A LOW-POWER CONTENT-ADDRESSABLE MEMORY BASED ON CLUSTERED-SPARSE NETWORKS

McGill University

ASAP 2013

Washington D.C.

Hooman Jarollahi – Ph.D. Candidate

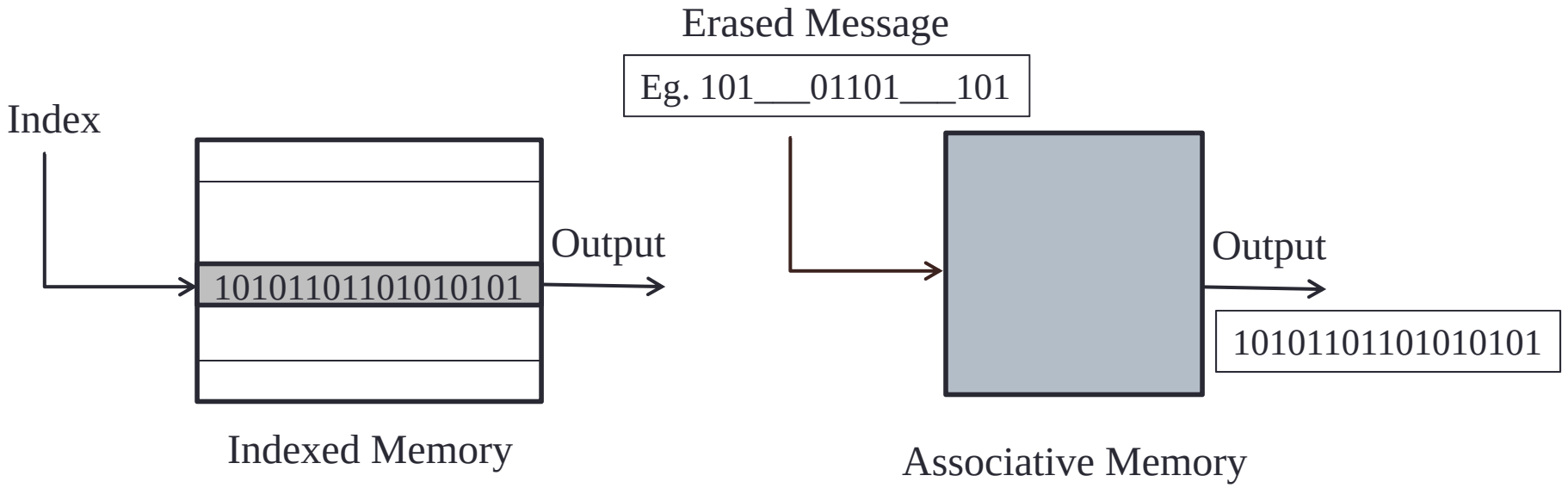
Vincent Gripon, Naoya Onizawa, Warren J. Gross

Outline

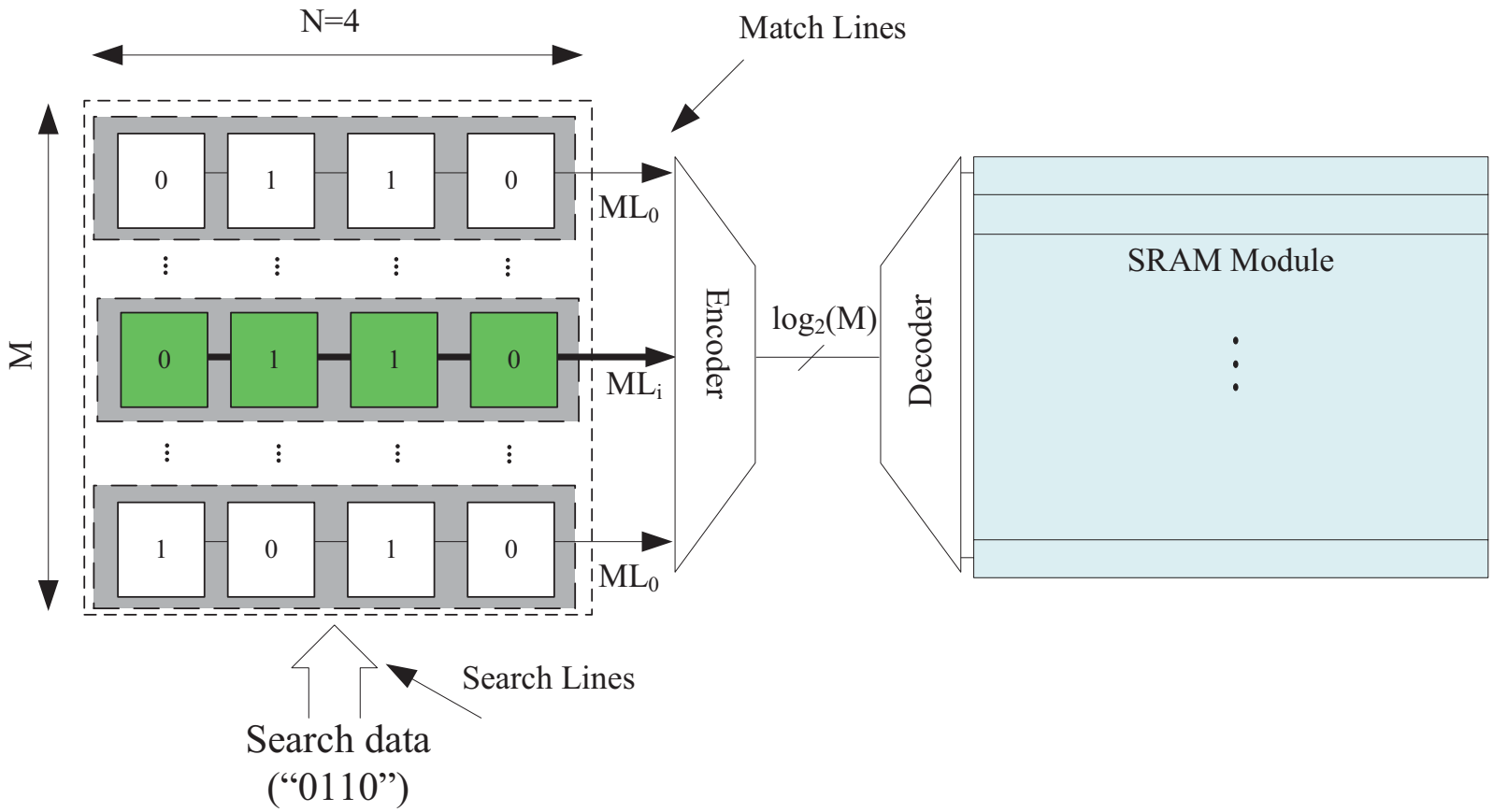
- Motivation and Background
- Clustered Neural Networks (CNN)
- CNN-based Content-Addressable Memory (CAM)
- Comparison with previous works
- Conclusions

Associative Memories

- Associative Memories: alternatives to indexed memories
- Contents are linked, links are stored
- No need to input an explicit address
- Part of the content of a message is used to retrieve the full message
- Applications: search engines, data mining, set implementation



Conventional CAM

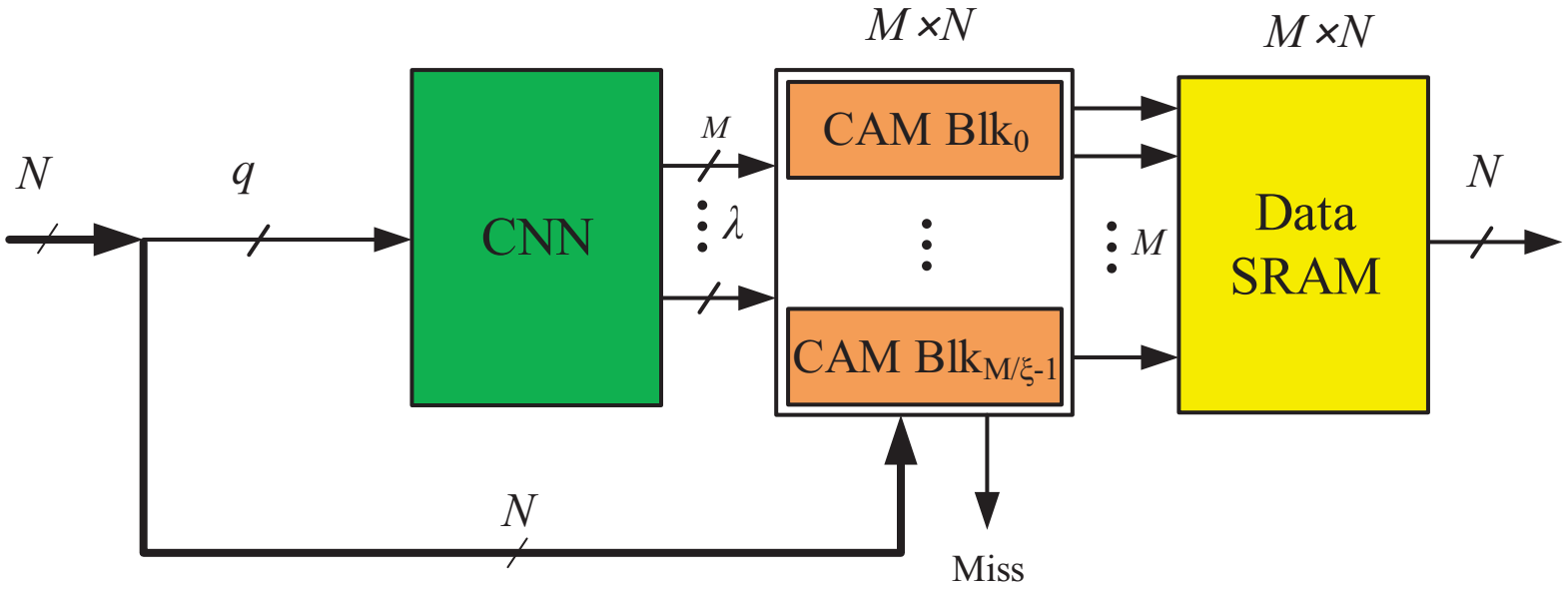


- Two field AM: Tag and data
- Fast parallel look-up operations
- High-power consumption due to parallel search.
- Cache Memory, Translation look-aside buffers (TLBs), network routers

Power reduction techniques in CAMs

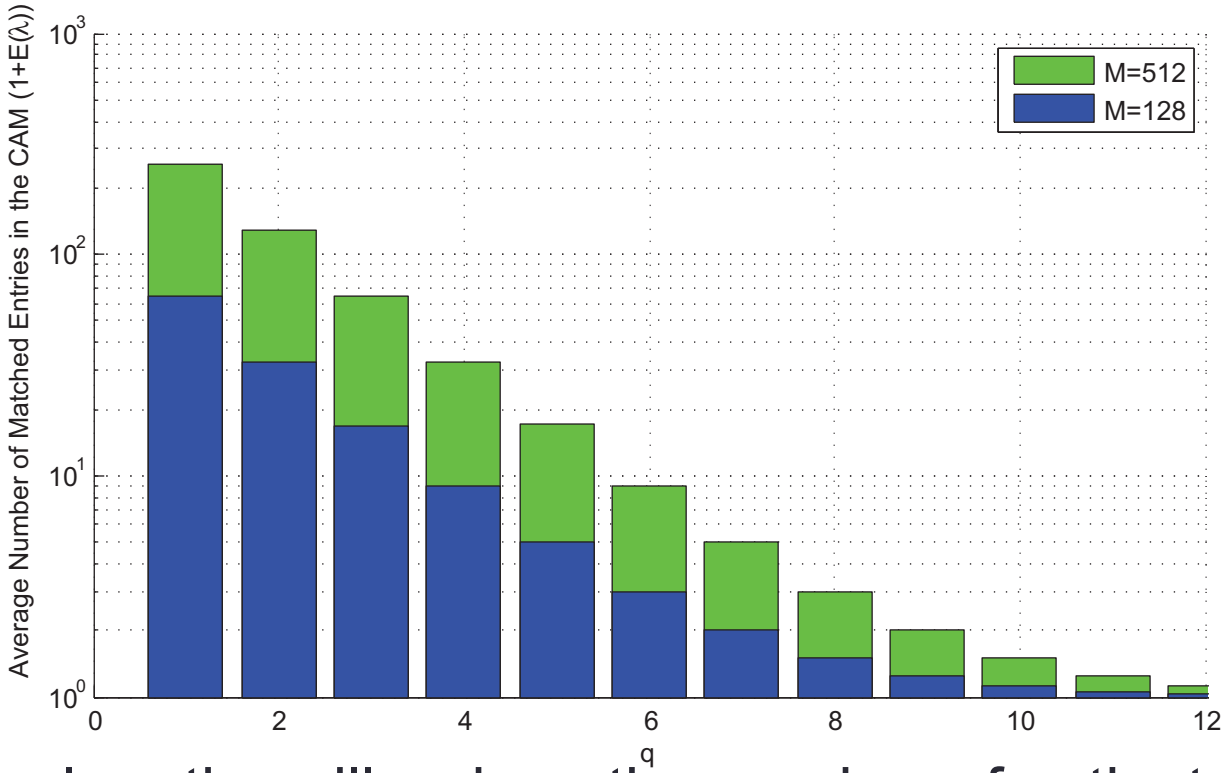
- Circuit-level techniques suffer low noise margins, charge sharing, analog components hence mismatch, etc.
  - Match-line sensing
    - Conventional Match-line pre-charge schemes: NOR-type and NAND-type cells
    - Low-swing schemes: external power supplies
    - Hybrid NAND-NOR, selective pre-charge, current-race, pipelining.
  - Search-line sensing schemes
    - Conventional pre-charge, Hierarchical, pre-charge elimination
- Architectural-level
  - Bank-selection: bank overflow
  - Pre-computation method: delay and complexity is increased as the tag length is increased
- Hybrid methods
  - Asynchronous CAM

CNN-CAM overview



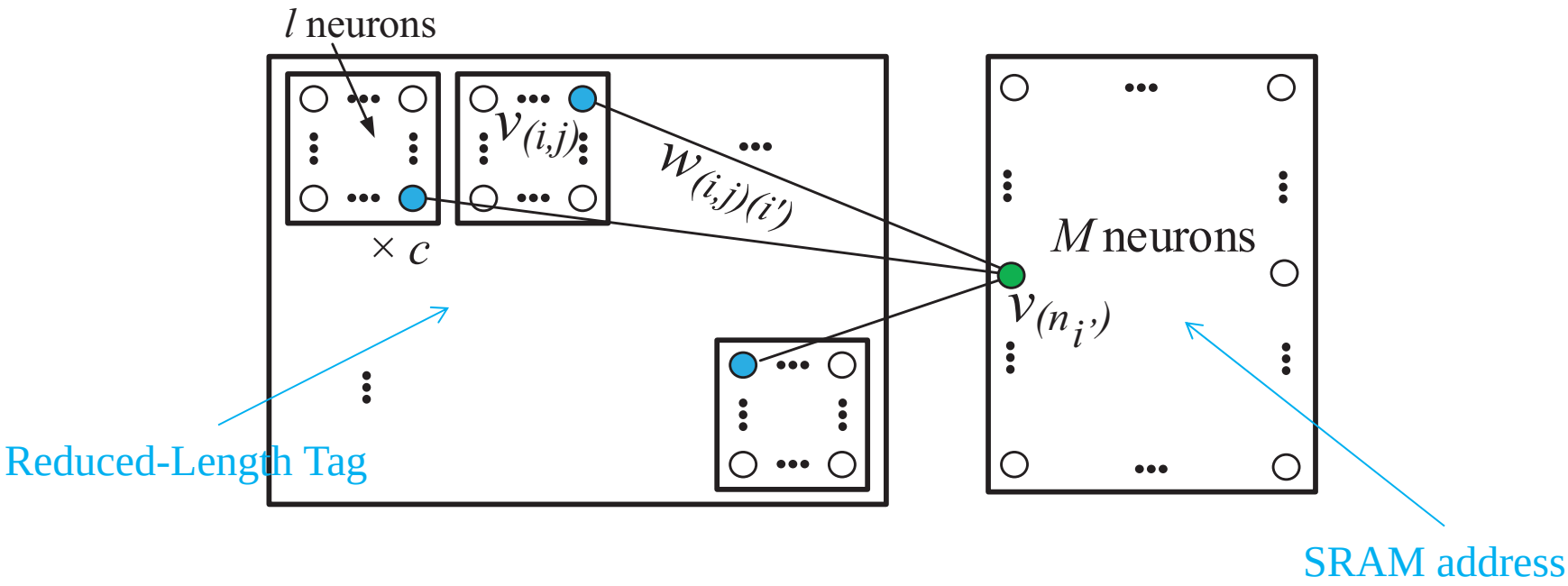
- High-performance NOR-type binary CAM-block is divided into multiple sub-blocks
- CNN is trained before the look-up
- Input is presented to CNN to predict the sub-block to be activated
- 2 sub-blocks are activated in average
- The entries in the activated sub-blocks are compared with original tag to find the match.

Reduced-length Tag vs. Number of Matched Entries



- Larger lengths will reduce the number of activated blocks
- Also increase the network complexity

Clustered Neural Networks



- Recently introduced classification algorithm, recently implemented*
- A network consisting of two parts: Tag and SRAM address
- Input tag is reduced in length into *q* bits
- Divided into *c* sub-tags
- mapped into a neuron per cluster and linked to neuron in the second part

*Jarollahi, H., Onizawa, N., Gripon, V., and Gross, W. J., "Architecture and implementation of an associative memory using sparse clustered networks," *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seoul, Korea, 20-23 May 2012, pp. 2901-2904

Network Parameters and Results

	PF-CDPD [12]	Hybrid [13]	STOS [3]	HS-WA [1]	Ref. NAND	Ref. NOR	Proposed
Configuration	256×128	128×32	256×144	128×128	512×128	512×128	512×128
CAM type	BCAM	BCAM	BCAM	BCAM	BCAM	BCAM	BCAM
Cell type	NAND	NAND-NOR	NAND	NAND-NOR	NAND	NOR	NOR
Technology	0.18 μm	0.13 μm	90 nm	32 nm	0.13 μm	0.13 μm	0.13 μm
Delay [ns]	2.10	0.60	0.26	0.145	2.30	0.55	0.70
Energy metric [fJ/bit/search]	2.33	1.3	0.162	1.07	1.30	2.39	0.124

- System Parameters
  - *512×128* CAM
  - 9 bits reduced-length tag
  - 3 clusters
  - 8 neurons per cluster number of neurons per cluster
- Also Implemented Conv. NAND, NOR for comparison (SPECTRE, 0.13 µm CMOS)
- Proposed design:
  - Energy: 9.5% of conv. NAND
  - Delay: 30.4% of conv. NAND
  - 3.4% higher number of transistors

	Parameter	Value
CNN	M	512
	N	128
	ζ	8
	β	64
	$E(\lambda)$	1
	q	9
	c	3
	l	8
CAM	CAM type	XOR
	ML Arch.	NOR
	Supply Voltage	1.2V
	Technology	0.13 μm

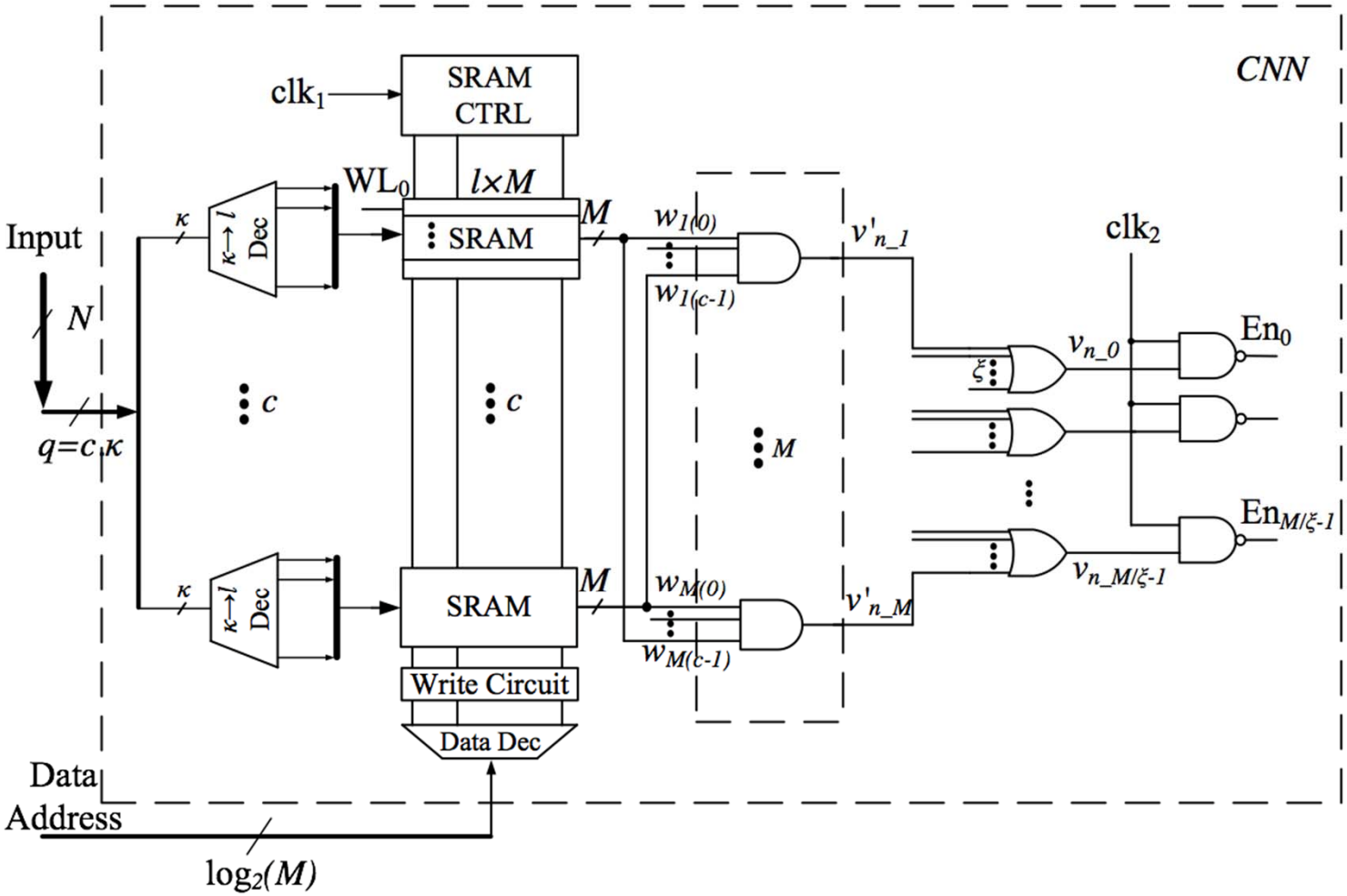
Conclusions

- Content Addressable Memories:
  - Cache memories, TLBs, Network routers
- Previous implementation of GBNN
  - Many circuit level techniques few architectural research.
- New architecture and implementation of CNN-CAM
- Comparison with conventional and other research results

Thank you!

Q/A

CNN Schematic



GBNN: Learning

- A network consists of
  - *c* clusters
  - *l* neurons per cluster
  - *n* neurons in total
- A message consists of
  - *K* bits
  - K/c -bit messages
  - $\kappa = K/c = log_2(l)$
- A learned message is represented as a:
  - **Clique** : activated neurons with fully-interconnected links