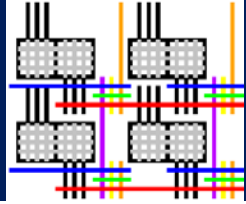


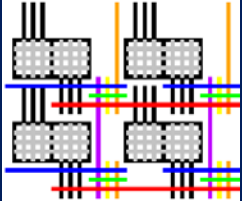
Area-Efficient Near-Associative Memories on FPGAs

Udit Dhawan André DeHon

Penn
University of Pennsylvania

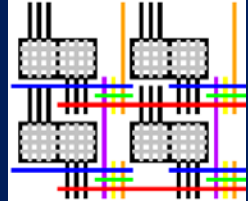
February 13, 2013

Overview

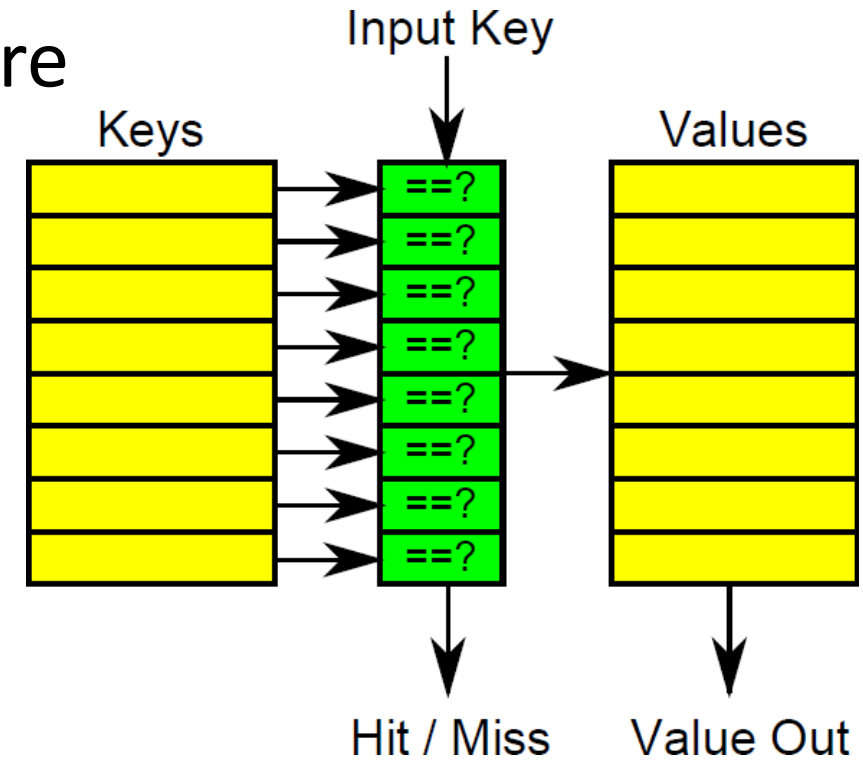


- Motivation
  - Associative Memories on Modern FPGAs
  - Our Near-Associative Solution
- Dynamic Multi-Hash Cache Architecture
  - Concept
  - Operation
  - Performance

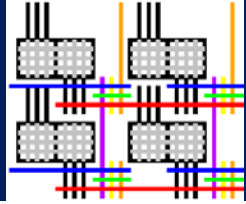
Associative Memories



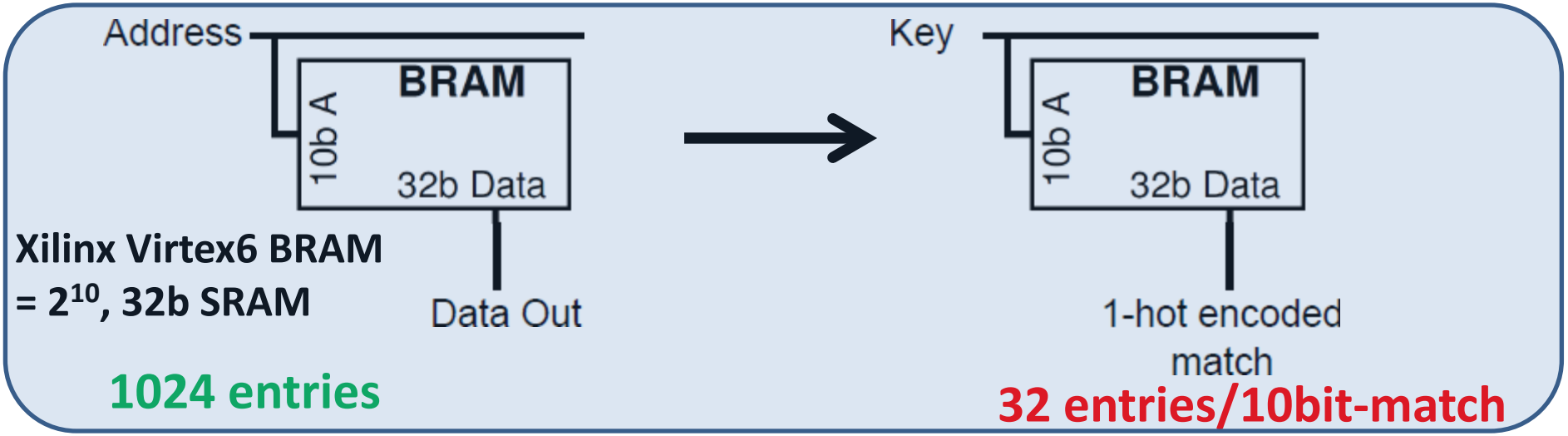
- Act as high performance parallel searching structure
- Conflict-free memories
  - capacity-bound
- Used commonly in computing structures



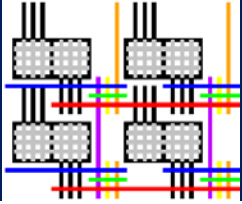
Associative Memories on Modern FPGAs



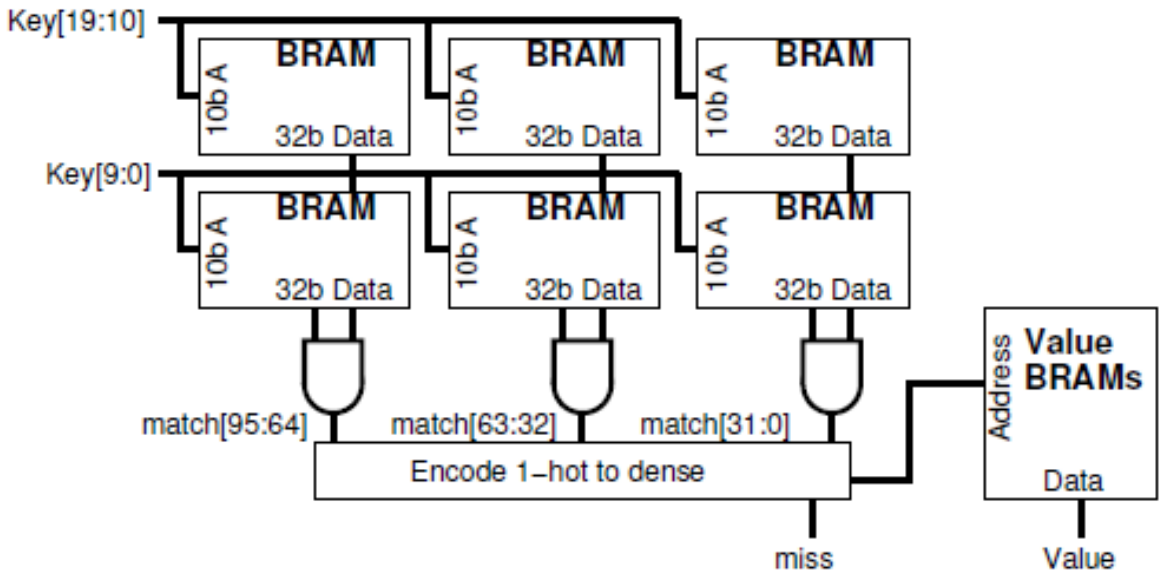
- Need to build these out of Block RAMs
  - Implement programmable match using BRAMs with fixed address-match SRAMs
- Xilinx Coregen provides a canonical design style in XAPP1151



Associative Memories on Modern FPGAs



- Can further build deeper and wider associative memories



96-entry, 20-bit key CAM

32 Kbit BRAMs (xc6vlx240t-1)

↓

512-entry CAM with 40-bit key

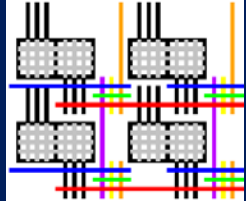
↓

64 BRAMs for performing match

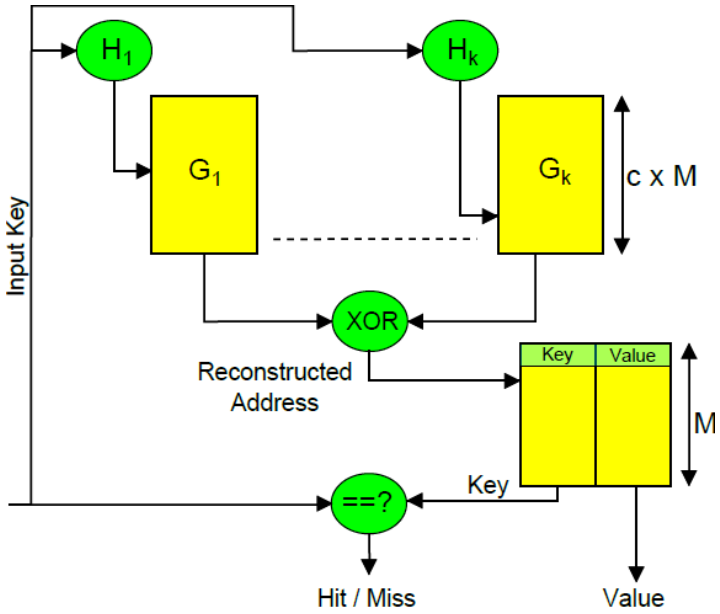
↓

1 BRAM for storing keys

Our Near-Associative Solution



- Do we really need **Full Associativity**?
  - reduce conflict probability close to 0, if not 0
    - **Near-Associativity**



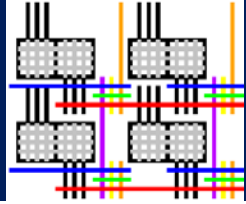
32 Kbit BRAMs

↓

5 BRAMs for key match
(conflict miss-rate → 0.06)

↓

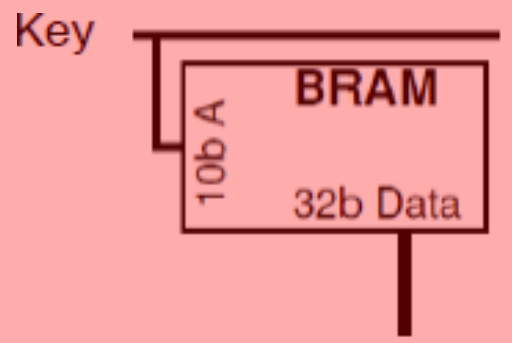
1 BRAM for storing keys



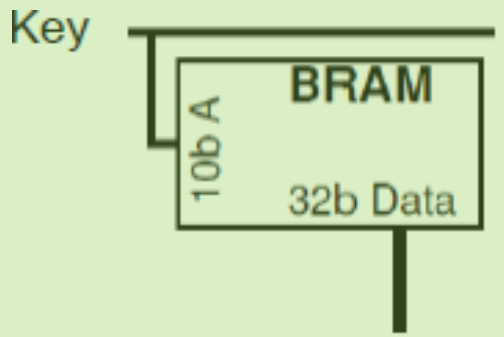
DYNAMIC MULTI-HASH CACHE ARCHITECTURE

dMHC Goals

- Goal 1:
  - Densely store only present key-value pairs
    - Coregen: exhaustive implementation



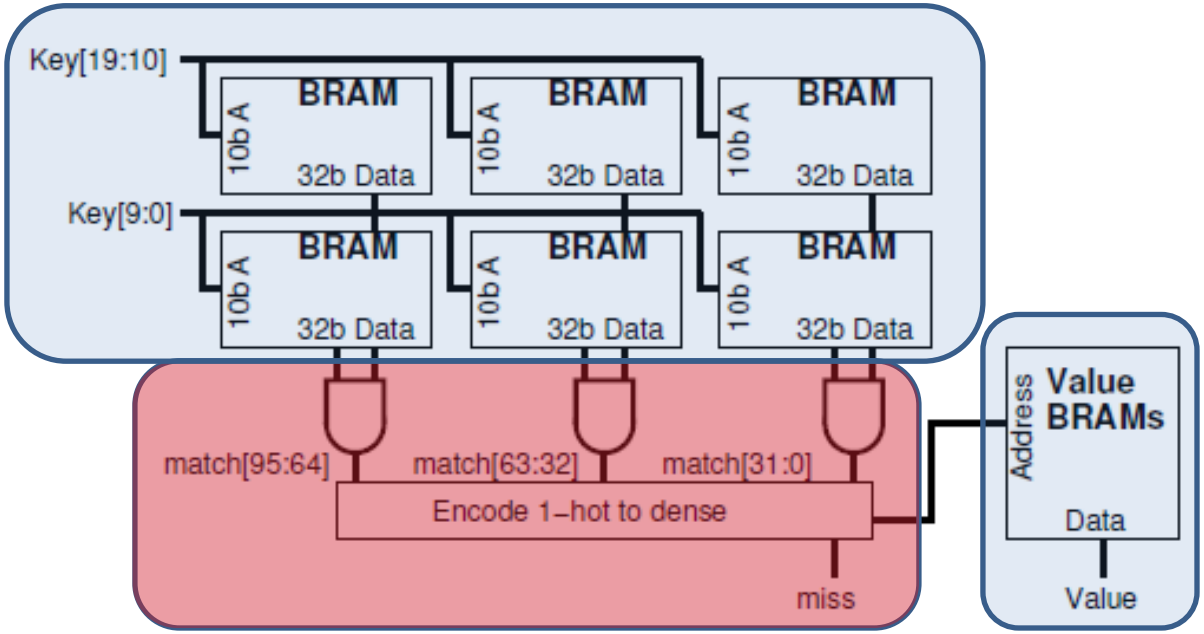
Stores 32 entries



store 1024 entries

dMHC Goals

- Goal 2:
  - Densely store match results
    - Coregen: One-hot encoded match result



dMHC Architecture

Reduce conflicts

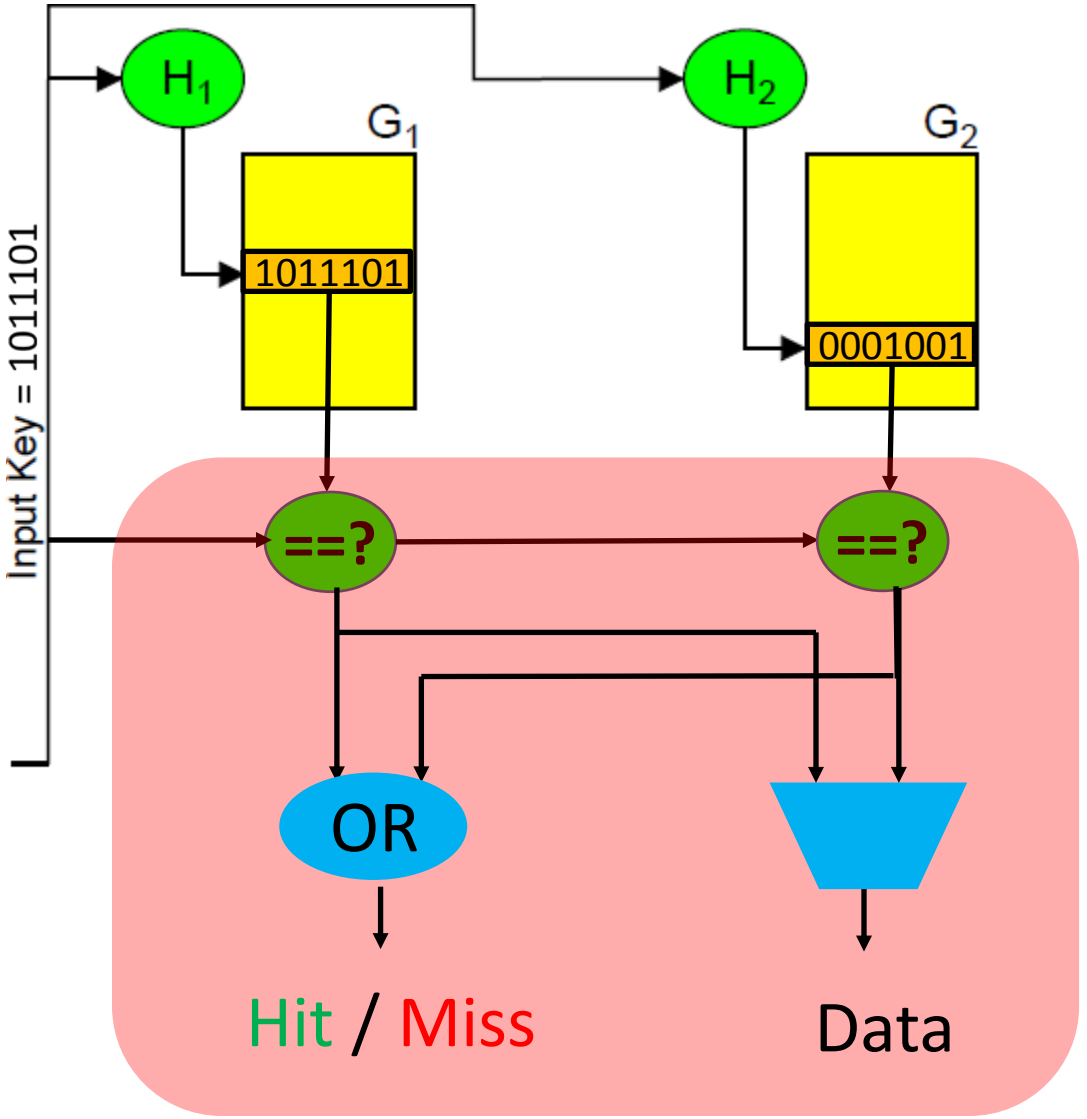
Conflict-rate = 1/c

Use multiple hash tables

Conflict-rate = $1/c^2$

Input key

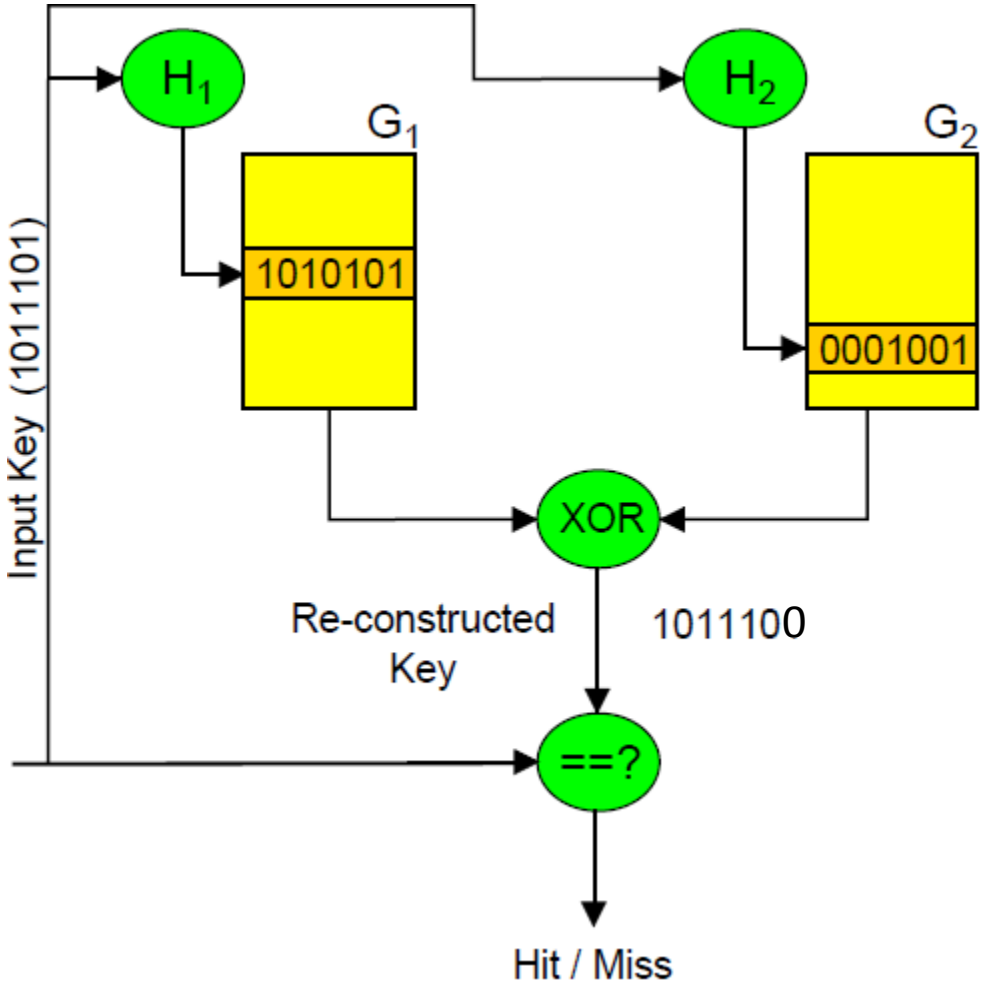
1011101



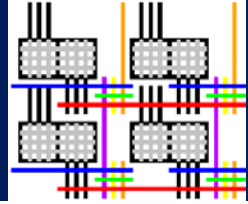
dMHC Hardware Architecture

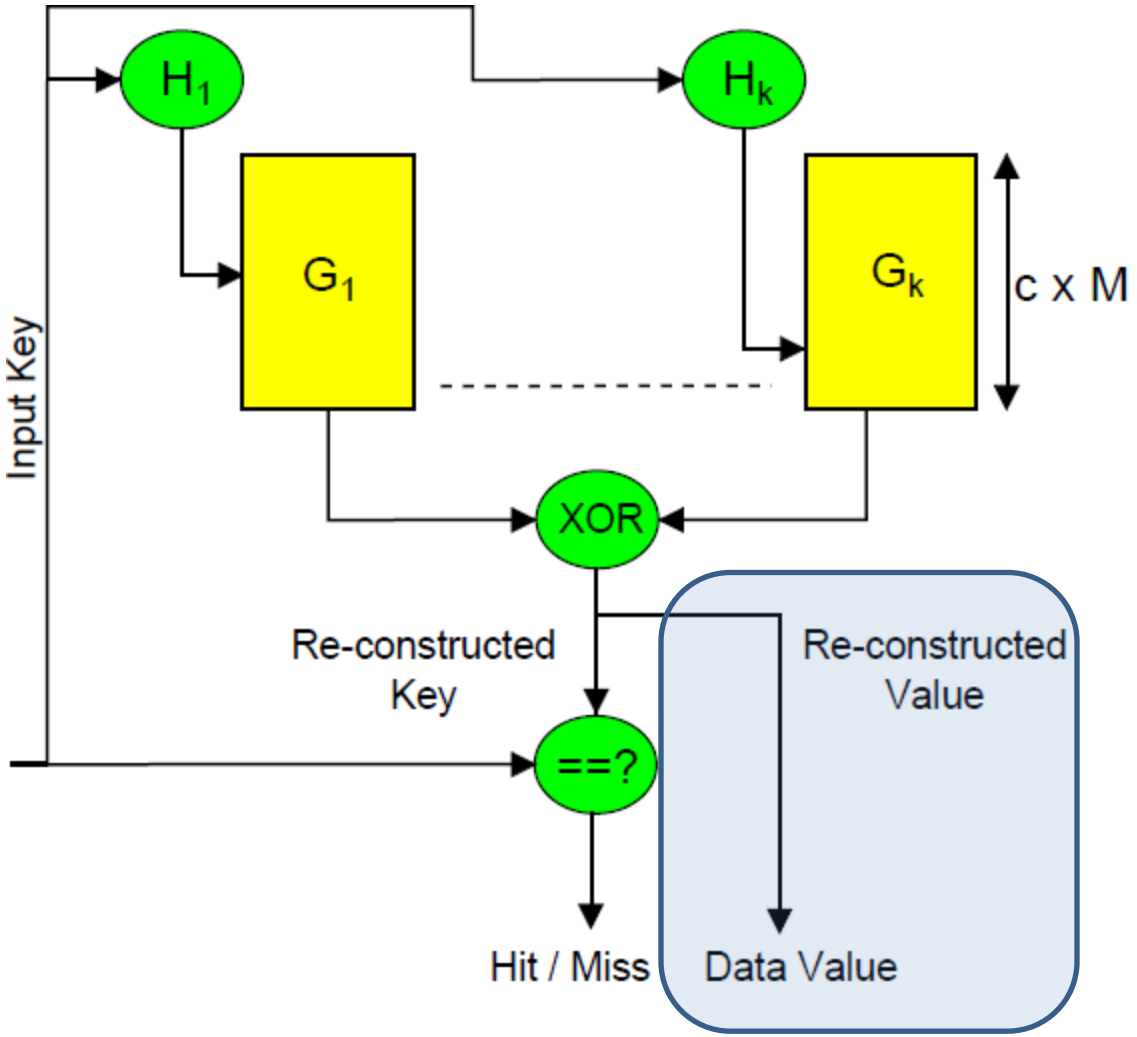
Input key

1011100



dMHC Hardware Architecture





Generalized dMHC(k,c) architecture

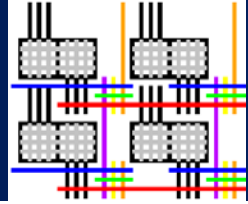
Parameterized by :

- Number of entries, M
- Number of hashes, k
- Sparsity factor, c

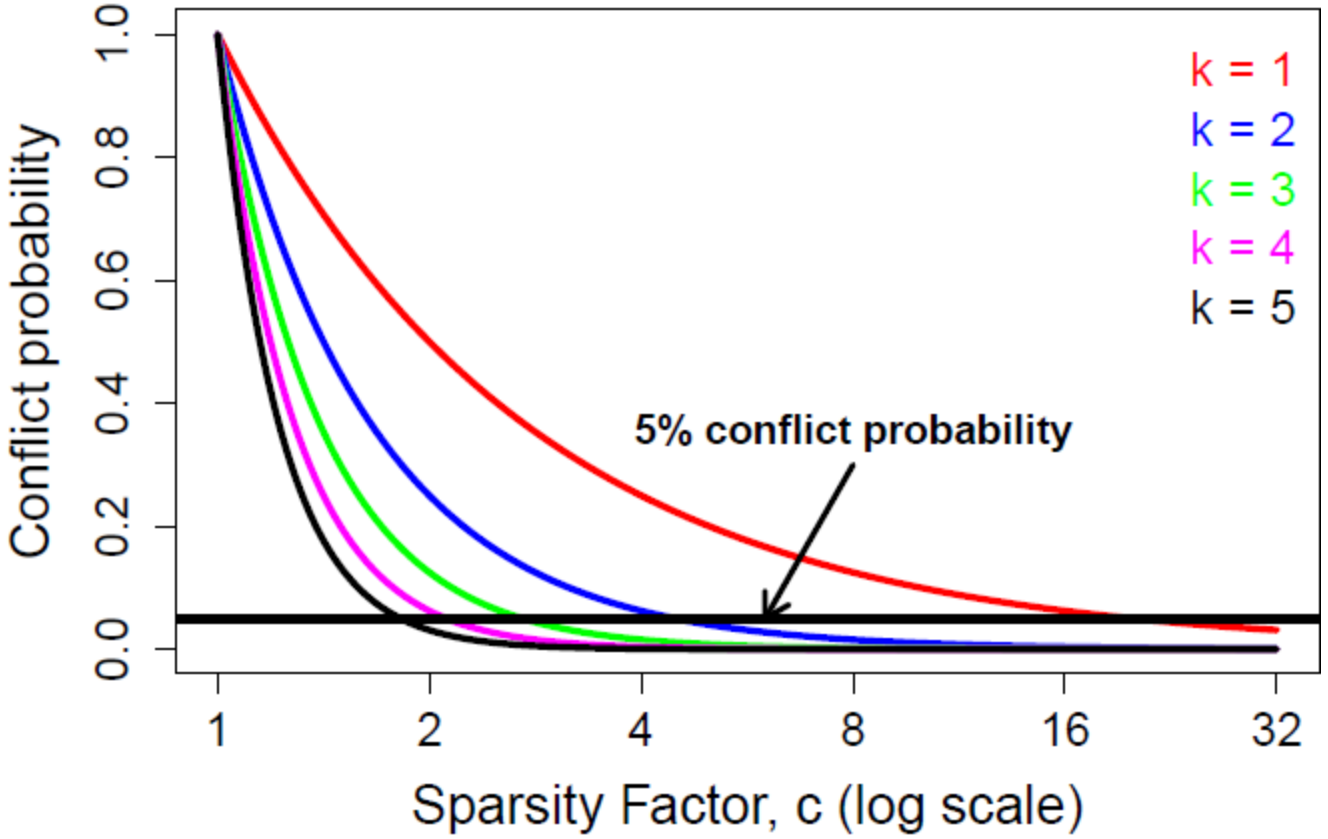
Collision Probability

$$\propto \frac{1}{c^k}$$

dMHC(k,c) Conflict Probability



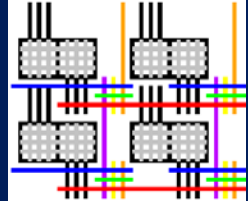
Define near-associative conflict-rate: 5% of evictions

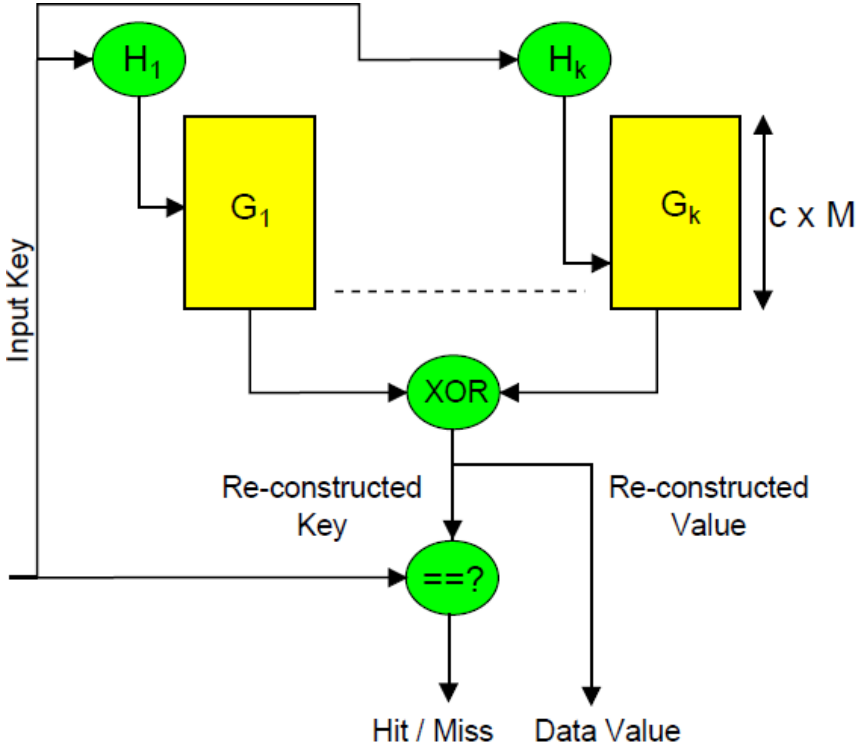


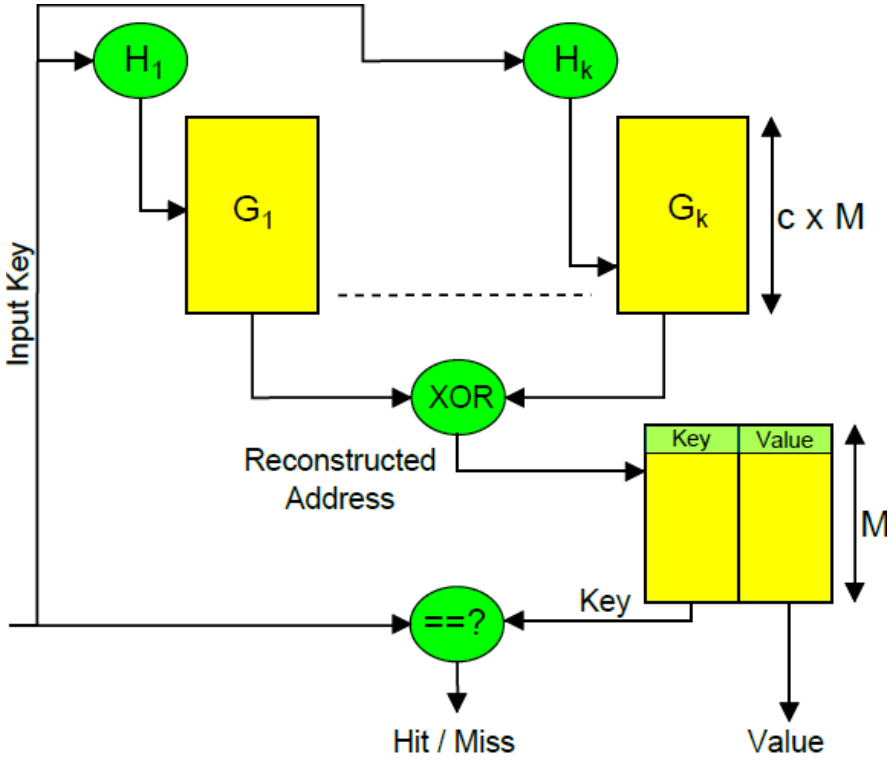
k=4, c=2 -> 5% criterion

Conflict probability configurable in terms of k and c

dMHC Variants



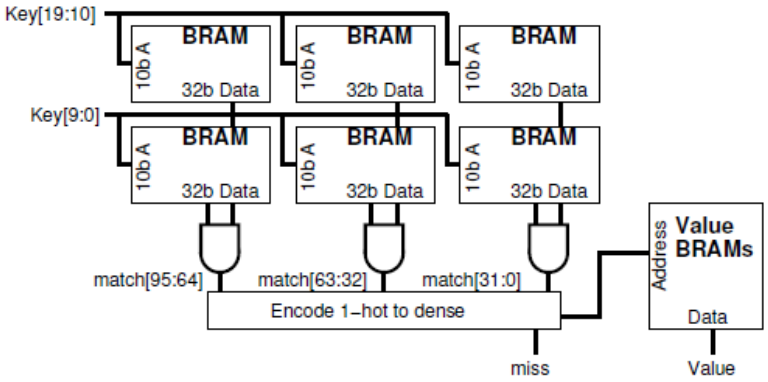


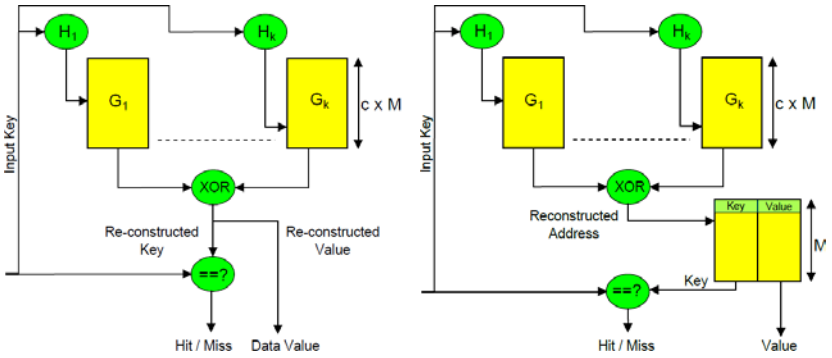


- Flat dMHC
- 1 cycle latency
- Wide G tables

- Two-Level dMHC
- 2 cycle latency
- Narrow G tables

dMHC: Area Benefits over Coregen





True Fully Associative

↓

exhaustive population of keys

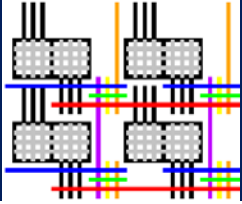
Near-Associative dMHC

↓

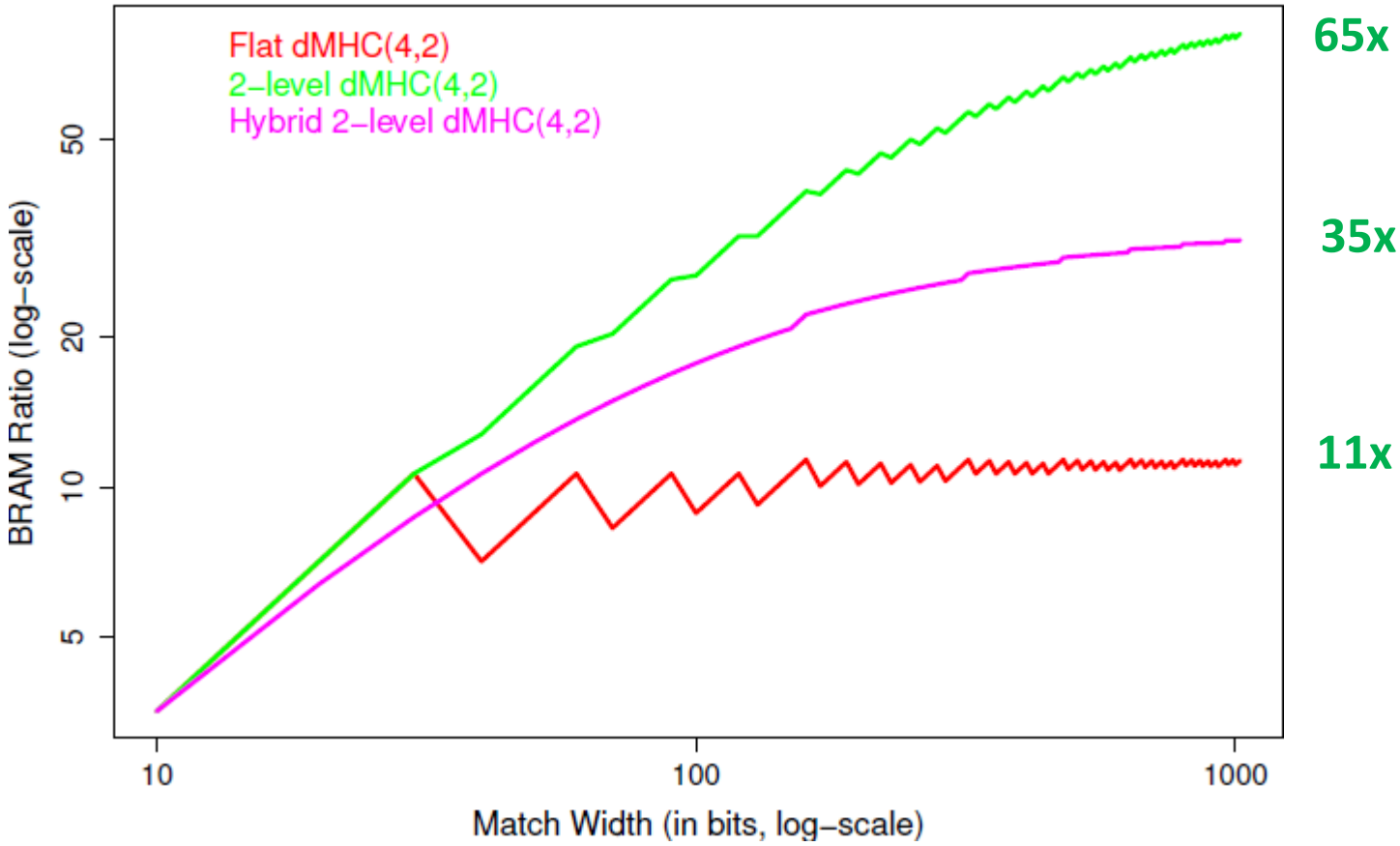
- only store present key-value pairs
- uses BRAMs efficiently

dMHC able to store the key-value pairs more densely than the exhaustive solution

dMHC: Area Benefits over Coregen

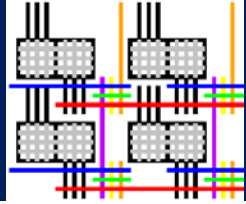


1024-entry memory



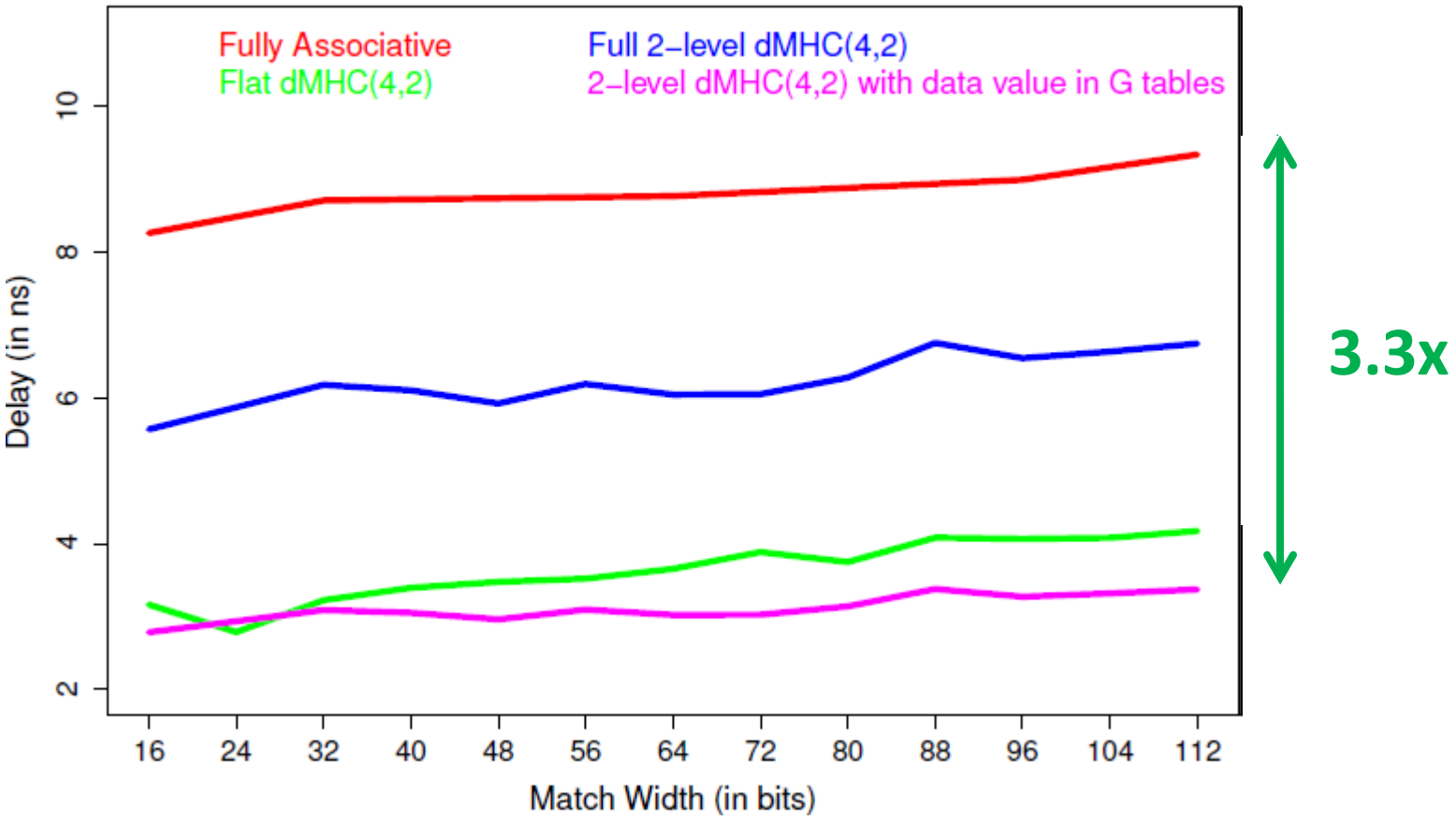
2-65x savings in BRAM consumption

Timing Comparison

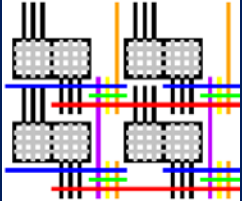


- Implemented instances of dMHC variants
- Fully placed and routed designs on Xilinx Virtex 6 (xcv6vlx240t-1) using ISE 13.2
- 1024-entry memory with 64-bit data values
  - Vary key-width

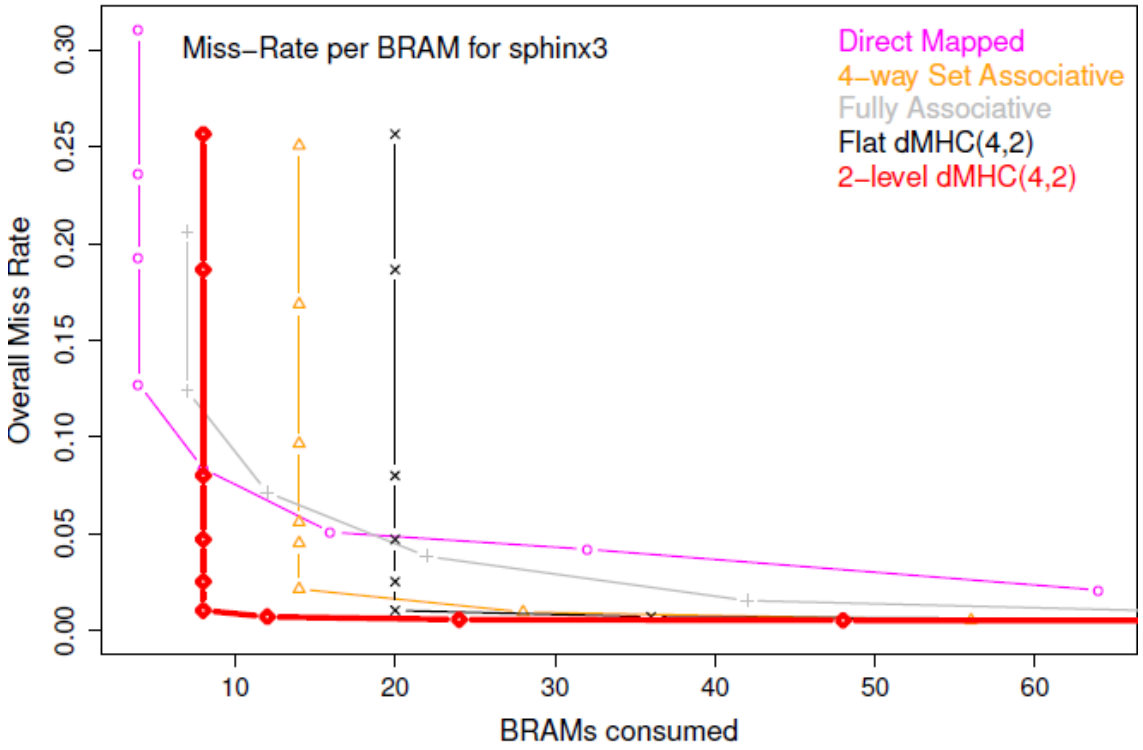
Timing Comparison



BRAM v/s Miss-Rate

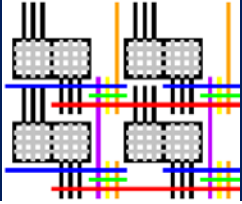


- Express in terms of BRAMs per unit miss-rate
  - L1 D-$ for sphinx3 benchmark



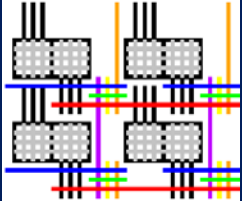
Two-level dMHC → lowest miss-rate per unit BRAM

Related Work



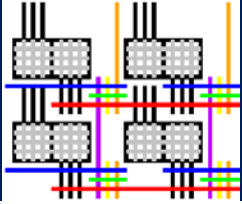
- dMHC derives heavily from
  - Perfect hashing schemes
    - only possible for static data
  - Bloom/Counting Bloom filters
    - however, they only report set membership
  - Multi-Level Hash Tables (Kirsch [TON 2010])
    - Can have multiple cycle latency
  - ZCache (Sanchez [MICRO 2010])
    - Hash-based set-associative cache

Conclusion



- Associative memories expensive
- Can achieve near-associative performance with significant area savings and timing improvement
  - Upto 65x area, 3.3x delay
- Configurable in area/miss-rate
  - Target BRAM budget
  - Achieve a certain conflict miss-rate
- Bluespec source code available online:
  - http://ic.ese.upenn.edu/distributions/dmhc_fpga.html

End



Thank you