Scaling IP Lookup to Large Databases using the CRAM Lens

Robert Chang, Pradeep Dogga, Andy Fingerhut,
Victor Rios, George Varghese

Takeaways

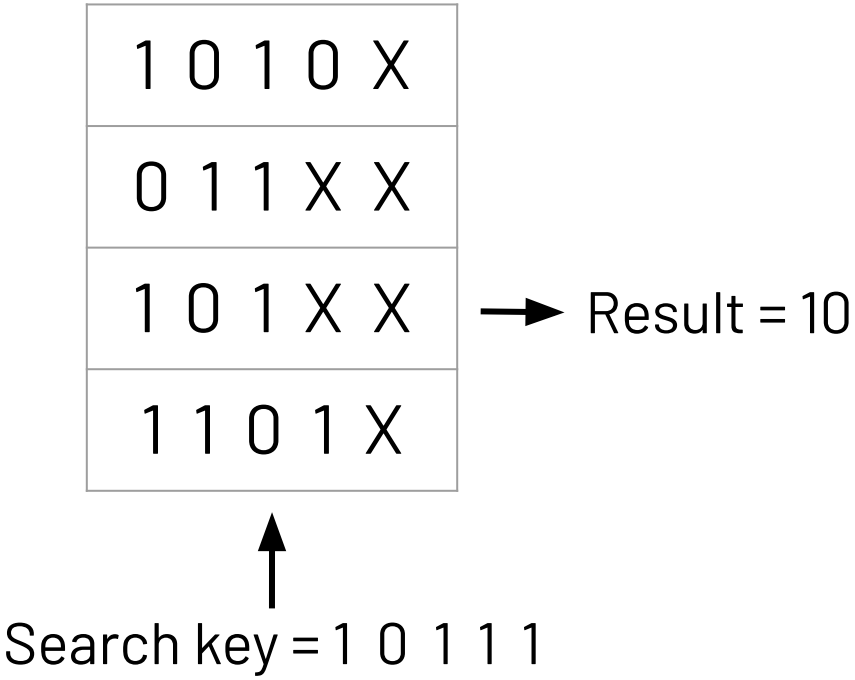
Algorithmic	Using *both* CAM and RAM synergistically introduces a new research direction for scaling network processing algorithms.
Complexity model	Analogous to the RAM model, our new CRAM model provides a simple predictive model of performance on chips like Intel's Tofino-2 with a set of accompanying design idioms.
IP lookup	Using CRAM, we scale IPv4 and IPv6 lookup table sizes by 9X and 3.2X, respectively, compared to pure TCAM approaches.
Architectural	A *little* TCAM goes a long way; network chip designers should include both TCAM and SRAM in their chips.

Background

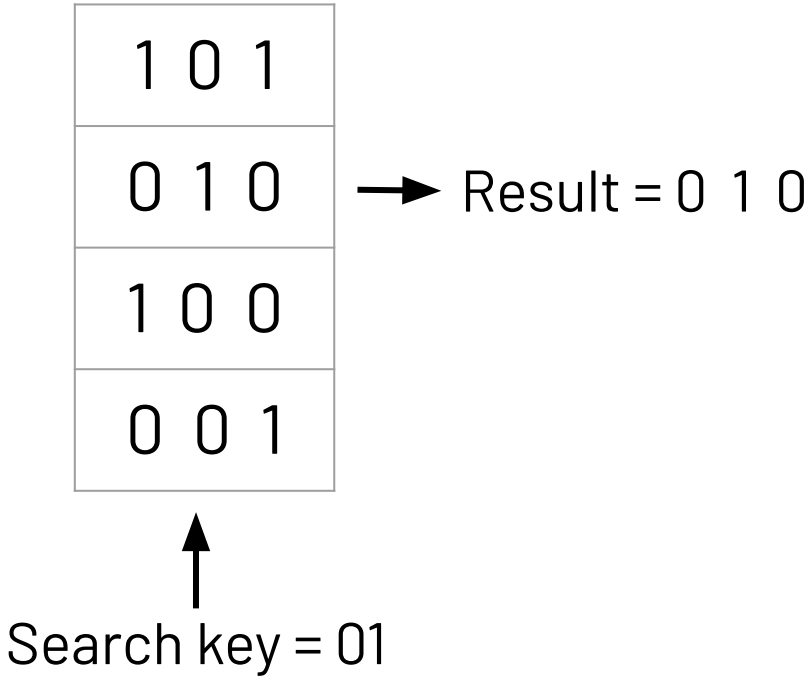
UCLA

TCAM vs SRAM

Ternary content-addressable memory (TCAM)

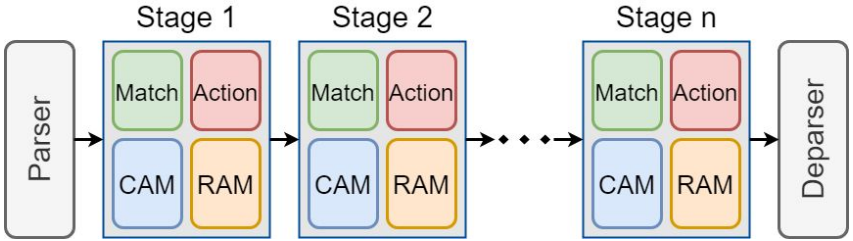


Static random-access memory (SRAM)

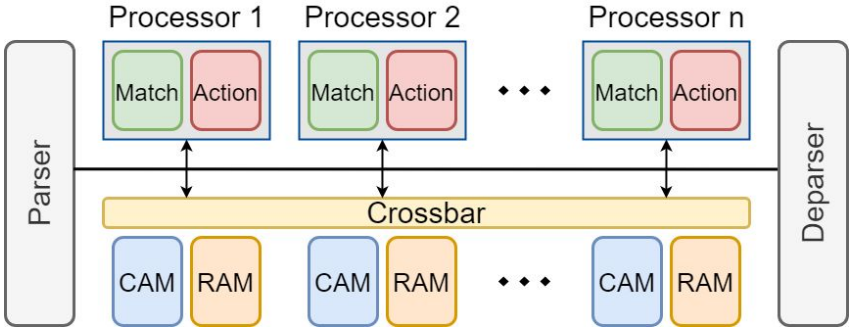


Packet processing architectures: RMT & dRMT

Reconfigurable Match-Action Tables (RMT)



Disaggregated RMT (dRMT)



Availability of TCAM *and* SRAM

Parallelism

Programmability

Motivation

Lack of complexity model

Lack of an abstract model for *quickly* evaluating and comparing algorithms on RMT and dRMT chips.

- Parallel lookups per stage
- TCAM and SRAM dimensions
- Stages per pipeline
- TCAM and SRAM per stage
- Sequential ops. per stage
- Metadata storage
- Arithmetic ops. per stage
- Bits reserved for actions

Single-resource limitations

SRAM/DRAM only	• Do not translate well to RMT • Optimized strictly for the RAM model
TCAM only	• Consume significantly more power • Scale poorly

How can we leverage *both CAM and RAM* to revisit all of network algorithmics?

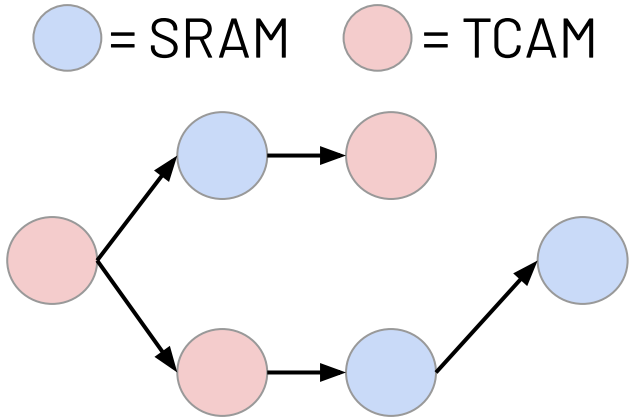
The CRAM Lens

The CRAM model

Formal model for estimating performance on RMT and dRMT chips.

Algorithm

Optimization Idioms



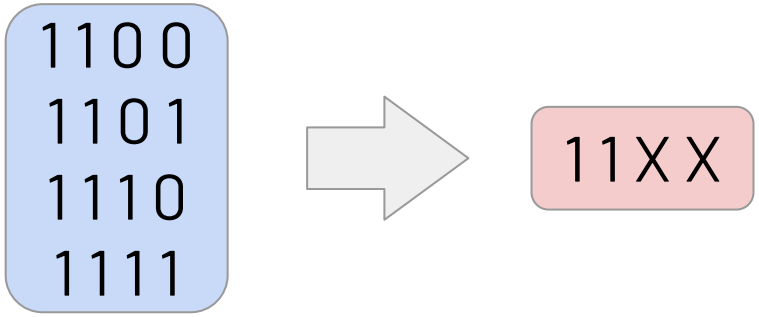
Memory metric: sum *TCAM and SRAM bits* across all tables.

Latency metric: number of *steps* in the longest directed path.

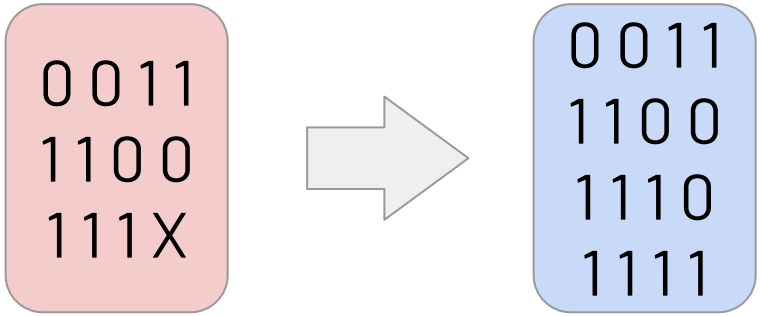
Optimization idioms

= SRAM = TCAM

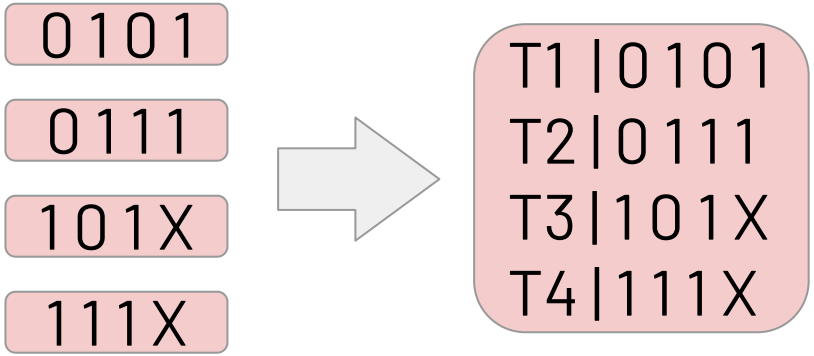
01 Compress with TCAM



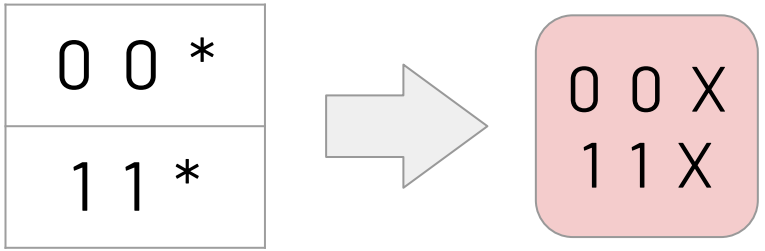
02 Expand to SRAM



03 Table coalescing



04 Look-aside TCAM



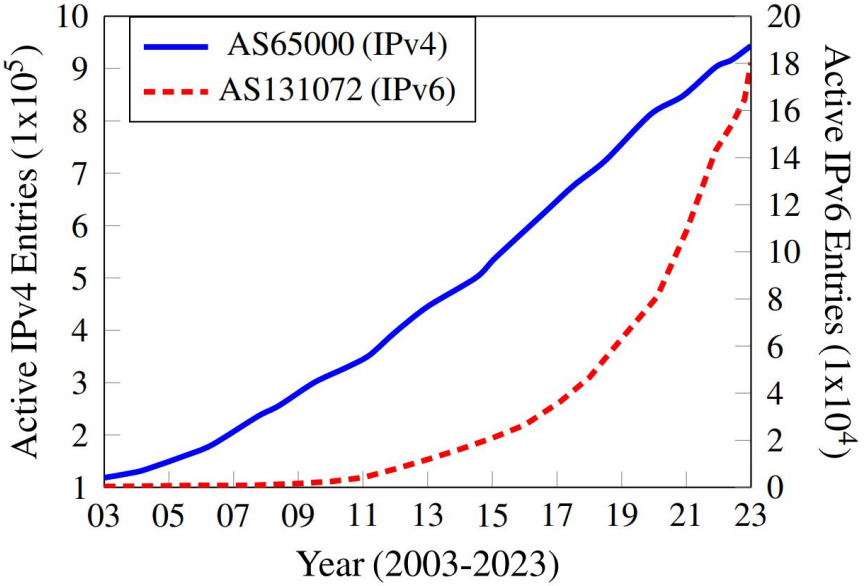
Applying CRAM to IP lookup

UCLA

Why scaling IP lookup matters

Key observations:

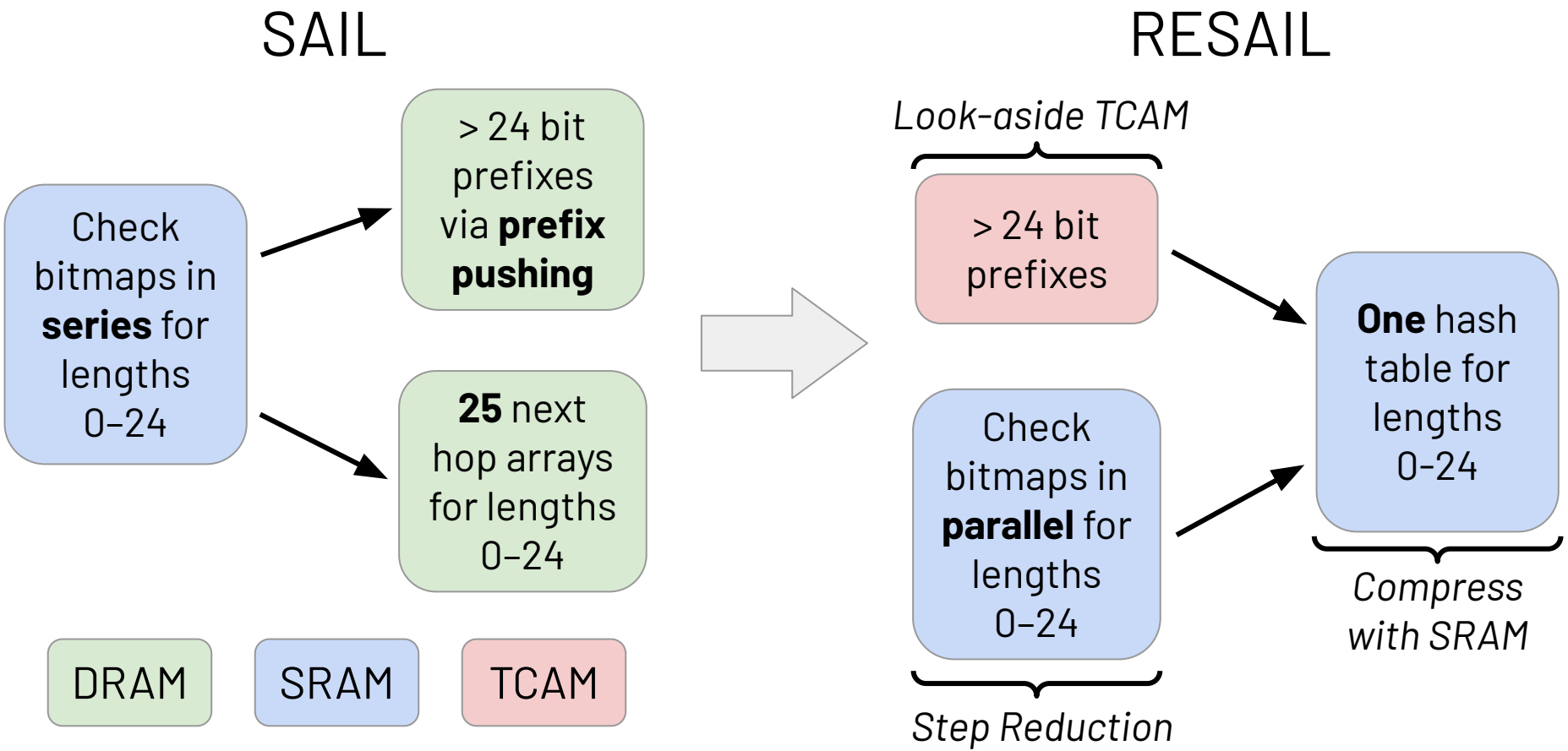
1. Continued IPv4 growth
2. Rapid IPv6 deployment



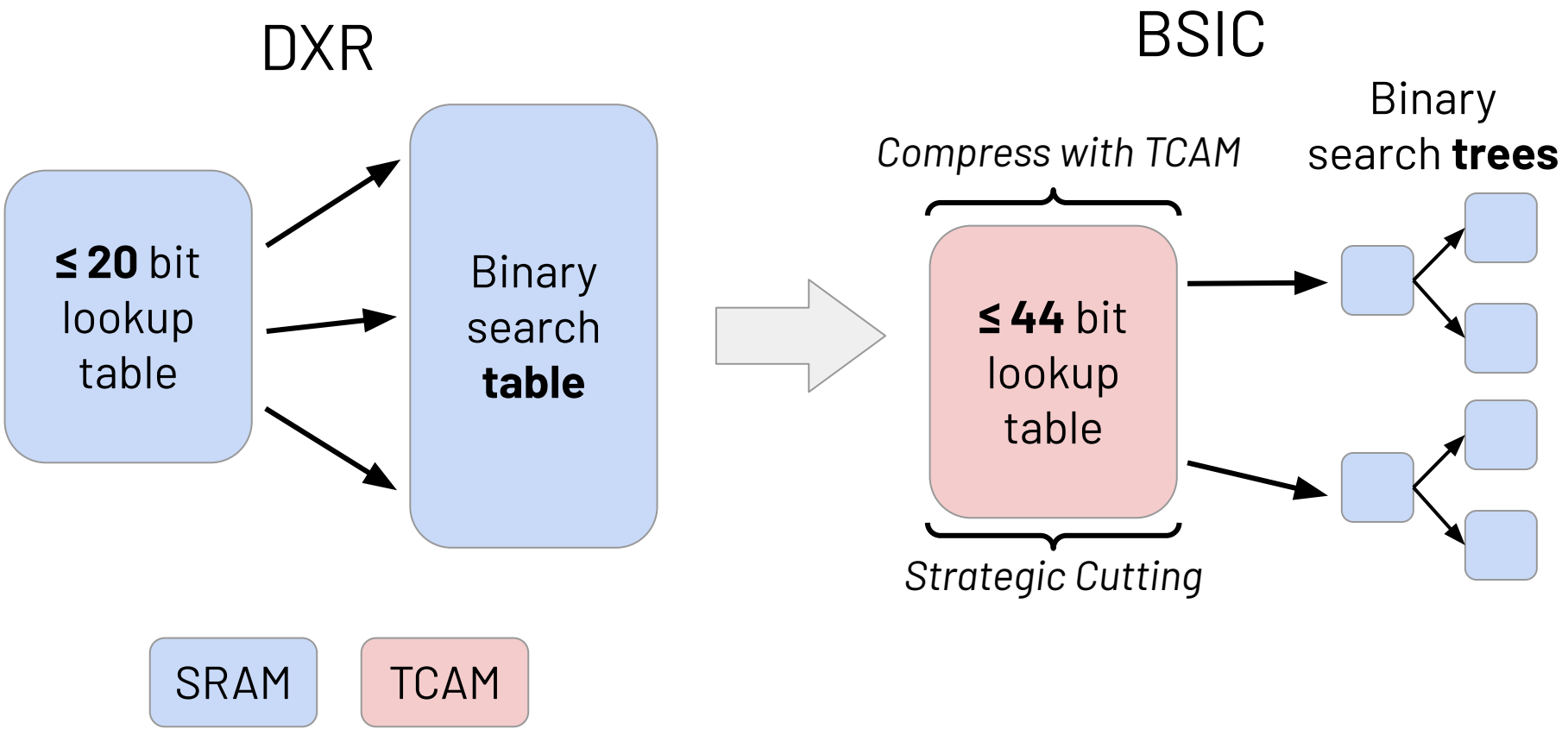
BGP routing table size over the past two decades

Takeaway: From trends, IPv4 table→2M entries by 2033.

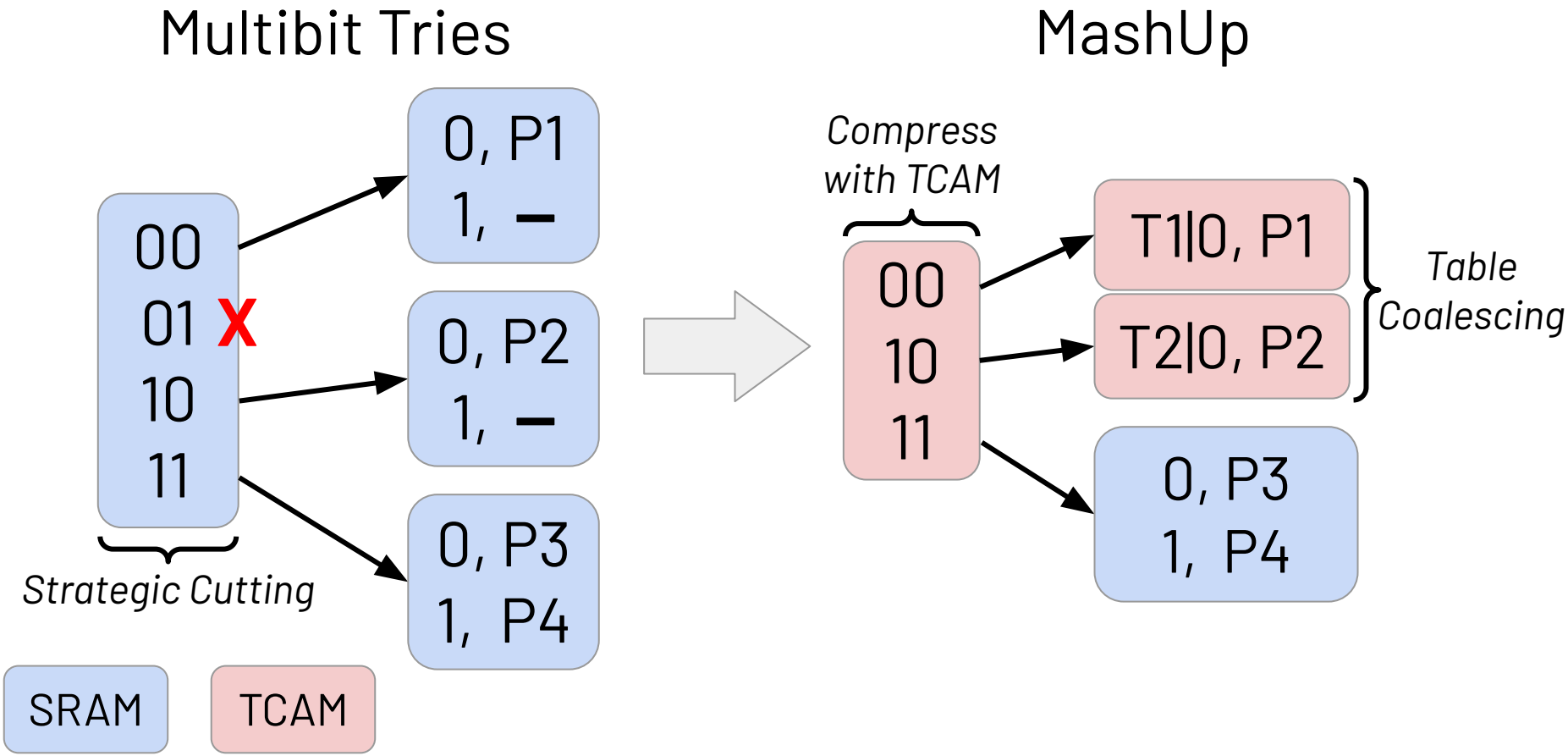
Applying CRAM to SAIL to get RESAIL



Applying CRAM to DXR to get BSIC



Applying CRAM to Multibit Tries to get MashUp



Results for IP lookup

UCLA

Methodology

For our experiments, we used 3 abstraction models:

CRAM model	• Quick, high-level estimates • ***Zero*** knowledge of data sheet required
Ideal RMT model	• TCAM blocks, SRAM pages, pipeline stages • ***Basic*** knowledge of data sheet required
Tofino-2 implementation	• Accounts for all chip-specific details • ***Expert*** knowledge of data sheet required

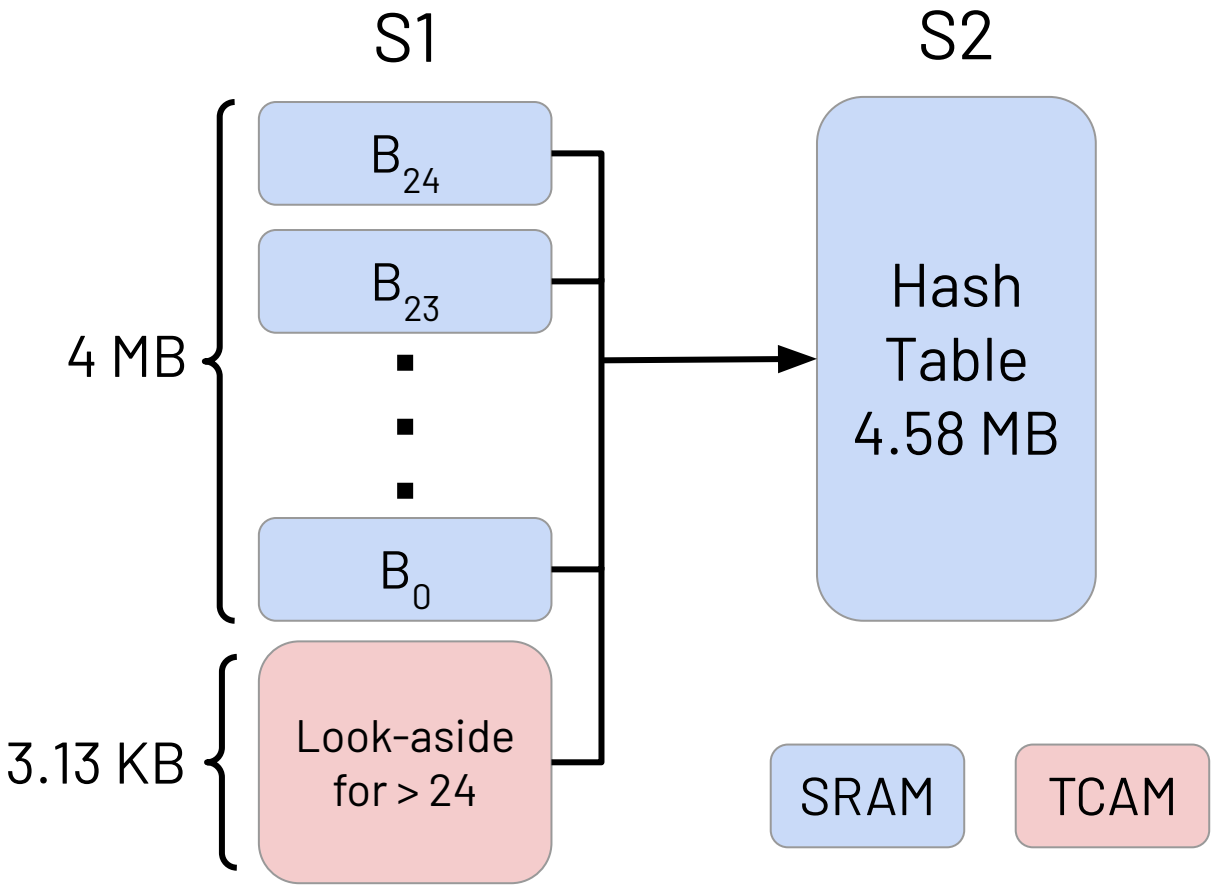
Example: CRAM model for RESAIL on AS65000

RESAIL (IPv4)

TCAM: 3.13 KB

SRAM: 8.58 MB

Steps: 2



What improvements does the CRAM model predict?

CRAM metrics for *AS65000* (IPv4)

Algorithm	TCAM Bits	SRAM Bits	DRAM Bits	Steps
SAIL	–	4 MB	32 MB	26
RESAIL	3.13 KB	8.58 MB	–	2
Multibit Tries	–	12.04 MB	–	4
MashUp	0.31 MB	5.92 MB	–	4

Takeaways: a) Scalability using *both* CAM and RAM.
b) A *little* TCAM goes a long way.

How accurate was the CRAM Model?

RESAIL: CRAM predicts *2 steps*, RMT predicts *9 stages* because of stage constraints, but actual Tofino-2 requires *16 stages* because it only allows for 50% SRAM utilization.

BSIC: CRAM and RMT predict *14 steps*, but actual Tofino-2 uses *30 stages* because 3-way branching requires 2 stages.

Takeaway: Still within a factor of 2 as in Big O estimates.

How scalable are our algorithms?

RESAIL: Scales to *~2.25 million* IPv4 prefixes on Tofino-2, *9X* what pure TCAM can support. SAIL runs out of SRAM.

BSIC: Scales to *~390k* IPv6 prefixes on Tofino-2, *3.2X* what pure TCAM can support. Hi-BST runs out of stages.

Can scale *even further* on an ideal RMT chip without Tofino-2 hardware limitations.

Takeaway: Can support table growth for a decade.

Conclusions

UCLA

Future directions for CRAM research

Hardware Architectures

- Programmable switch ASICs
- SmartNICs
- FPGAs
- Custom fixed-function ASICs

Network Applications

- Packet classification
- Measurement algorithms
- Regular expression matching
- In-network machine learning

Takeaways

Algorithmic: Using both CAM and RAM synergistically can help scale network processing algorithms.

Complexity model: We introduce the CRAM model for predicting performance on chips like Intel's Tofino-2.

IP lookup: We scale IPv4 and IPv6 lookup table sizes by 9X and 3.2X, respectively, compared to pure TCAM approaches.

Architectural: A little TCAM goes a long way; network chip designers should include both TCAM and SRAM in their chips.

Thank You

UCLA