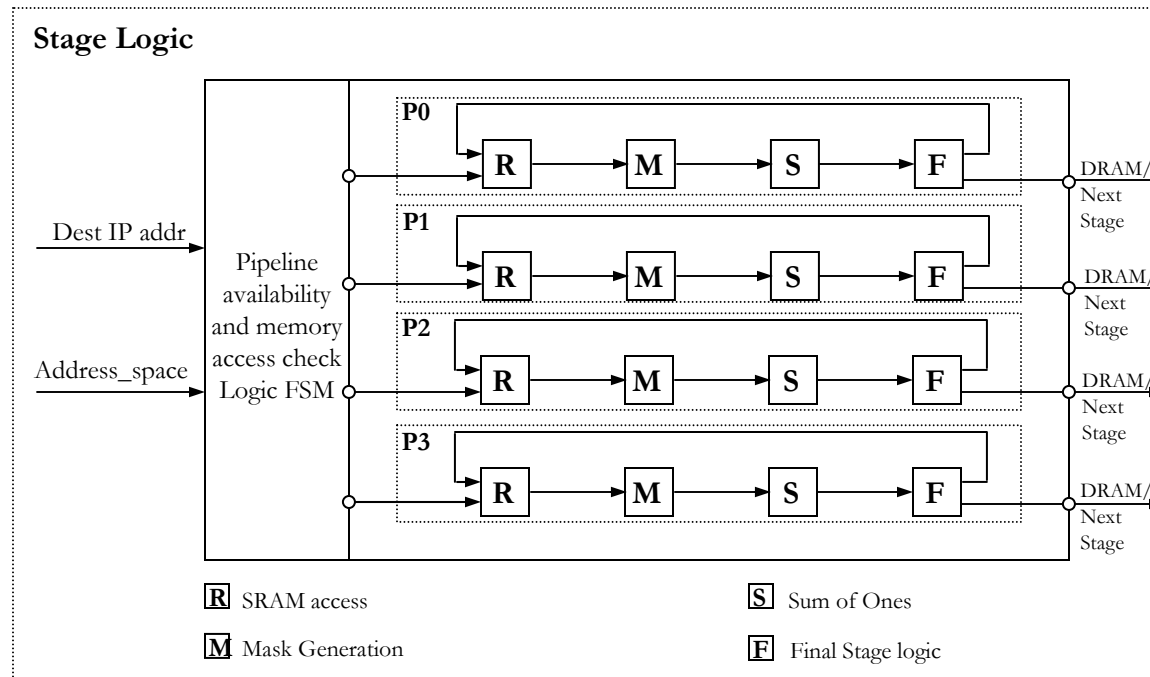
QoS Class	Next Hop Address	Firewall Action
Class A	42	Permit
Class B	34	Deny
Class C	52	Deny
Class B	128	Permit

Architecture



- > Four Stages with four internal pipelines.
- > Address space LUT determines the corresponding address space
 - ◆ Software configured
- > Memories partitioned on Address spaces

Single Pipeline Stage



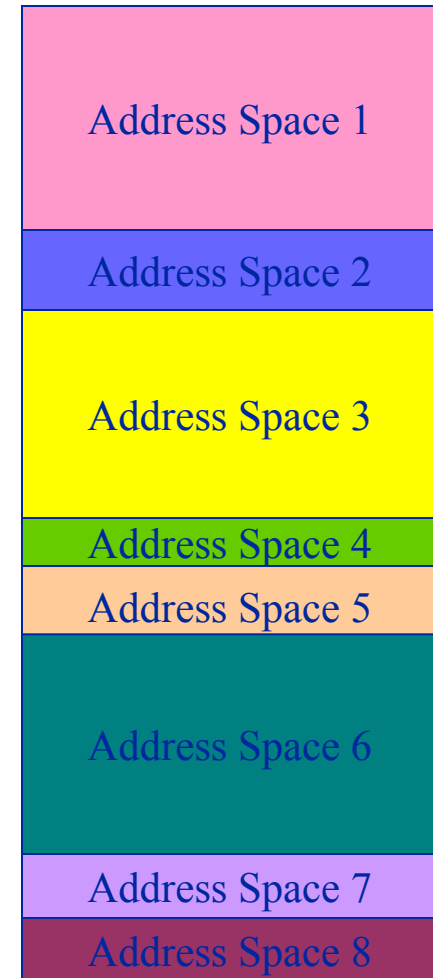
- > Each Stage has an internal pipeline of 4.
- > An FSM controls entry of packets in the design so as to maintain constant look up time and avoid memory contention.
- > The Final lookup is a DRAM access that gives the routing, firewall & QoS result.

Memory Structure per stage

> Features

- ◆ The Memory is split into 8 different address spaces based on IP address Prefix.
- ◆ The address spaces are then split based on the levels.
- ◆ The design is pipelined.

Level 0 -7



Classification Engine Memory Structure



2.25 MB for Forwarding Engine for a million entries of IPV6

Performance Results

- > Area is 4.8 sq mm in 0.25um library
 - ◆ ~1 sq.mm in 130 nm
- > Throughput of 28 Million lookups per second
- > The Memory sizes are
 - ◆ 2.25 MB for Forwarding Engine for a million entries (for IPv6)
 - ◆ 90 KB for Diffserv Engine for 20K entries.
 - ◆ 45 KB for Firewall Engine for 10K entries.
- > The update time is 8ms.

Related Work

Design	Configurable	Throughput	Memory	Feature
This Work (IPv4 & IPv6)	Yes	28 Million lookups per second (~8 Gbps)	2.25MB 90KB 45KB	One classification engine for Qos, firewall, forwarding
Baboescu et al. (IPv4)	No	2.5 Gbps	~ 2Mb (1700 rules)	Uses ABV
Singh et al. (IPv4)	No	2 Gbps	235 KB (85 rules)	Uses hypercuts

Conclusions

- > The Classification engine performs the following tasks at 28 Million lookups per second.
 - ◆ Forwarding
 - ◆ Differentiated Services
 - ◆ Firewall
- > The Classification engine is configurable and can handle ranges of data.
 - ◆ Controlled by instructions
 - ◆ Controlled by action code
 - ◆ Memory boundaries are flexible.
- > The update time is 8ms which leads to dynamic updates.
- > The classification is not halted while updating of entries.

Questions?