Energy Efficient Content-Addressable Memory

Advanced Seminar Computer Engineering

Institute of Computer Engineering
Heidelberg University
Fabian Finkeldey

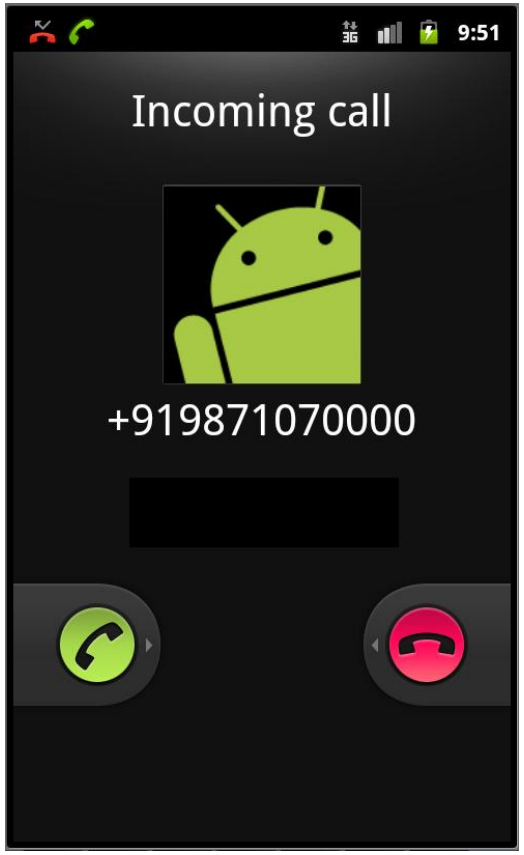
Table of Contents

Introduction

Use-Cases and basic design of content-addressable memory

Example: Looking up a phone number

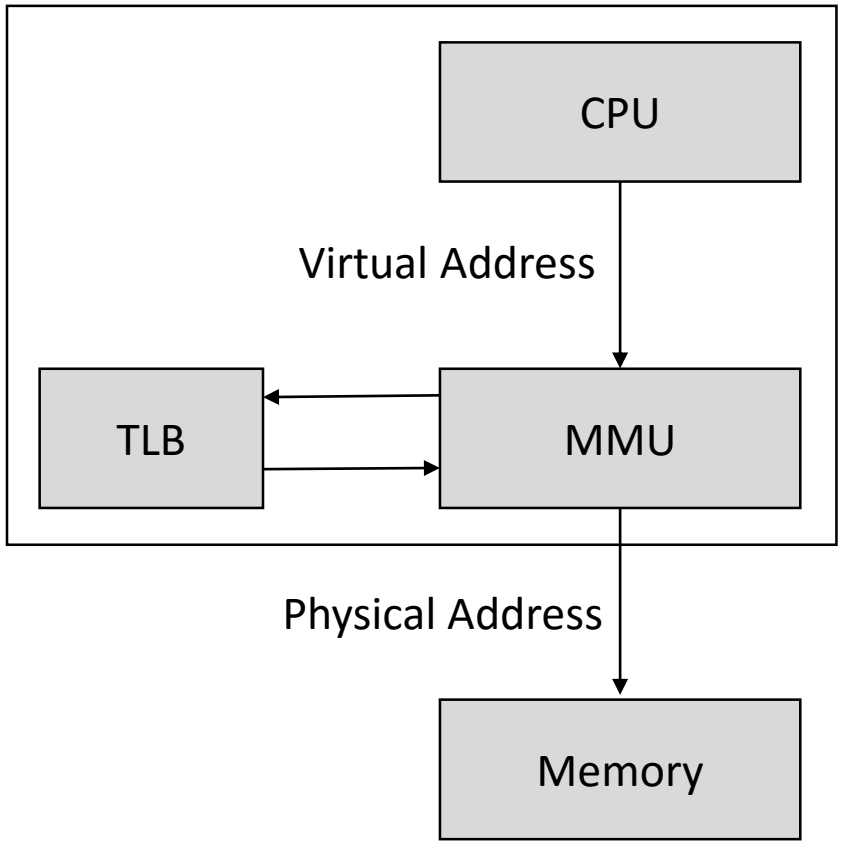
- Problem: Find a name to a given phone number
- Linear Search:
  - Looking up every entry in a phone book takes a lot of time...
- A phonebook that can deliver a name to a given number is needed



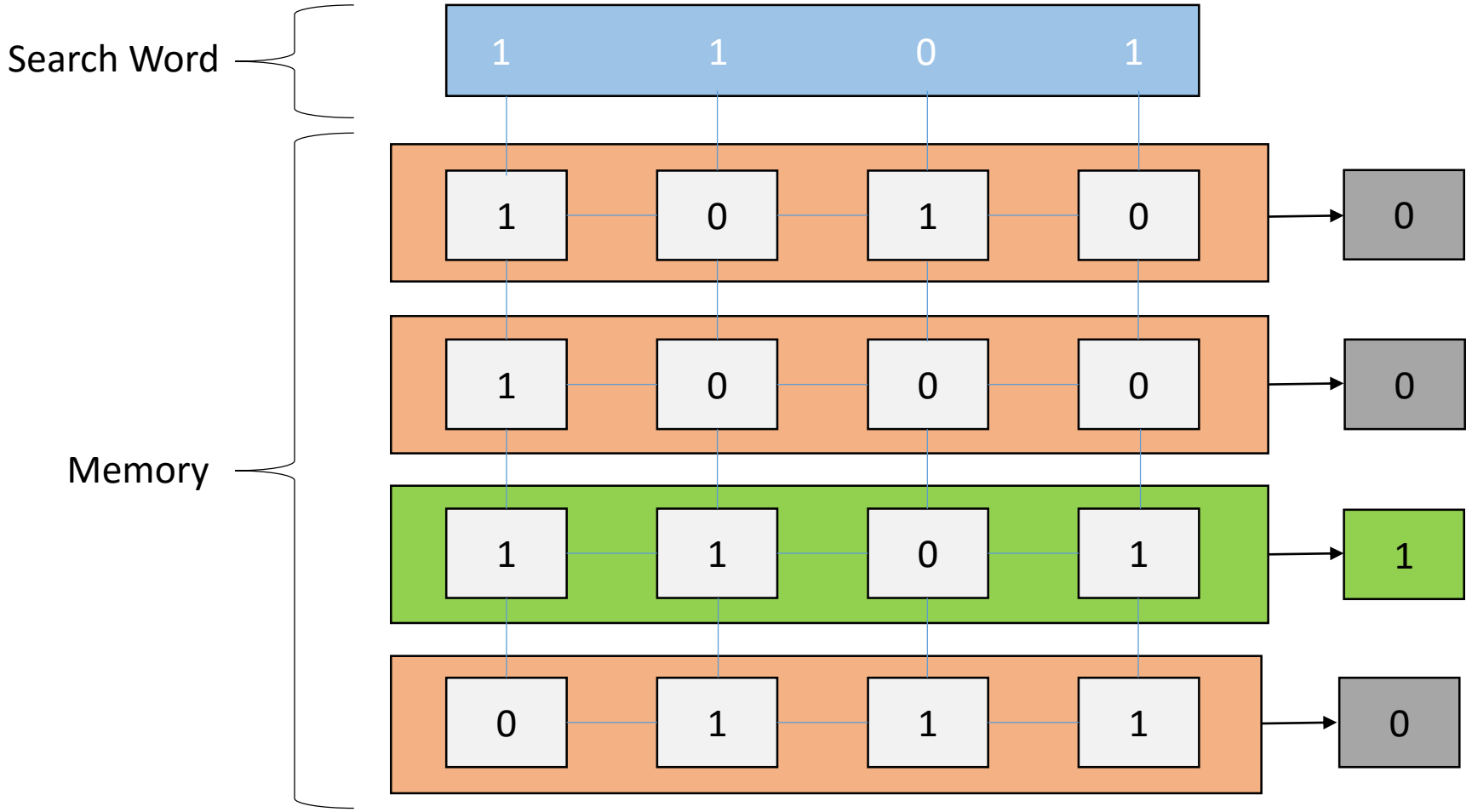
Ref. 3

Example II: Translation Lookaside Buffer

- TLB: Cache for address translation
  - Typical size ~1024 entries
  - Faster than page table access
- Needs to be searched:
  - Search-key: Virtual Address
  - Search-result: Physical Address



Basic concept of a CAM

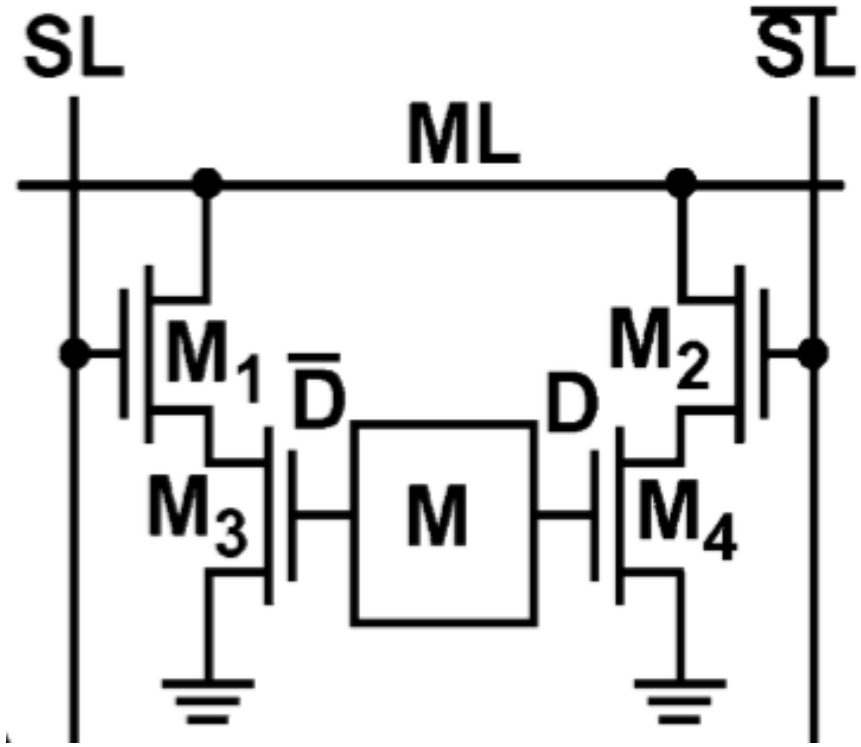


Standard Circuit Design

A conventional CAM

Conventional CAM Cell

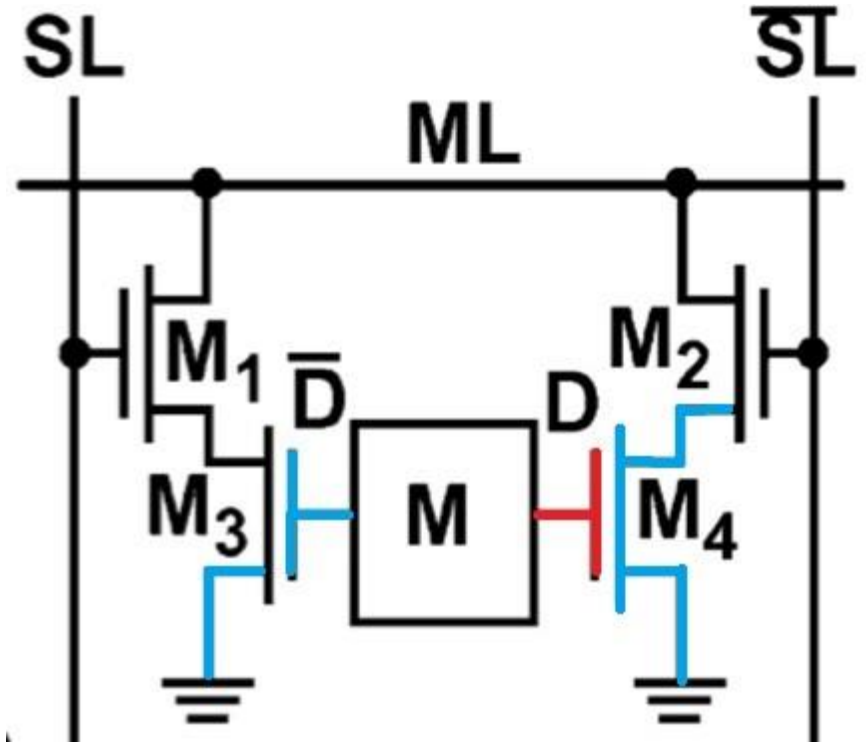
- M is a standard SRAM-Cell
- D is the stored Data value
- Search-Data is applied to SL
- ML indicates match-state



Ref. 1

CAM Cell – search operation 1

- Assume:
  - D = 1 -> $\overline{D}$ = 0

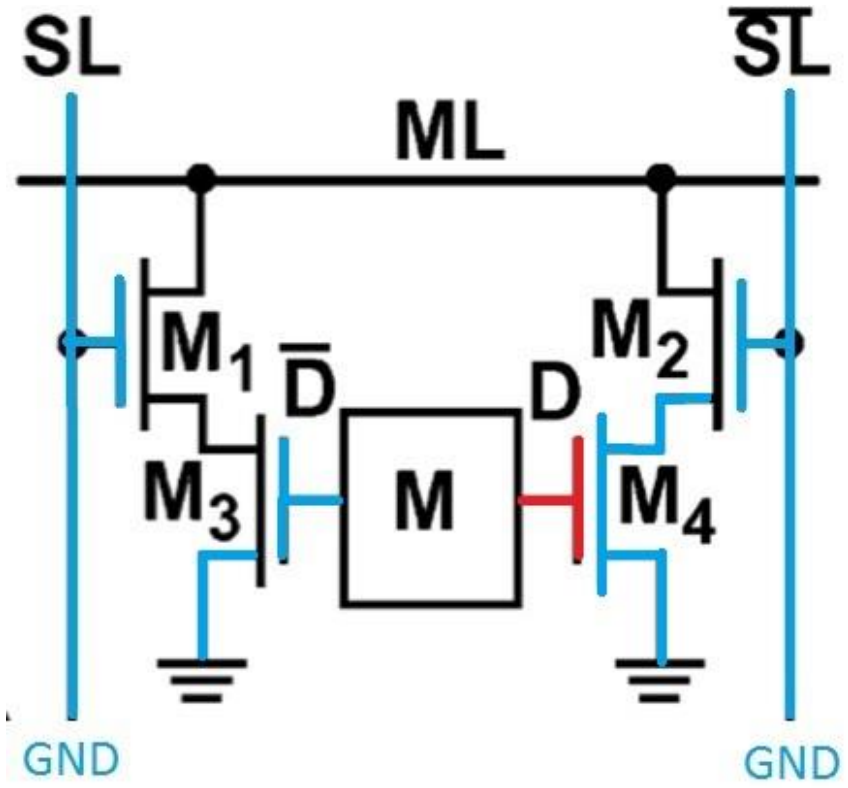


Ref. 1

CAM Cell – search operation 1

1. SL = 0 and $\overline{SL}$ = 0

   -> M_1 and M_2 are switched off



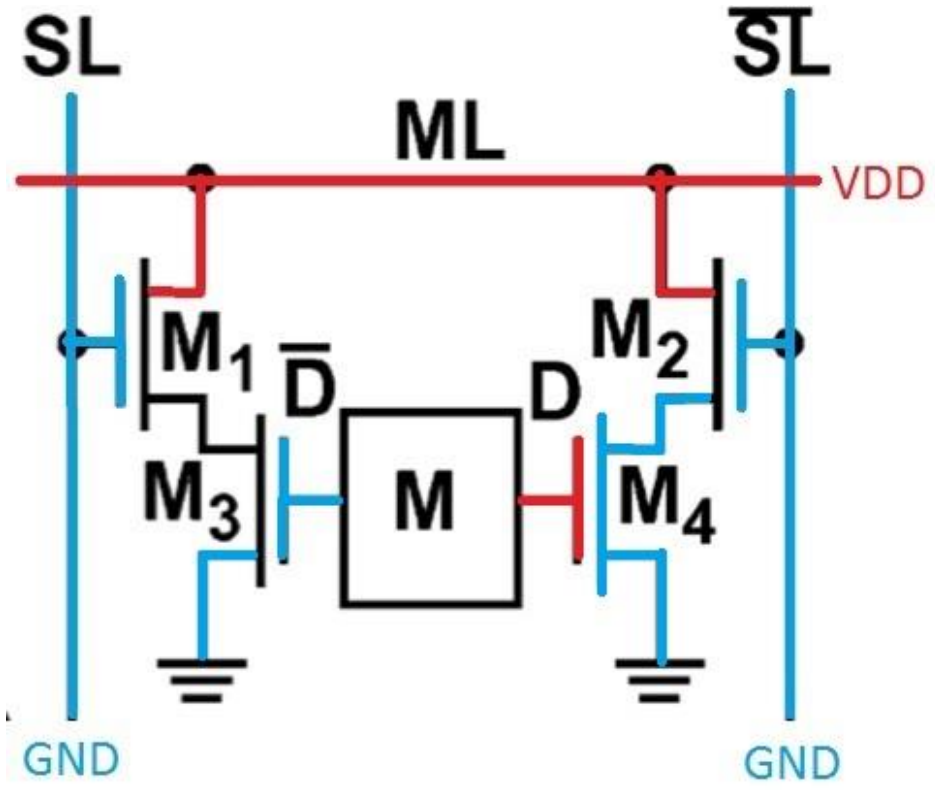
Ref. 1

CAM Cell – search operation 2

1. SL = 0 and $\overline{SL}$ = 0
2. ML is precharged to V_{DD} -> ML = 1

   M_1 and M_2 are switched off

   -> No path ML to GND



Ref. 1

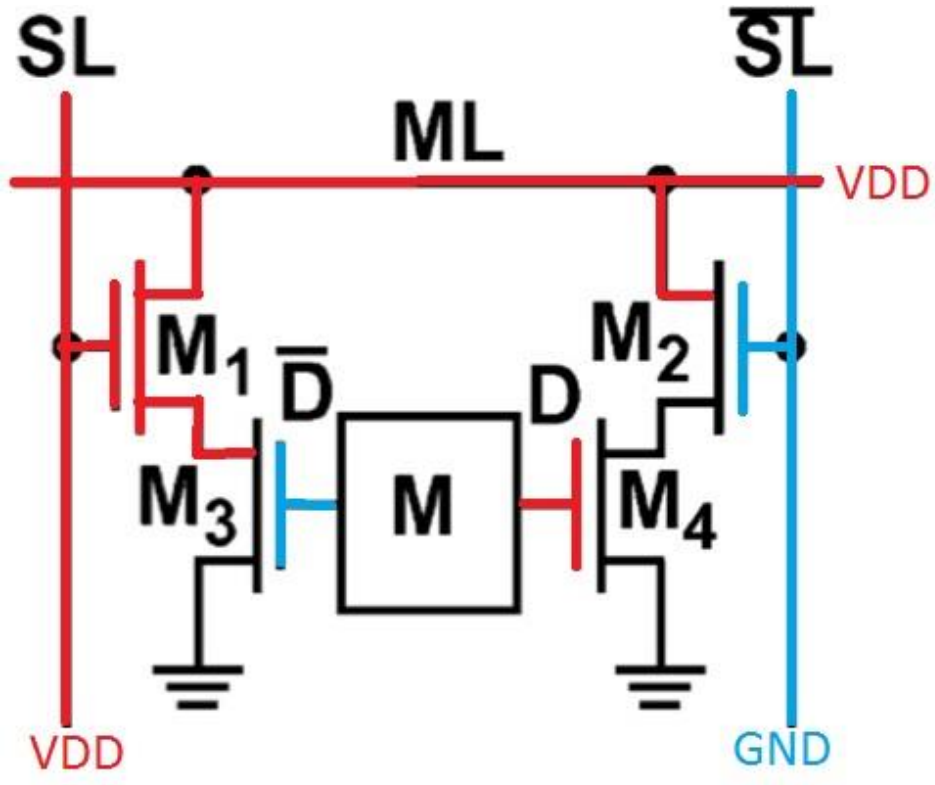
CAM Cell – search operation 3 (match)

1. SL = 0 and $\overline{SL}$ = 0
2. ML is precharged to V_{DD} -> ML = 1
3. Assume SL = 1 -> $\overline{SL}$ = 0

   M_2 and M_3 are switched off

   -> No path ML to GND

ML stays at VDD -> ML = 1, match!



Ref. 1

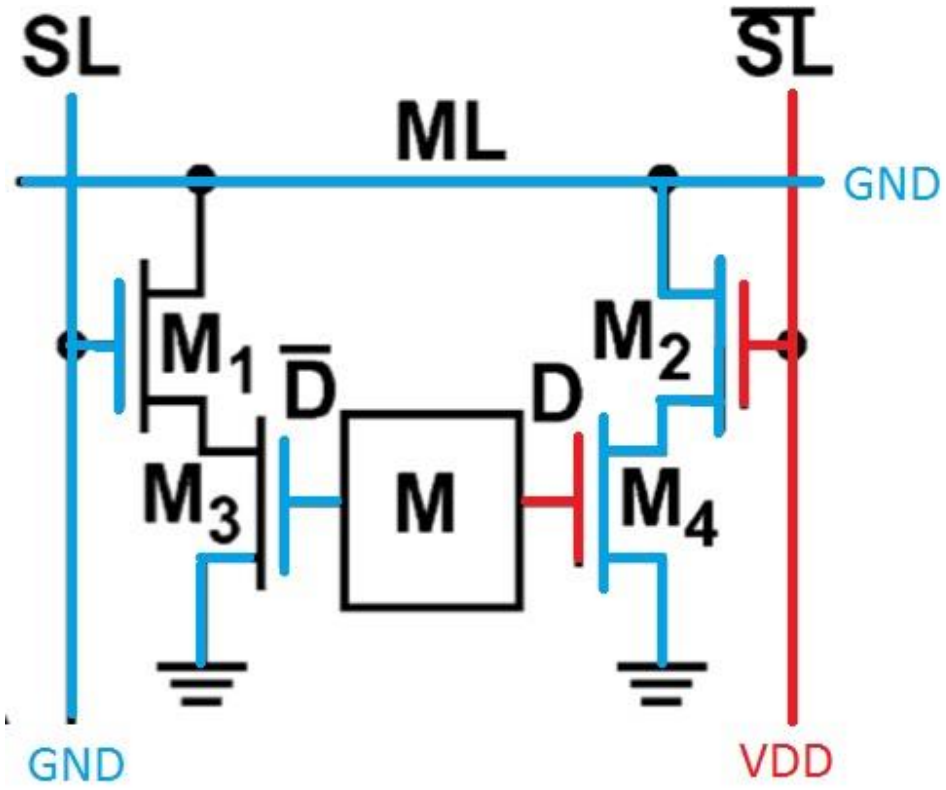
CAM Cell – search operation 3 (mismatch)

1. SL = 0 and $\overline{SL}$ = 0
2. ML is precharged to V_{DD} -> ML = 1
3. Assume SL = 0 -> $\overline{SL}$ = 1

   M_2 and M_4 are switched on

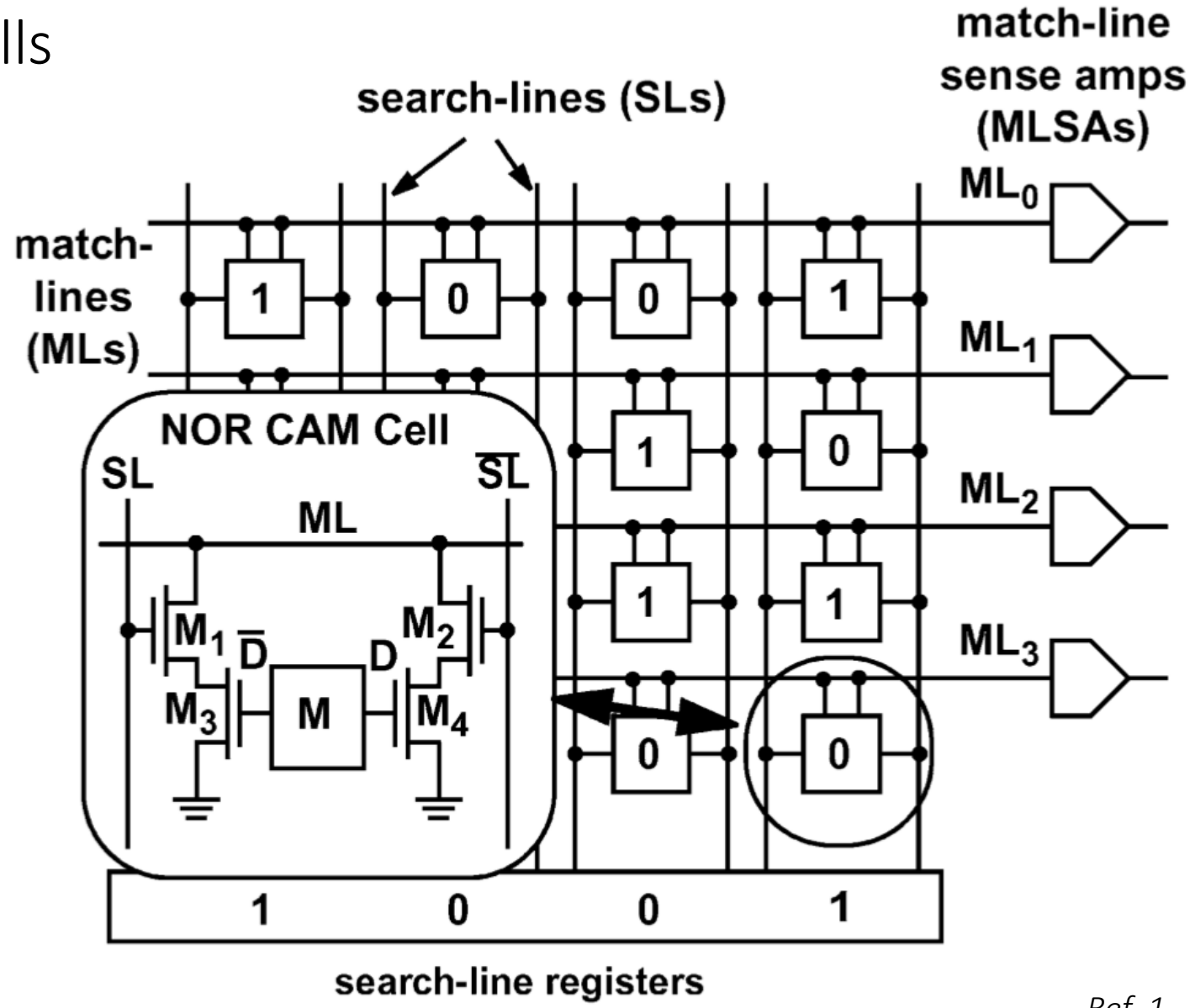
   -> Path ML to GND

ML discharges -> ML = 0, no match



Ref. 1

Array of Cells



Ref. 1

Power consumption

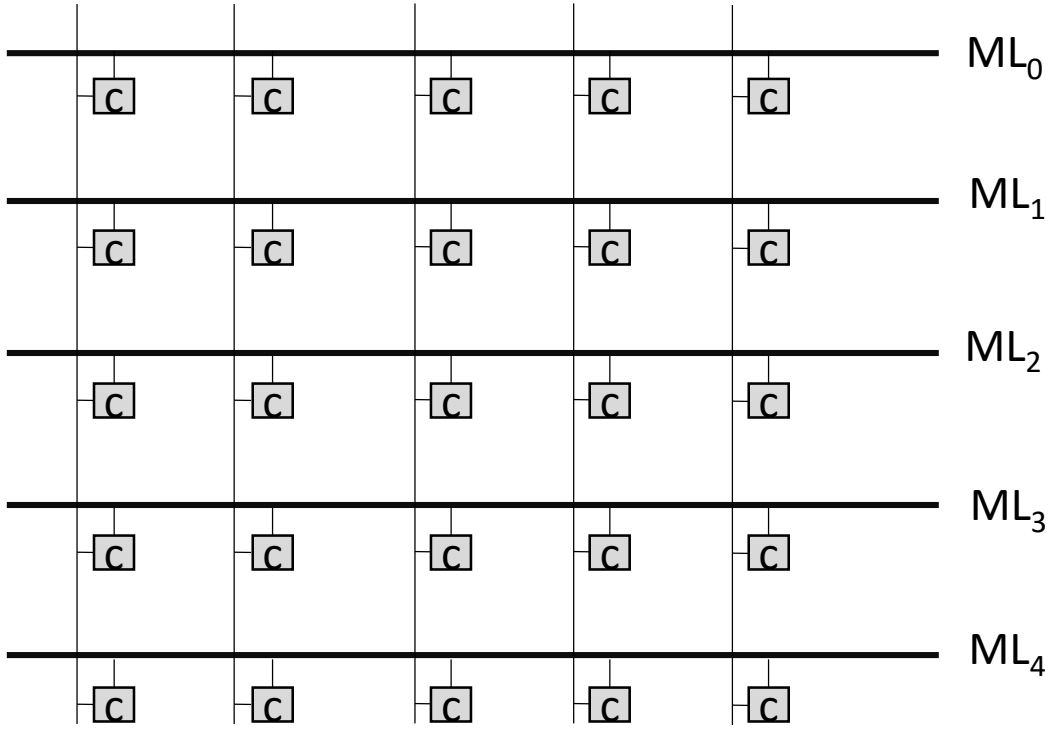
- **Matchlines:**
  - Long Lines with high capacitance:
    - Wire capacitance
    - Diffusion capacitance of the pull-down Transistors
  - Assumption: Miss in most cases
  - ML is precharged and discharged in every cycle
- **Searchlines:**
  - Long Lines with high capacitance:
    - Wire capacitance
    - Gate capacitance of the match-Transistors
  - SL and $\overline{SL}$ are pulled to GND in every cycle
  - Either SL or $\overline{SL}$ is charged to V_{DD}

Energy Efficient Design

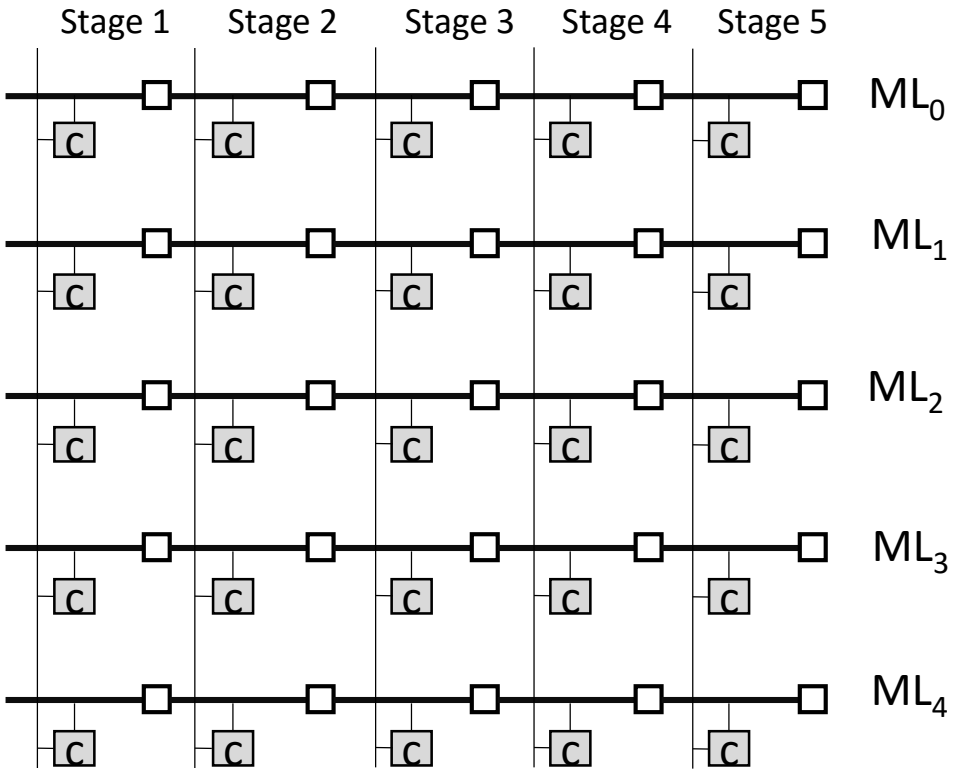
Reducing the power consumption

1. Pipelining the match line

Non-pipelined



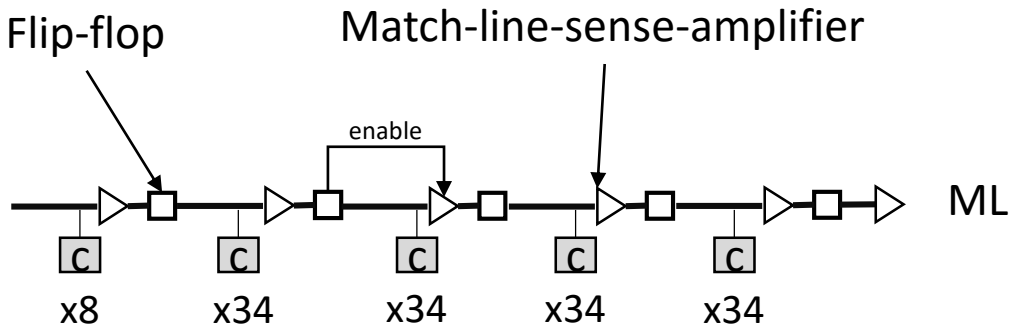
Pipelined



1. Pipelining the match-line

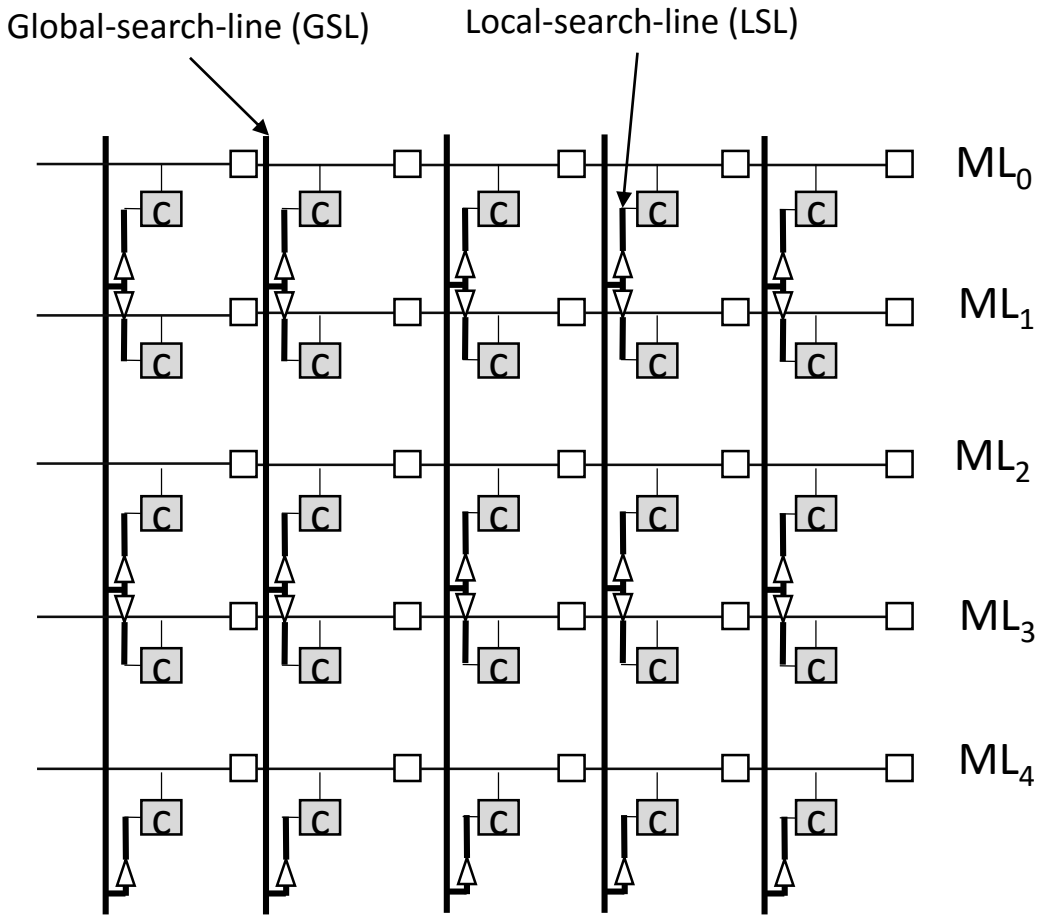
- Breaking up the long ML in stages
- In case of match, the following stage is activated
- In most cases, the ML is only partly precharged

  -> Reduced power consumption



2. Hierarchical search-lines

- Global-search-lines
  - Not directly connected to CAM-cells
    - To reduce capacitance
  - Driven in every cycle
- Local-search-lines
  - Short, connected to a few CAM-cells
  - enabled, if match-line-segment is activated



2. Hierarchical search-lines

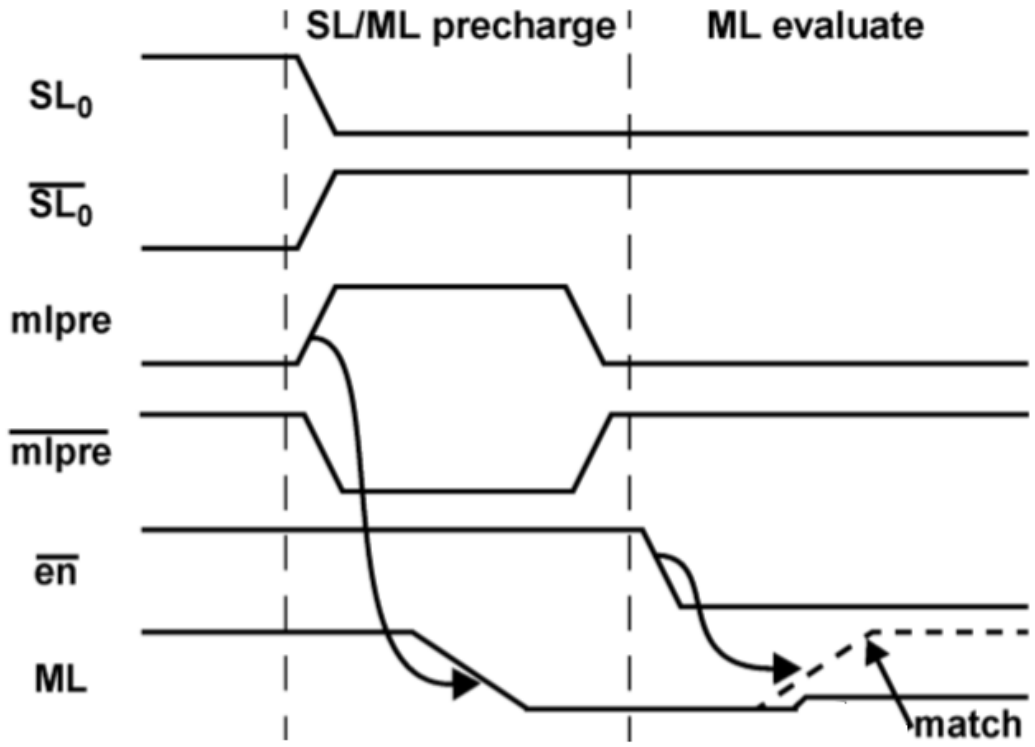
- SL power consumption: $P_{sl} = C_{sl} * V_{sl}^2$
- Usually: $V_{sl} = V_{DD}$
- Lower voltage reduces power consumption
  - Lower Gate-Overdrive -> Decreased Speed

- Solution:
  - Lower voltage V_{DDLow} on global-search-lines
  - Amplifier to drive local-search-lines with V_{DD}

3. ML-Precharge low

- ML-Precharge high requires to precharge SL low
  - Contributes to SL power consumption
- Precharge low:
  1. Discharge all ML to GND
  2. Apply Data to SL
  3. Drive fixed current I_{ML} to all ML
  4. In match state there is no path to GND

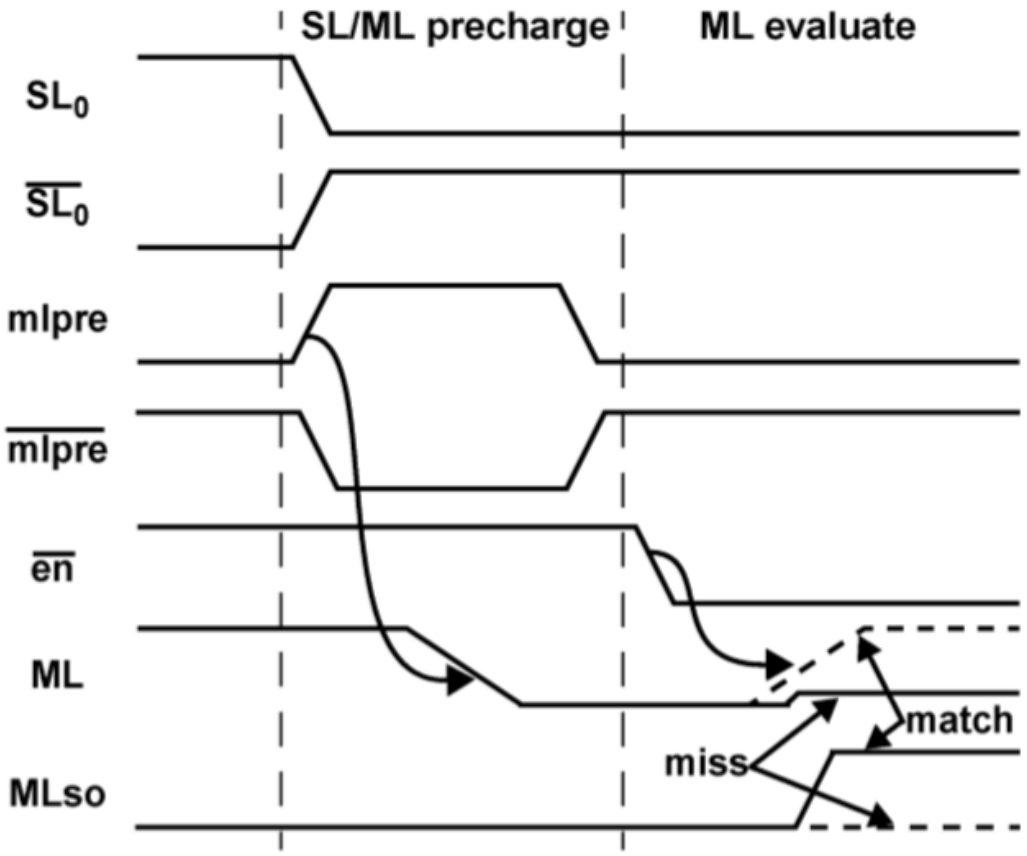
     -> Voltage will rise



Ref. 1

3. ML-Precharge low

- Low Swing on match-line
  - Match-line-sense-amplifier triggered at $V_{Th} < V_{DD}$
    => No need to charge ML to V_{DD}
- No need to precharge SL low
  - No problem with path to GND in cells in mismatch state



Ref. 1

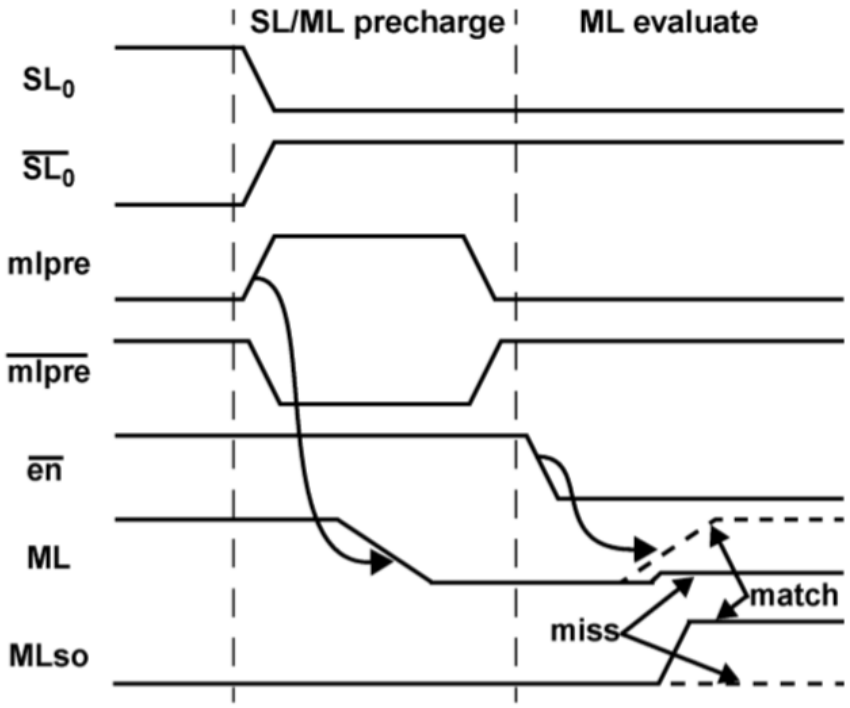
Simulation

Simulation Setup

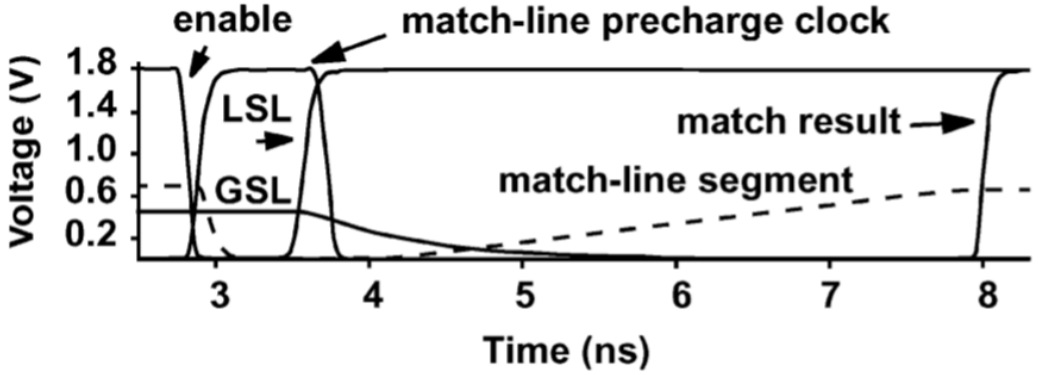
- 1024 x 144 Bit Cam
- 1x 8Bit, 4x 34Bit Segments
- 180nm Cmos
- 1,8V VDD

- Typical Workload:
  - Populated with random data
  - 1 Match per search

Schematic vs. Waveform



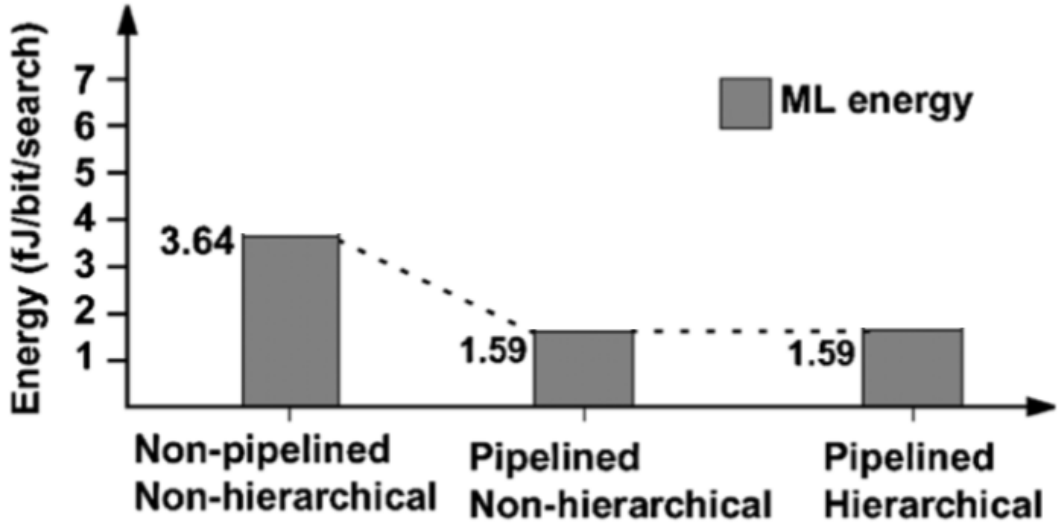
Ref. 1



Ref. 1

Simulation: Pipelined Matchlines

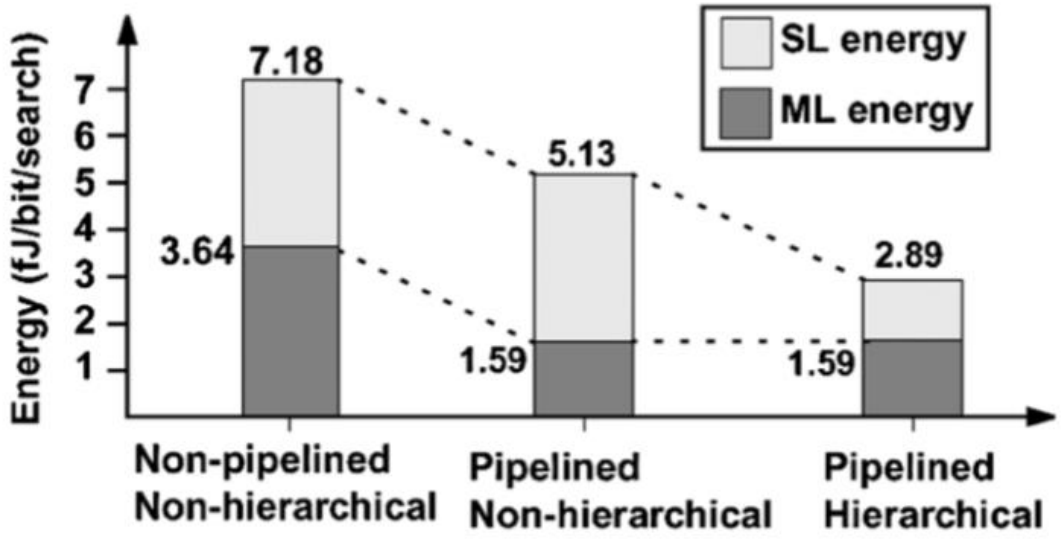
- Assumption:
  - Most ML segments miss in the first 8 Bit
- Expectation:
  - Power consumption reduced by 136/144 or 95%
- Result:
  - 1,59/3,64 or 56%
- Explanation:
  - Overhead of clocking the additional ML-Flip-Flops and repeated circuitry



Ref. 1

Simulation: Total power consumption

- Adding hierarchical searchlines:
  - 63% Reduction in SL power consumption
- Total power consumption reduced by 60%



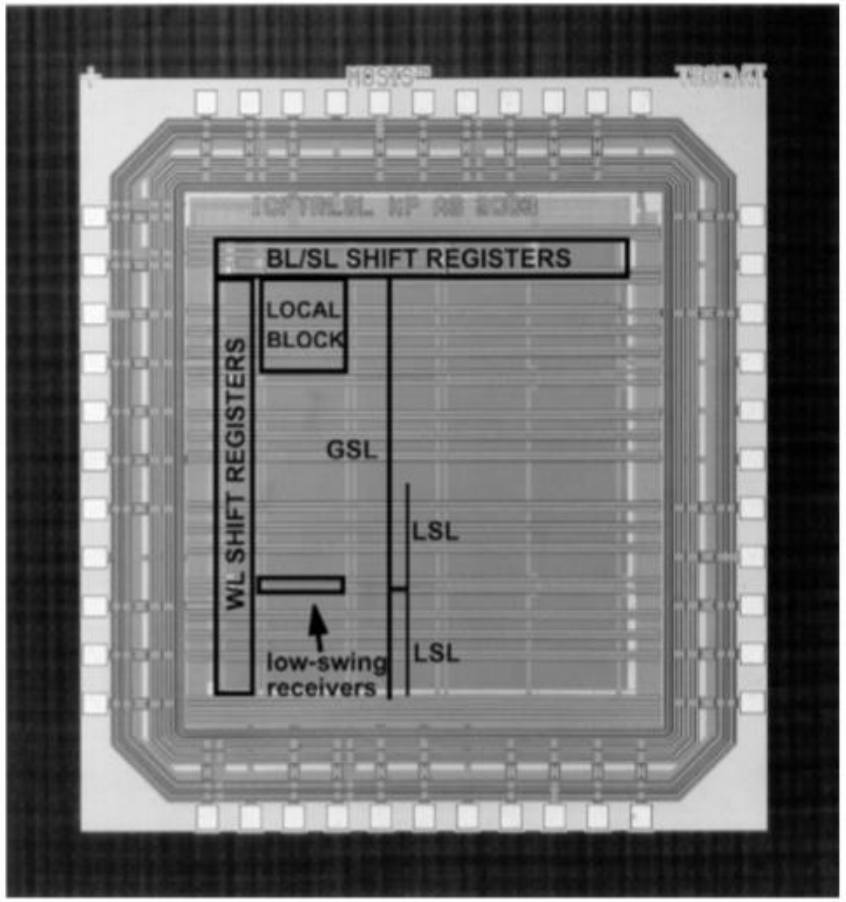
Ref. 1

Implementation

A real-world Test chip

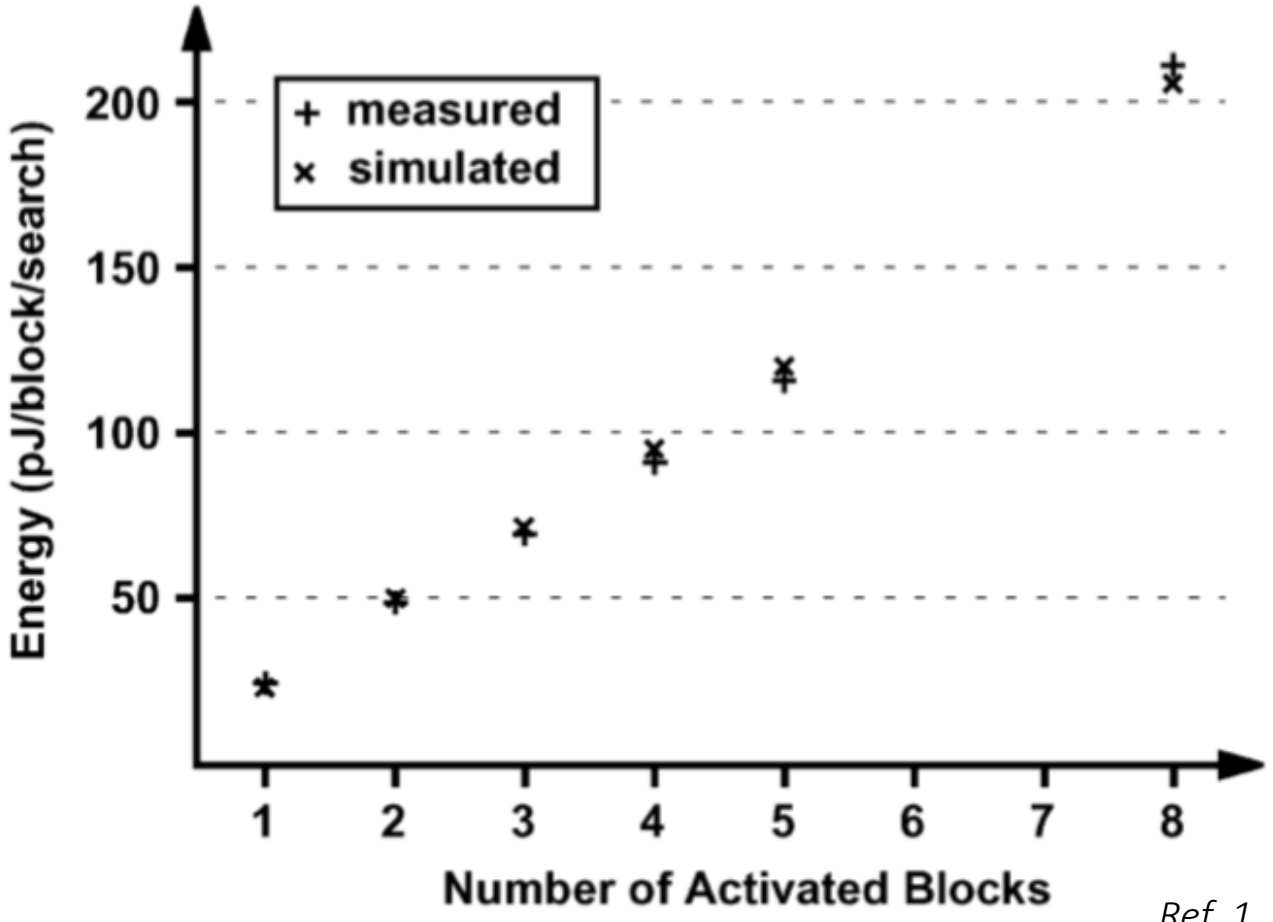
Test Chip

- VDD: 1,8V
- Process: 180nm
- Size: 2,3 x 2,1mm
- Cycle Time: 7ns
- 256 x 144 Bit CAM
  - 1x 8Bit, 4x 34Bit Segments
- Only two segments use hierarchical SL
  - Allows Direct Comparison of power consumption



Ref. 1

Simulation vs. Implementation



Ref. 1

Conclusion

Conclusion

- Presented Techniques to reduce power consumption:
    1. Pipelined matchlines
    2. Hierarchical searchlines
    3. Precharge low scheme
- Expected reduction of power consumption: ~60%
- Slightly increased area needs: ~6%
- Similar cycle times to conventional designs
- Pipelined architecture introduces additional latency
- Reduced Noise Immunity

References

- This Talk is based on:
  1. K. Pagiamtzis, A. Sheikholeslami, “A low-power content-addressable memory (CAM) using pipelined hierarchical search-scheme”, IEEE Journal of solid-state Circuits, 2004
  2. K. Pagiamtzis, A. Sheikholeslami, “Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey”, IEEE Journal of solid-state Circuits, 2006
- Figures and graphics:
  3. Android, incoming call, 2016 - http://i.stack.imgur.com/5eHW2.png