Highly-Associative Caches for Low-Power Processors

Michael Zhang

Krste Asanovic

{rzhang|krste}@lcs.mit.edu

MIT Laboratory for Computer Science

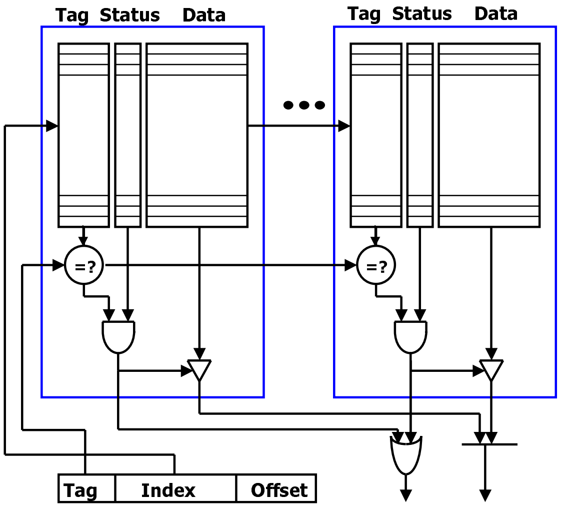
Motivation

- **Cache uses *30-60%* processor energy in embedded systems.**
  - □ ***Example: 43% for StrongArm-1***
- **Many academic studies on cache**
  - ● ***[Albera, Bahar, ’98] – Power and performance trade-offs***
  - ● ***[Amrutur, Horowitz, ‘98,’00] – Speed and power scaling***
  - ● ***[Bellas, Hajj, Polychronopoulos, ’99] – Dynamic cache management***
  - ● ***[Ghose, Kamble,’99] – Power reduction through sub-banking, etc.***
  - ● ***[Inoue, Ishihara, Murakami,’99] – Way predicting set-associative cache***
  - ● ***[Kin,Gupta, Mangione-Smith, ’97] – Filter cache***
  - ● ***[Ko, Balsara, Nanda, ’98] – Multilevel caches for RISC and CISC***
  - ● ***[Wilton, Jouppi, ’94] – CACTI cache model***
- **Many Industrial Low-Power Processors use CAM (*content-addressable-memory*)**
  - □ ***ARM3 – 64*-way set-associative – *[Furber et. al. ’89]***
  - □ ***StrongArm – 32*-way set-associative – *[Santhanam et. al. ’98]***
  - □ **Intel *XScale – 32*-way set-associative – *’01***
- ***CAM*: *Fast* and *Energy-Efficient***

Talk Outline

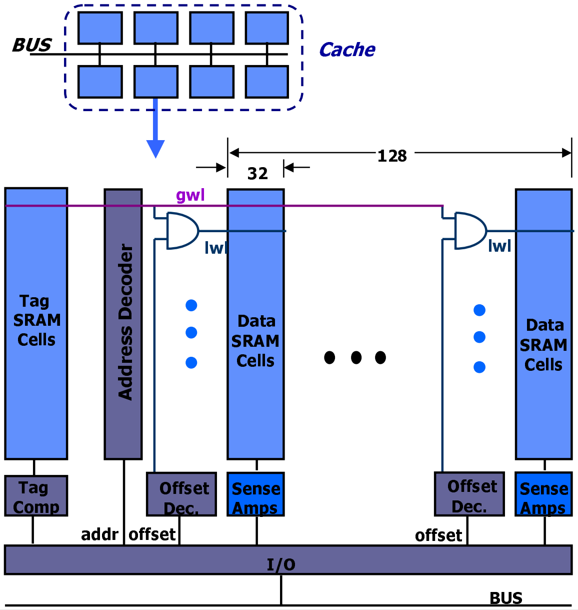
- Structural Comparison
- Area and Delay Comparison
- Energy Comparison
- Related work
- Conclusion

Set-Associative RAM-tag Cache



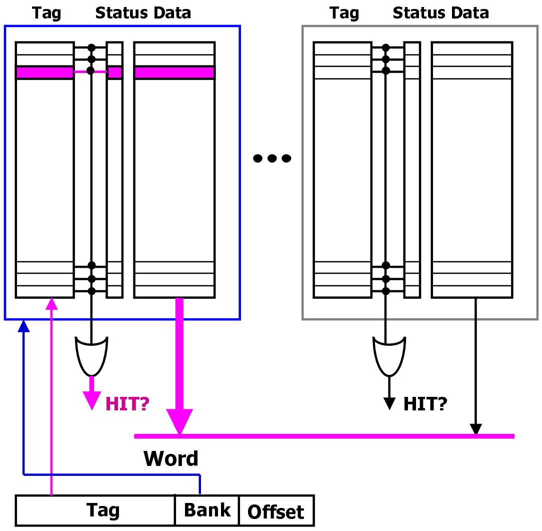
- Not energy-efficient
  - □ All ways are read out
- Two-phase approach
  - □ More energy-efficient
  - □ 2X latency

Set-Associative RAM-tag Sub-bank



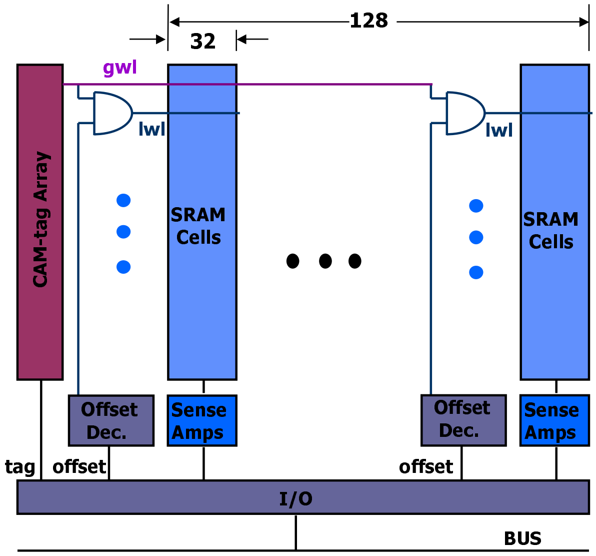
- Not energy-efficient
  - □ All ways are read out
- Two-phase approach
  - □ More energy-efficient
  - □ 2X latency
- Sub-banking
- *1* sub-bank = *1* way
- Low-swing Bitlines
  - □ Only for reads, writes performed full-swing
- Wordline Gating

CAM-tag Cache



- Only one sub-bank activated
- Associativity within sub-bank
- Easy to implement high associativity

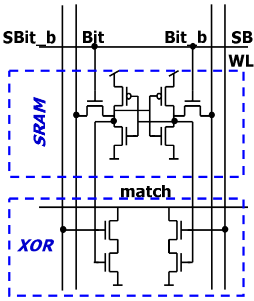
CAM-tag Cache Sub-bank



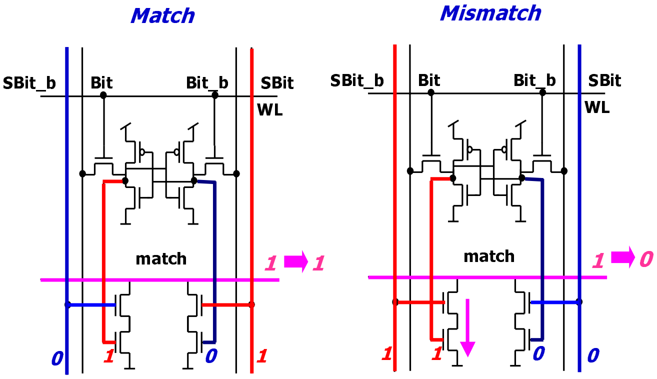
- Only one sub-bank activated
- Associativity within sub-bank
- Easy to implement high associativity

CAM Functionality and Energy Usage

- *CAM Energy Dissipation*
  - □ Search Lines
  - □ Match Lines
  - □ Drivers

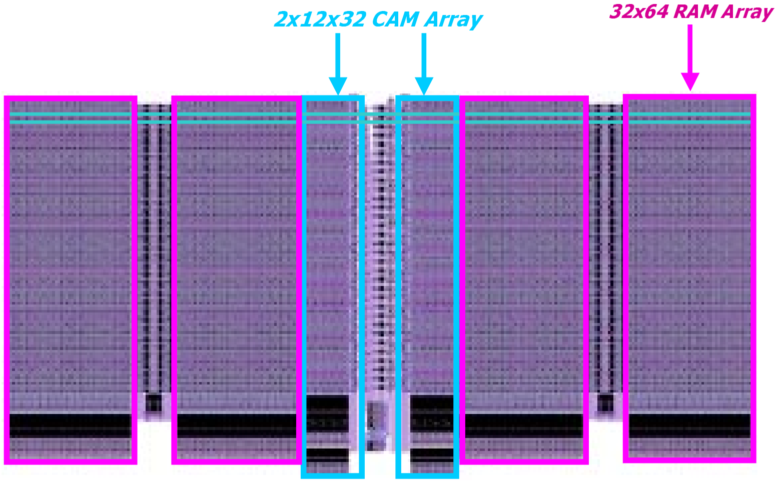


10-T CAM Cell With Separate Write/Search Lines And Low-Swing Match Line



CAM-tag Cache Sub-bank Layout

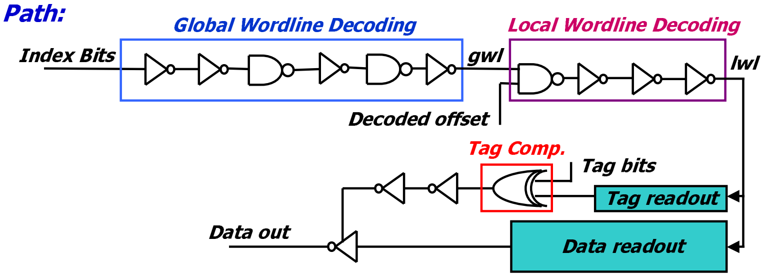
1-KB Cache Sub-bank implemented in 0.25 μm CMOS technology



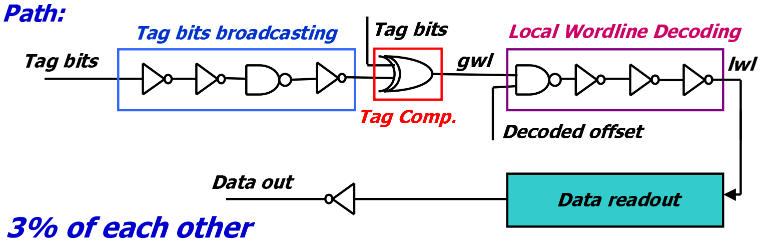
- *10% area overhead over RAM-tag cache*

Delay Comparison

RAM tag Cache Critical Path:

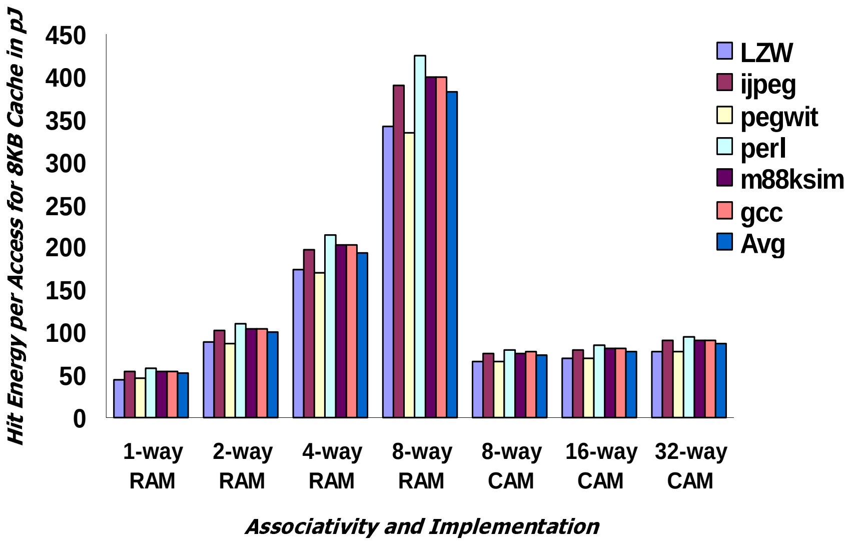


CAM tag Cache Critical Path:

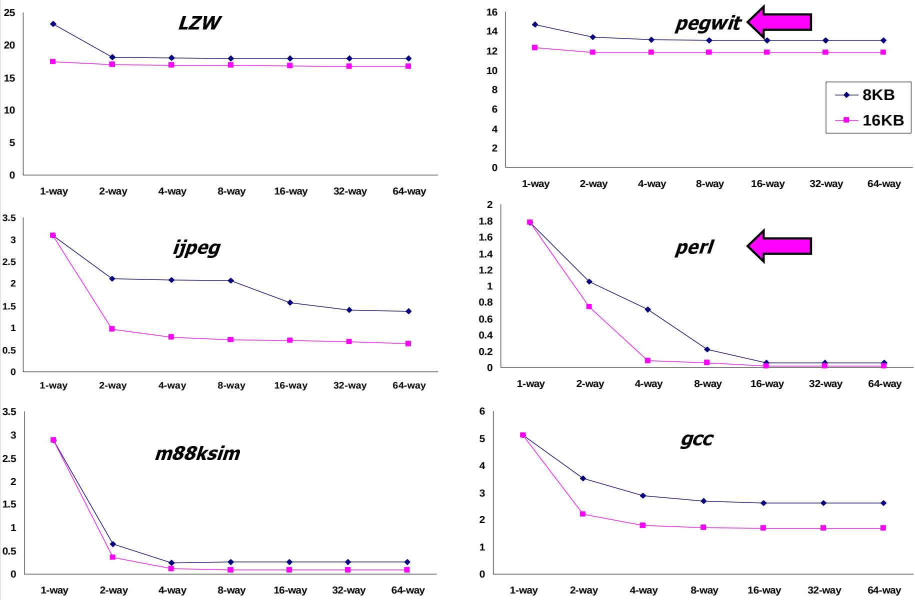


Within 3% of each other

Hit Energy Comparison

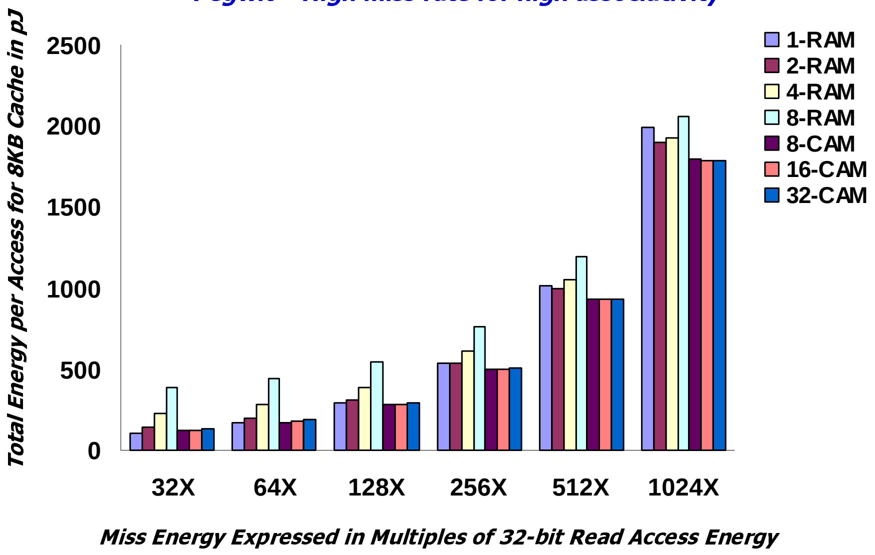


Miss Rate Results



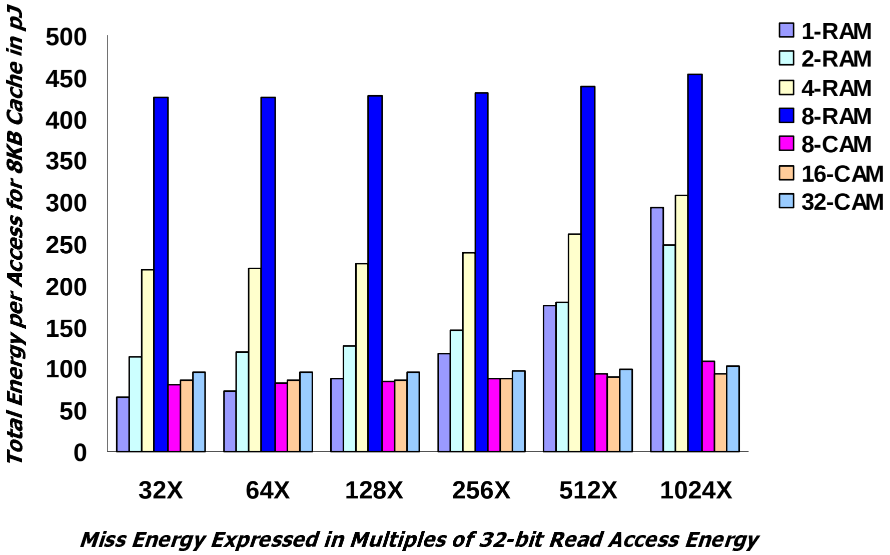
Total Access Energy (*pegwit*)

Pegwit – High miss rate for high associativity



Total Access Energy (*perl*)

Perl – Very low miss rate for high associativity



Other Advantages of CAM-tag

- Hit signal generated earlier
  - □ *Simplifies pipelines*
- Simplified store operation
  - □ *Wordline only enabled during a hit*
  - □ *Stores can happen in a single cycle*
  - □ *No write buffer necessary*

Related Work

- CACTI and CACTI2
  - □ *[Wilton and Jouppi '94],[Reinman and Jouppi, '99]*
  - □ Accurate delay and energy estimate
    - ● Results within *10%*
  - □ Energy estimate not suited for low-power designs
  - □ Typical Low-power features not included in CACTI
    - ● Sub-banking
    - ● Low-swing bitlines
    - ● Wordline gating
    - ● Separate CAM search line
    - ● Low-swing match lines
  - □ Energy Estimation 10X greater than our model for one CAM-tag cache sub-bank
    - ● Our results closely agree with *[Amruthur and Horowitz, 98]*

Conclusion

- *CAM tags – high performance and low-power*
  - □ *Energy consumption of 32-way CAM < 2-way RAM*
  - □ *Easy to implement highly-associative tags*
  - □ *Low area overhead (10%)*
  - □ *Comparable access delay*
  - □ *Better CPI by reducing miss rate*

Thank You!

http://www.cag.lcs.mit.edu/scale/