Deep and Narrow Binary Content-Addressable Memories using FPGA-based BRAMs

a place of mind
THE UNIVERSITY OF BRITISH COLUMBIA

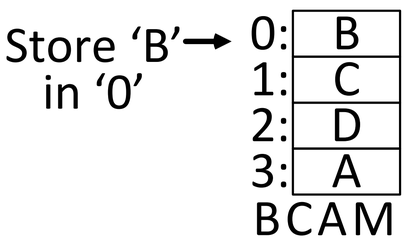
Ameer M.S. Abdelhadi and Guy G.F. Lemieux
{ameer,lemieux}@ece.ubc.ca

Department of Electrical and Computer Engineering
The University of British Columbia, Vancouver, Canada

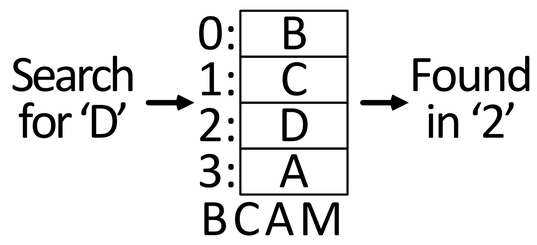
Binary Content-Addressable Memories (BCAMs)

Hardware-based Single-Cycle Parallel Search Engines

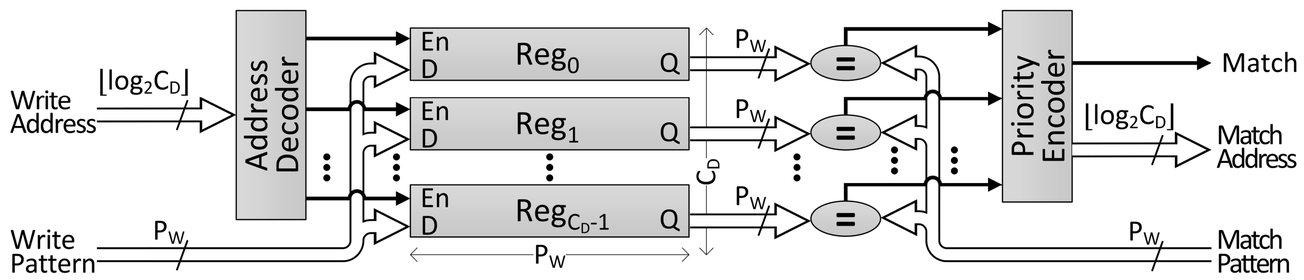
Write
Stores new data at specific address



Match
Search all addresses for a given data (pattern)



Register-based BCAM: Concurrent register read and compare

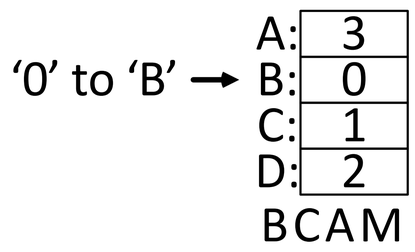


Single-cycle | Limited resources | Complex routing | Fits small BCAMs

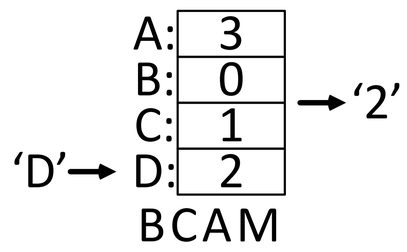
Traditional Approach: Brute-Force Transposed-Indicators-RAM (TIRAM)

Key idea: Transposed RAM – data becomes address

Write
Write '0' to location 'B'

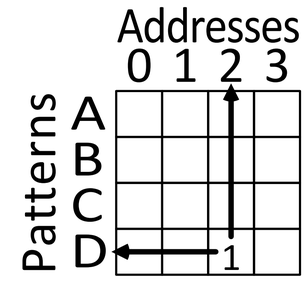


Match
Read location 'D' for match



Store data to multiple addresses:

- Specify addresses using one-hot coding
- Each bit indicates a match or "store at location"



Details given in paper, similar to implementations provided by FPGA vendors

SRAM-based BCAM | Single-cycle match / two-cycles write | Depth of CAM is limited by RAM data width

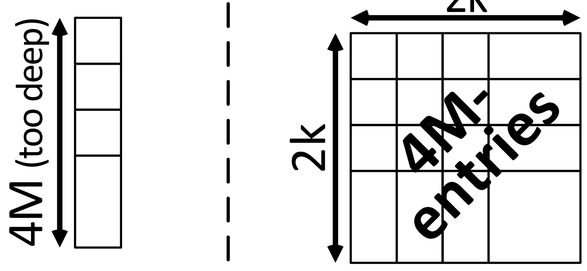
Segmented-Transposed-Indicators (STIRAM)

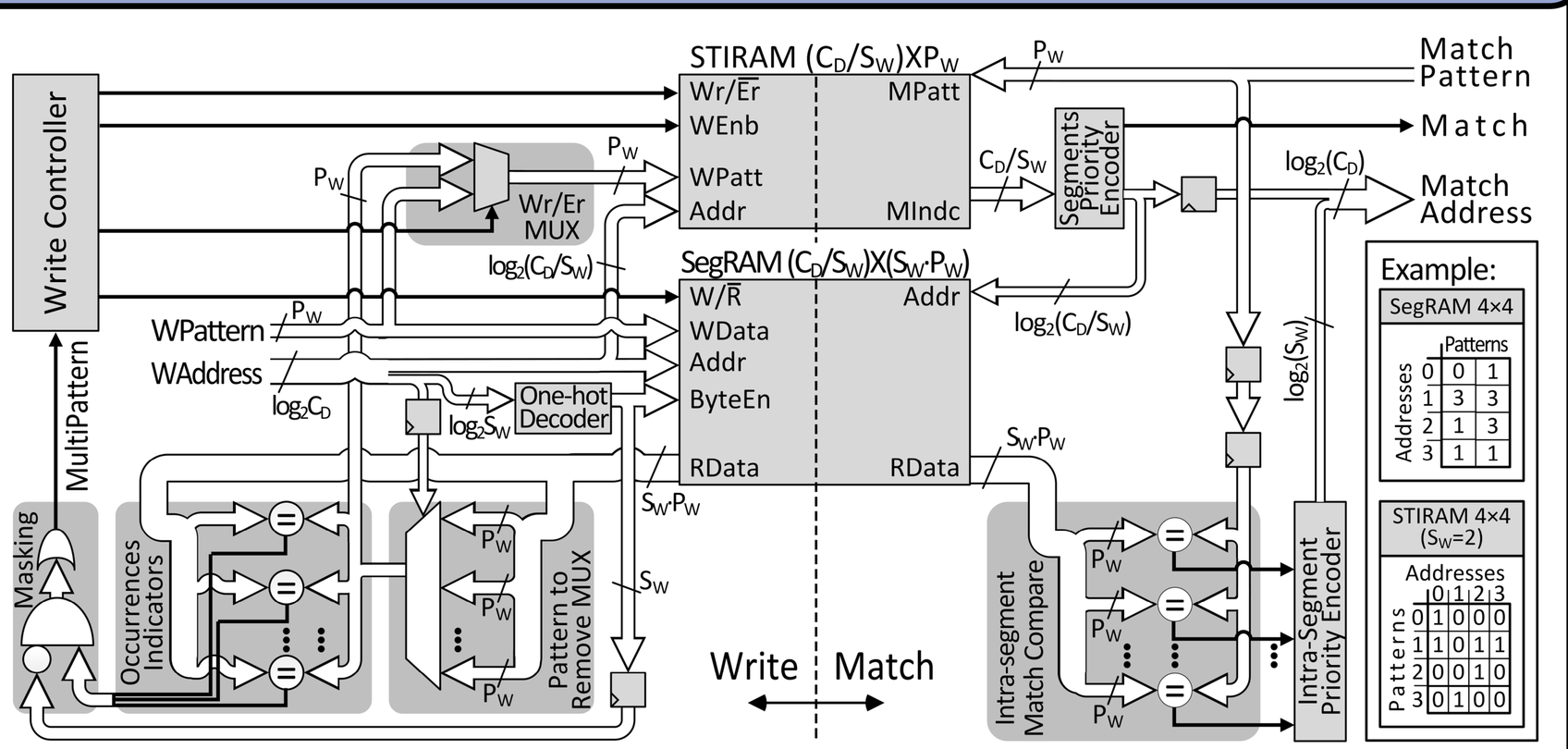
Key contribution:
Support very deep BCAMs

Key idea: Hierarchical search

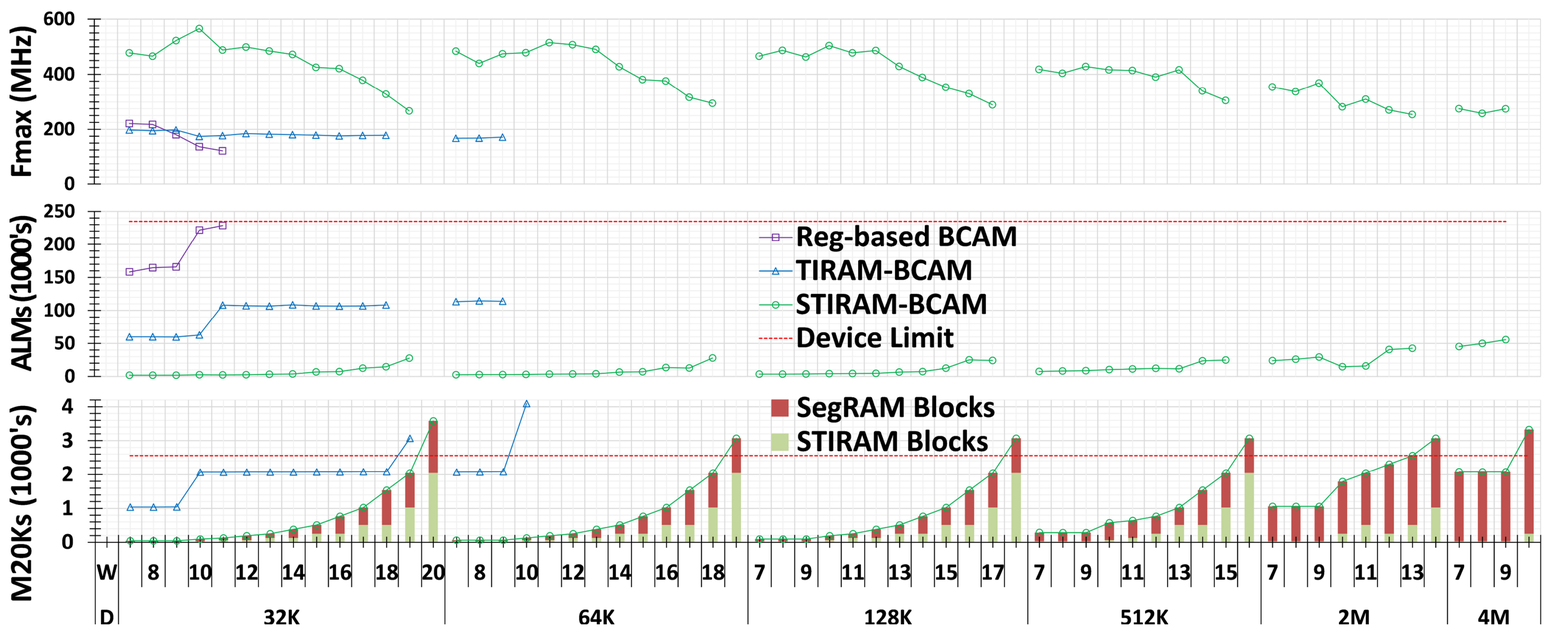
- Divide address space into segments
- Find a row (segment) with match using a 1D BCAM (*e.g.* TIRAM)
- Search this row (segment) in parallel for a specific match

1D BCAM | 2D BCAM





Experimental Results



- Altera's high-performance Stratix V device with 235k ALMs and 2560 M20K blocks
- Write : 2 cycles throughput / latency
- Match: 1 cycle throughput / 2 cycles latency
- Most efficient for deep and narrow CAMs
- Can implement a **4M-line** CAM
- Completely **dominates** past designs in area, Fmax and compile-time
- BRAM consumption is exponential to pattern width