Speedy FPGA-Based Packet Classifiers with Low On-Chip Memory Requirements

Chih-Hsun Chou, Fong Pong* and Nian-Feng Tzeng

*Broadcom Corporation

Center for Advanced Computer Studies

University of Louisiana, Lafayette, USA

BROADCOM.

Connecting everything®

Highlights

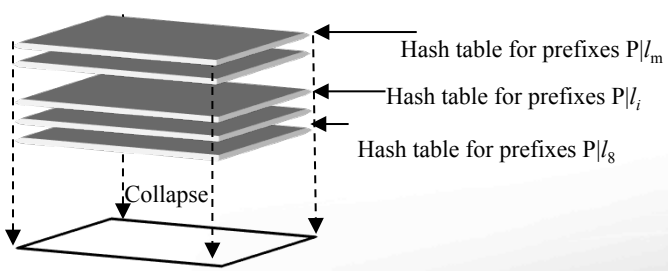
- A simple structured pipeline design
  - Based on the two-stage HaRP (Hash Round-down Prefixes) scheme (Usenix ATC 2009)
    - 1st stage percolates rules by prefix matching on (SIP, DIP)
    - 2nd stage inspects other dimensions (e.g. SP, DP, Protocol)
  - Clock @ 300MHz (Virtex-6)
  - Throughput @ 200 MPPS
- Easy to be replicated and parallelized for higher performance
  - Memory bandwidth bound
  - N instances deliver N times of throughput
- Enhancements

BROADCOM.
Connecting everything®

HaRP

- Prefix matching on (SIP, DIP)
  - Choose Designated Prefix Length (DPL) $\{l_1, l_2, \ldots l_i, \ldots l_m\}$, for example, {32, 28, 24,..}
  - Lookup Key - Round down prefix $P|w$, with $l_i \leq w < l_{i+1}$, to $P|l_i$, *e.g. 29 → 28*
  - Each DPL tread *logically* defines a hash table, but …. Achieve higher storage utilization and reduced overflows by
    - Lump all tables in one *k-way set-associative* table
    - Migrate (SIP, DIP) among buckets, identified by SIP' or DIP' , where SIP' >> SIP and DIP' >> DIP (Prefixes >> are transitive)
  - On lookup for (sip, dip), search all buckets pointed by sip|DPL and dip|DPL
- Simple linear search on ASI (Application-Specific-Information) in the 2nd stage

Total entries = B buckets * k entries per bucket

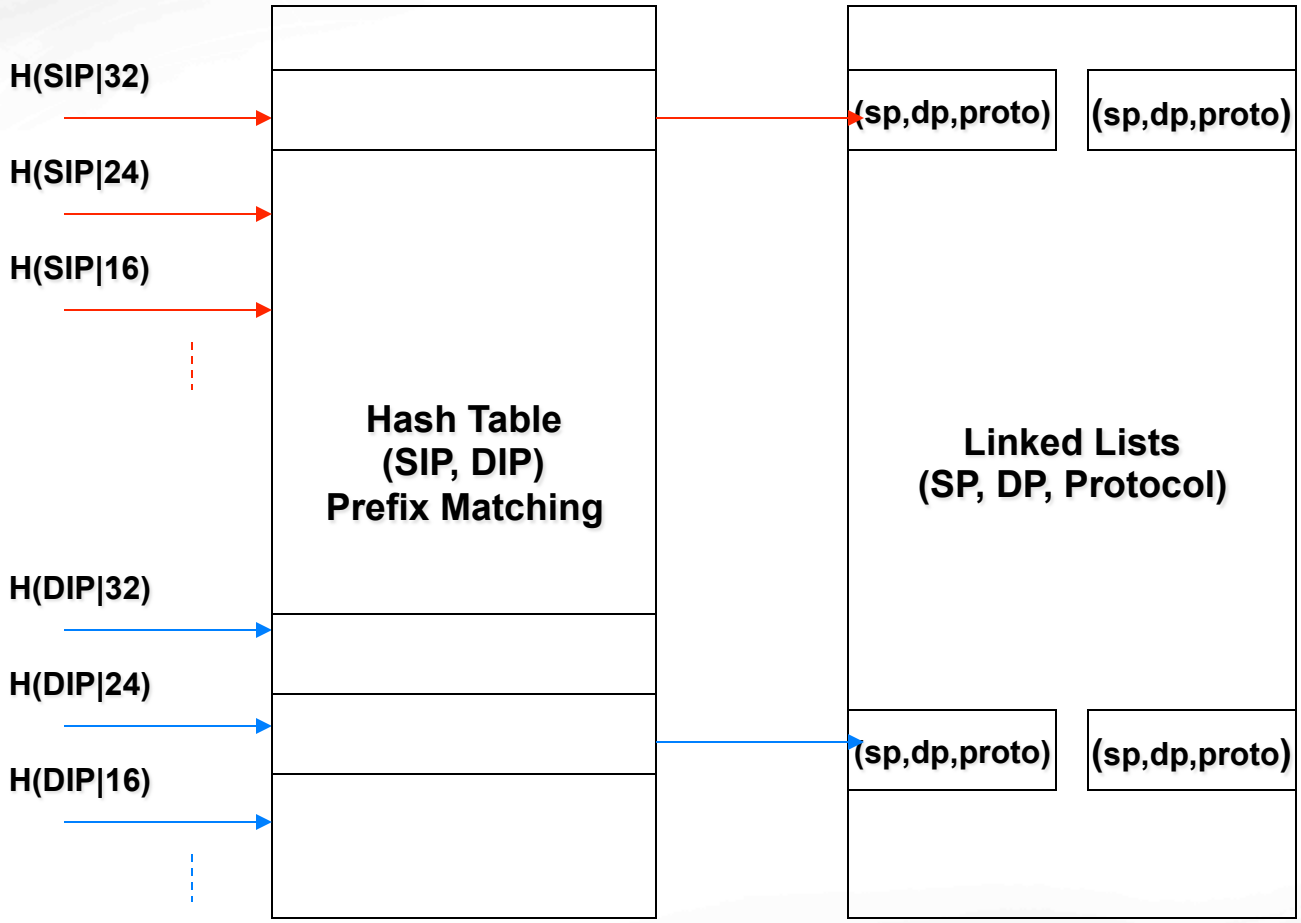


(SIP|n, DIP|m)	ASI
(SIP|n, DIP|m)	ASI
::	::
(SIP|n, DIP|m)	ASI

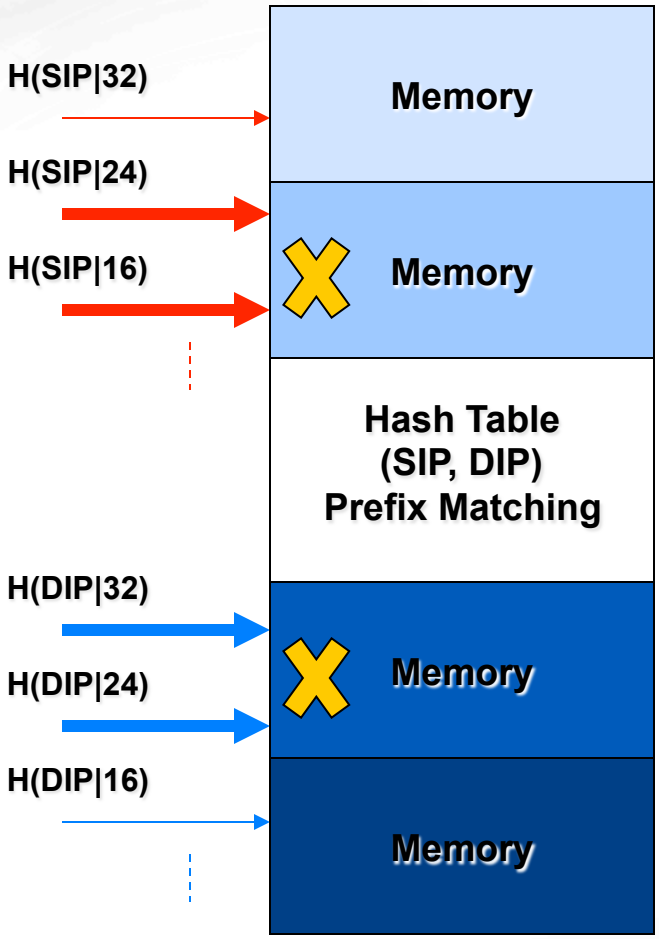
→ (sp, dp, pr) (sp, dp, pr) (sp, dp, pr) (sp, dp, pr) → (sp, dp, pr) (sp, dp, pr)

BROADCOM. Connecting everything®

Illustration of Lookup



Memory Collisions



- Memory collisions serialize probes and degrade lookup performance
- Use more memory modules (physical design constraints)
- Reduce number of probes (less hash migration opportunity)
- Exploit parallelisms by handling N packets in parallel

BROADCOM.
Connecting everything®

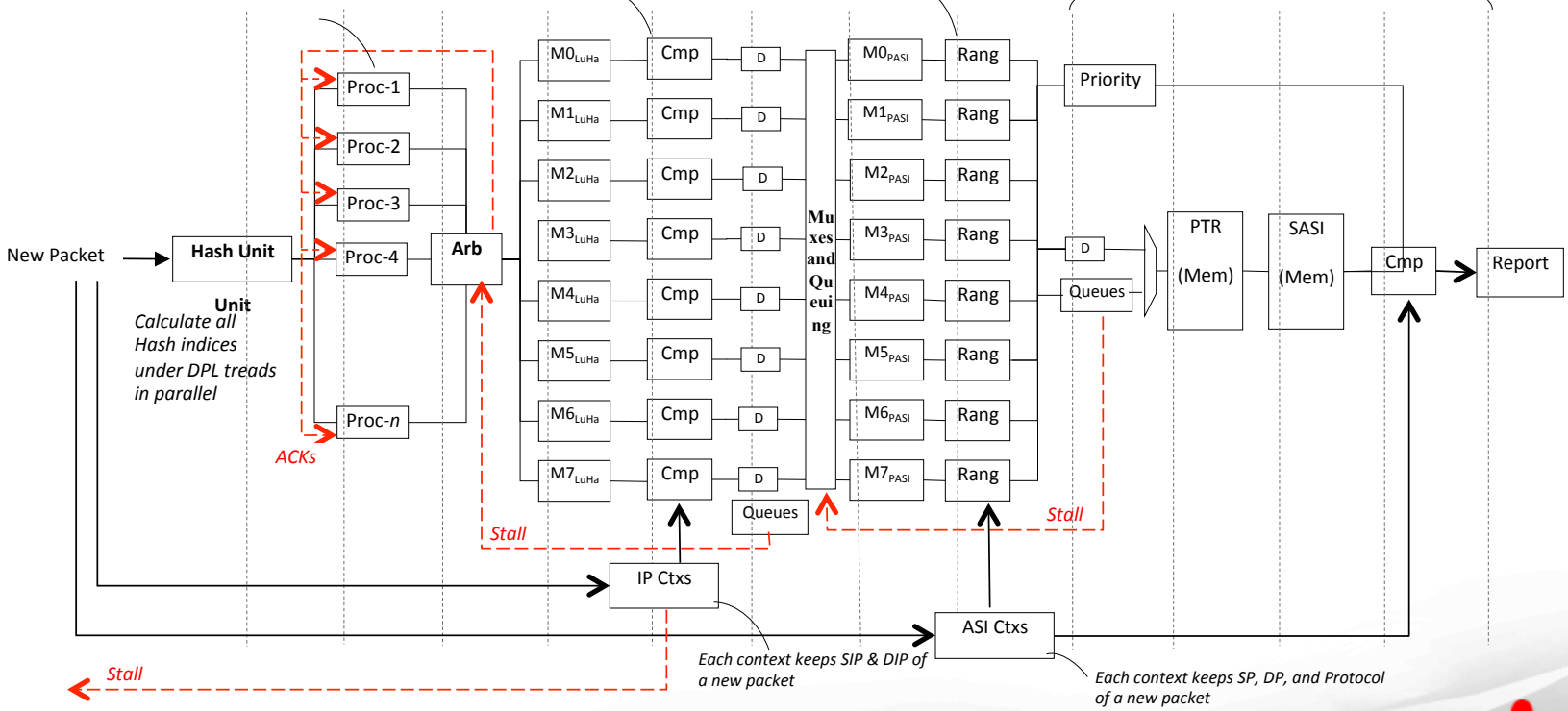
Pipeline (Virtex-6)

Each Proc handles one packet and calculates collision-free requests. The arbiter (Arb) receives n sets of conflict-free requests from those n Proc's. Arb schedules as many collision-free memory requests as possible in each cycle.

Prefix Matching: Range for Port Numbers and Protocol Type.

At most 8 (= 8 shown) LuHa accesses to packets are handled by 8 memory banks per cycle.

The secondary ASI table is read when an accessed ASI list has more than one element. correspondence between LuHa and PASI.



BROADCOM.
Connecting everything®

Consumed Memory Resources

Rule dataset	# of rules	Taken on-chip memory (Kbytes)	Usage ratio
ACL	28240	655.2	35.0 %
FW	28473	697.6	37.3 %
IPC	29876	776.8	41.5 %

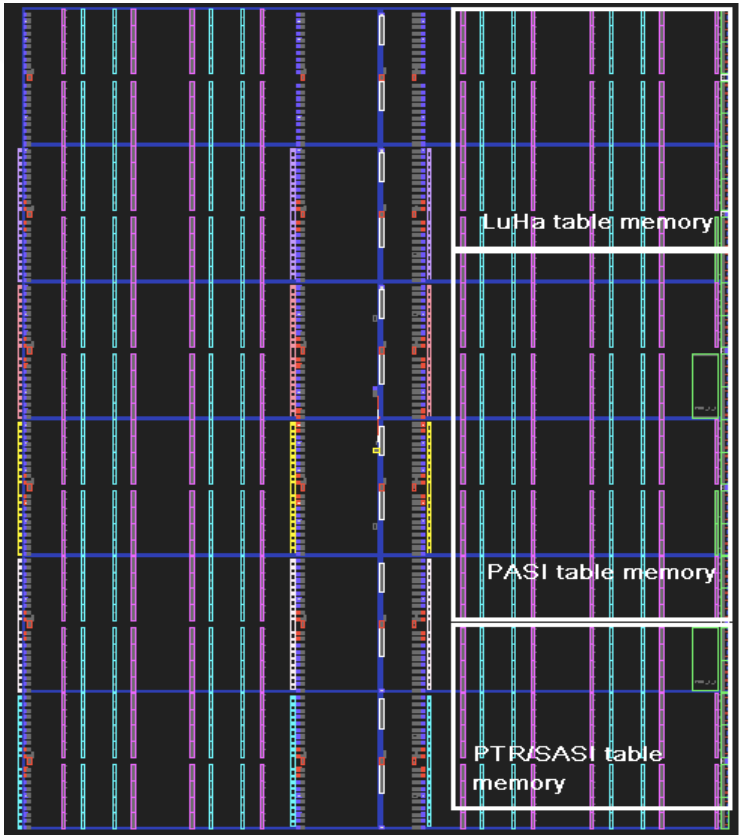
BROADCOM.
Connecting everything®

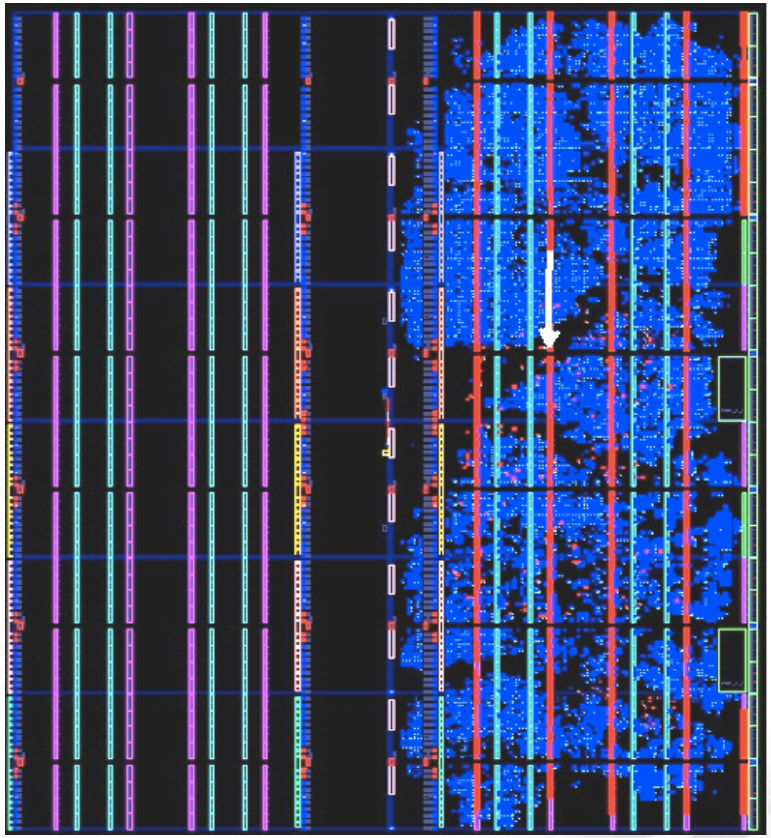
Consumed Hardware Logic

(no. of proc's, memory banks)	Consumed hardware breakdowns			
	Slice registers	Slice LUTs	Occupied slices	
(1, 1)	4,313 (1 %)	5,124 (3 %)	1,763 (4 %)	
(2, 2)	6,995 (2 %)	7,597 (5 %)	2,574 (6 %)	
(2, 4)	9,604 (3 %)	10,954 (7 %)	3,548 (9 %)	
(4, 4)	12,414 (4 %)	13,799 (9 %)	4,465 (11 %)	
(4, 8)	18,572 (6 %)	21,699 (14 %)	6,706 (17 %)	
(8, 8)	**25,944 (8 %)**	**31,048 (20 %)**	**9,374 (24 %)**	**300MHz, 200MPPS**
(8, 16)	41,923 (13 %)	49,048 (32 %)	14,452 (38 %)	slower clock at 220MHz
(16, 16)	63,856 (21 %)	85,253 (56 %)	24,888 (66 %)	even lower clock

BROADCOM.
Connecting everything®

Structured Layout





BROADCOM.
Connecting everything®

Enhancements

- Reduce number of hash probes
- Guided dynamically in exploiting hash buckets for migration
- For example of a DPL {32, 28, 24, ..}
  - Hash key P|24 is valid only if the LSB of P|24 is 1
  - Reduce the opportunity for hash (bucket) migration when installing rules
  - Reduce one hash probe during lookup
- 200MPPS → 250MPPS

BROADCOM.
Connecting everything®

Conclusion

- A simple method leads to a simple structured design
- It is well suited for parallel and pipeline hardware implementation

BROADCOM.
Connecting everything®

Q&A

Thank You!

BROADCOM.
Connecting everything®