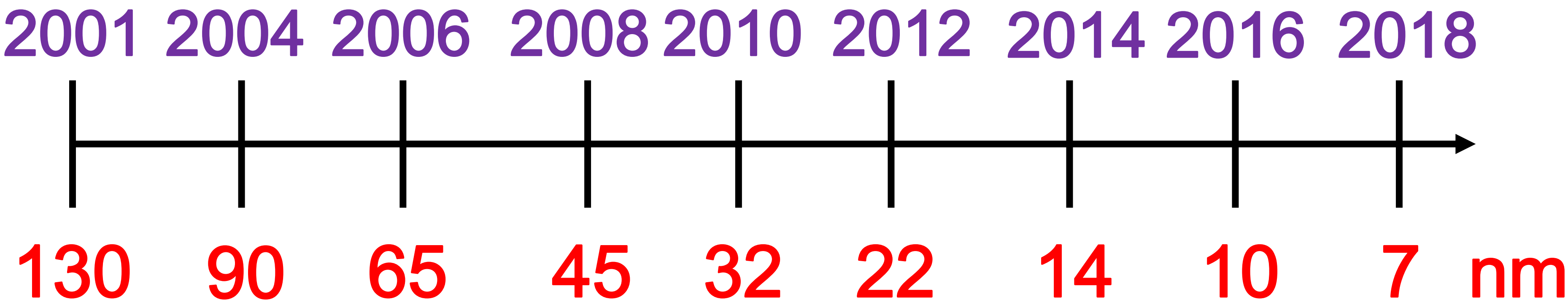
# Avoiding the Dark Ages with Memristors


Shahar Kvatinsky

Department of Electrical Engineering
Technion – Israel Institute of Technology

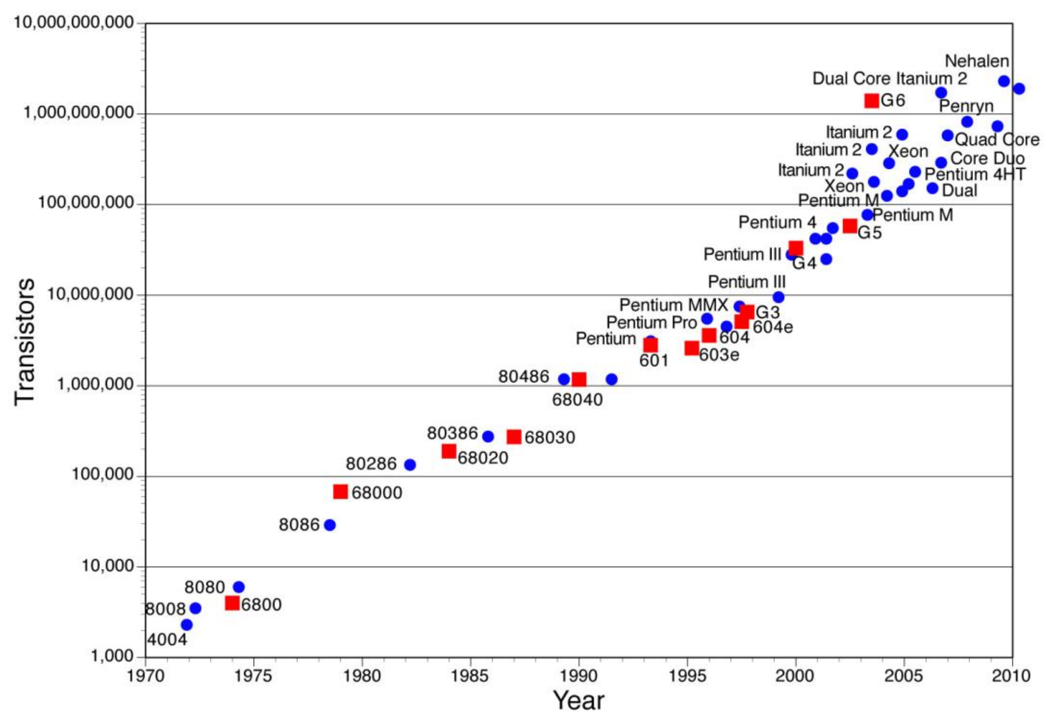

March 2016

# Scaling 101 – Moore's Law




$$S = \frac{45}{32} = \sim 1.4X$$

$$S^2 = \sim 2$$



# Scaling 101 – Dennard Scaling

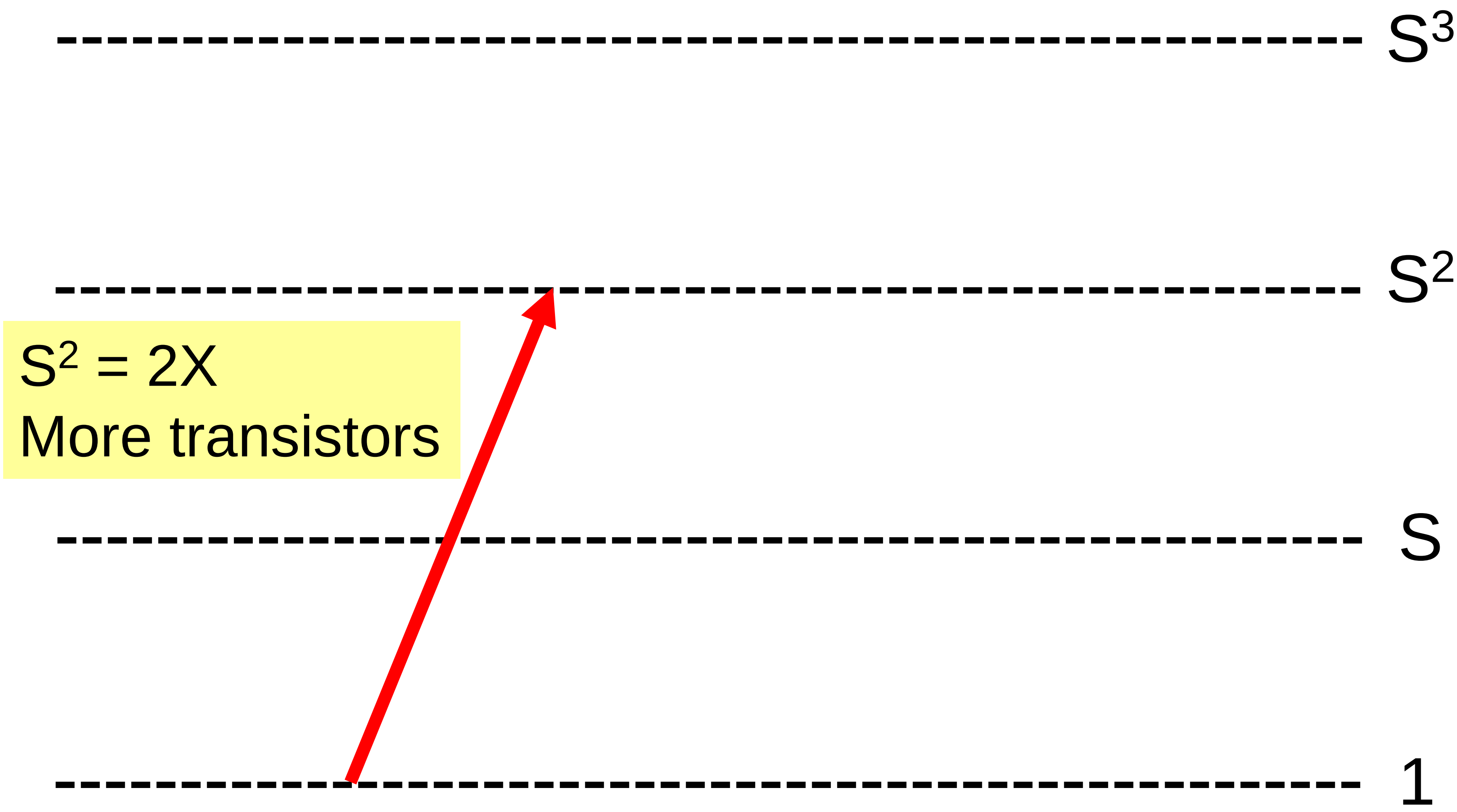
$S^3$

$S^2$

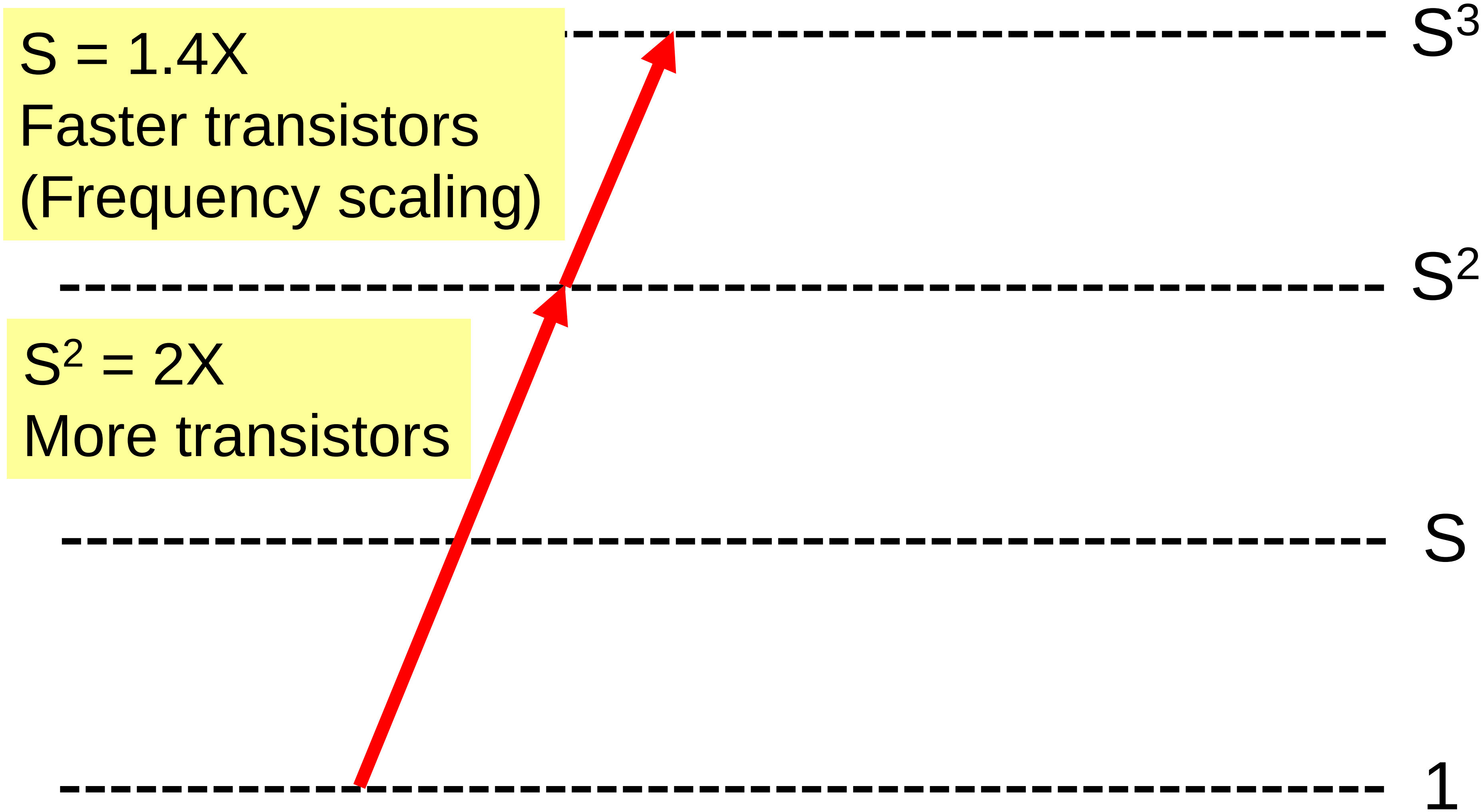
$S$

1

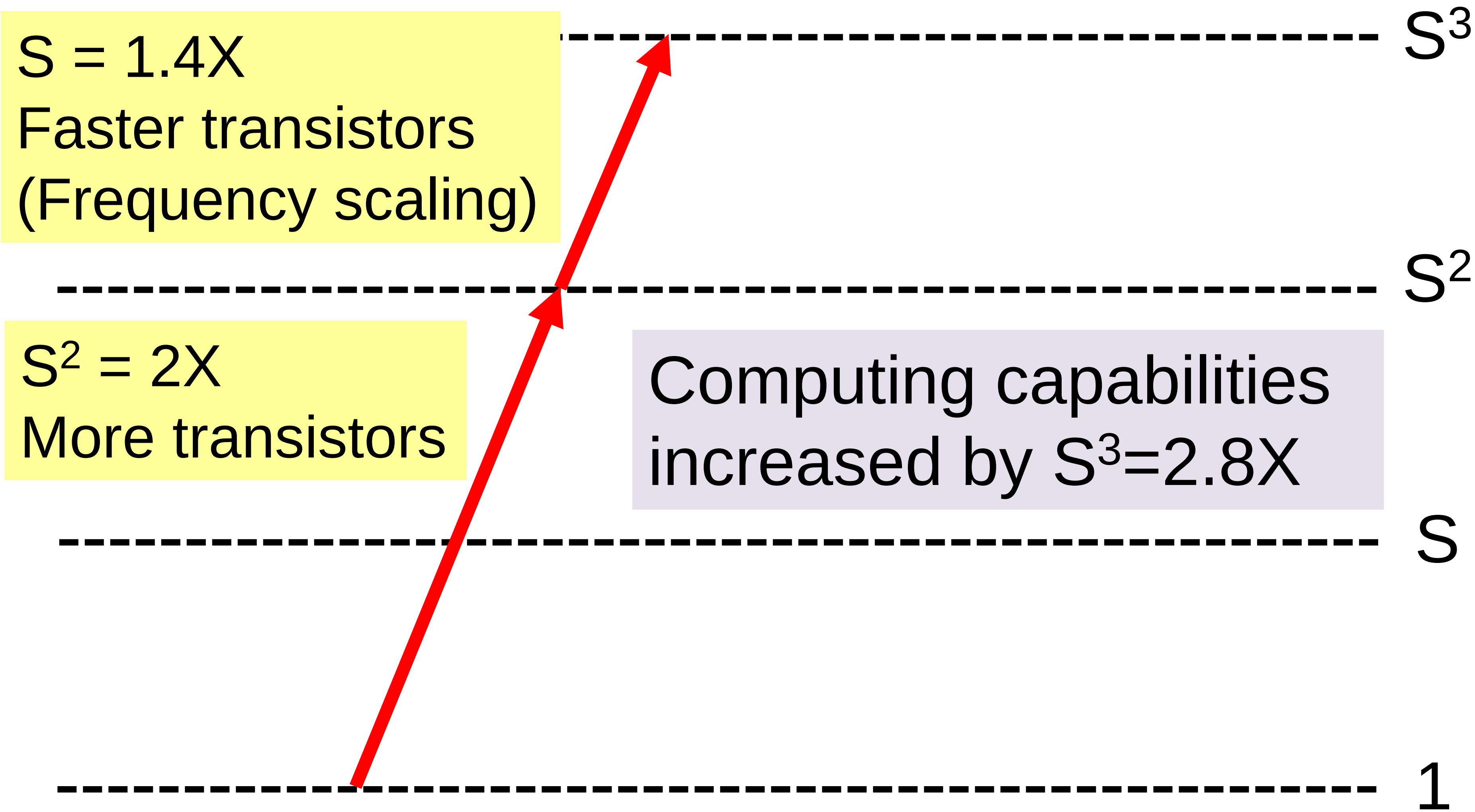
# Scaling 101 – Dennard Scaling



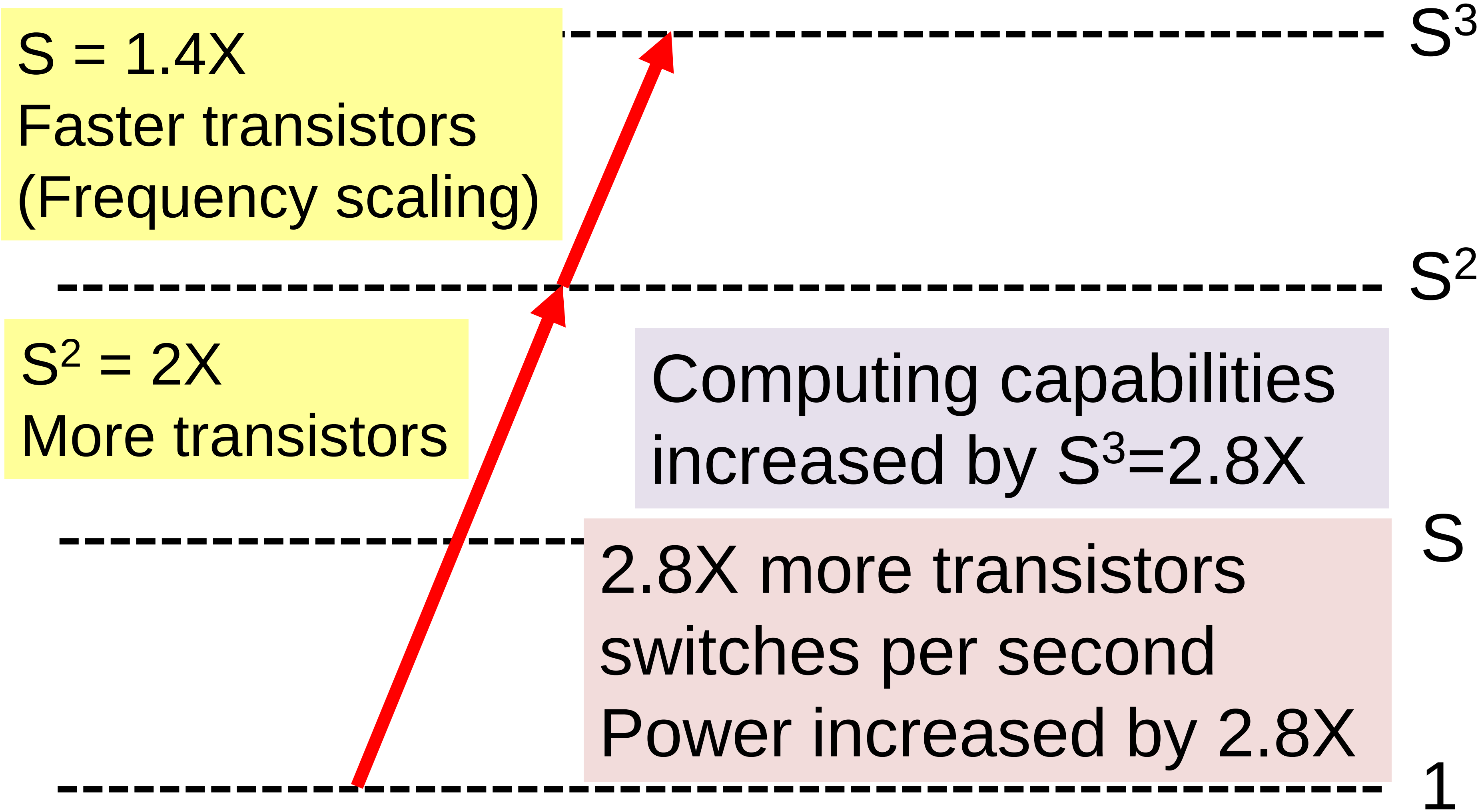
# Scaling 101 – Dennard Scaling



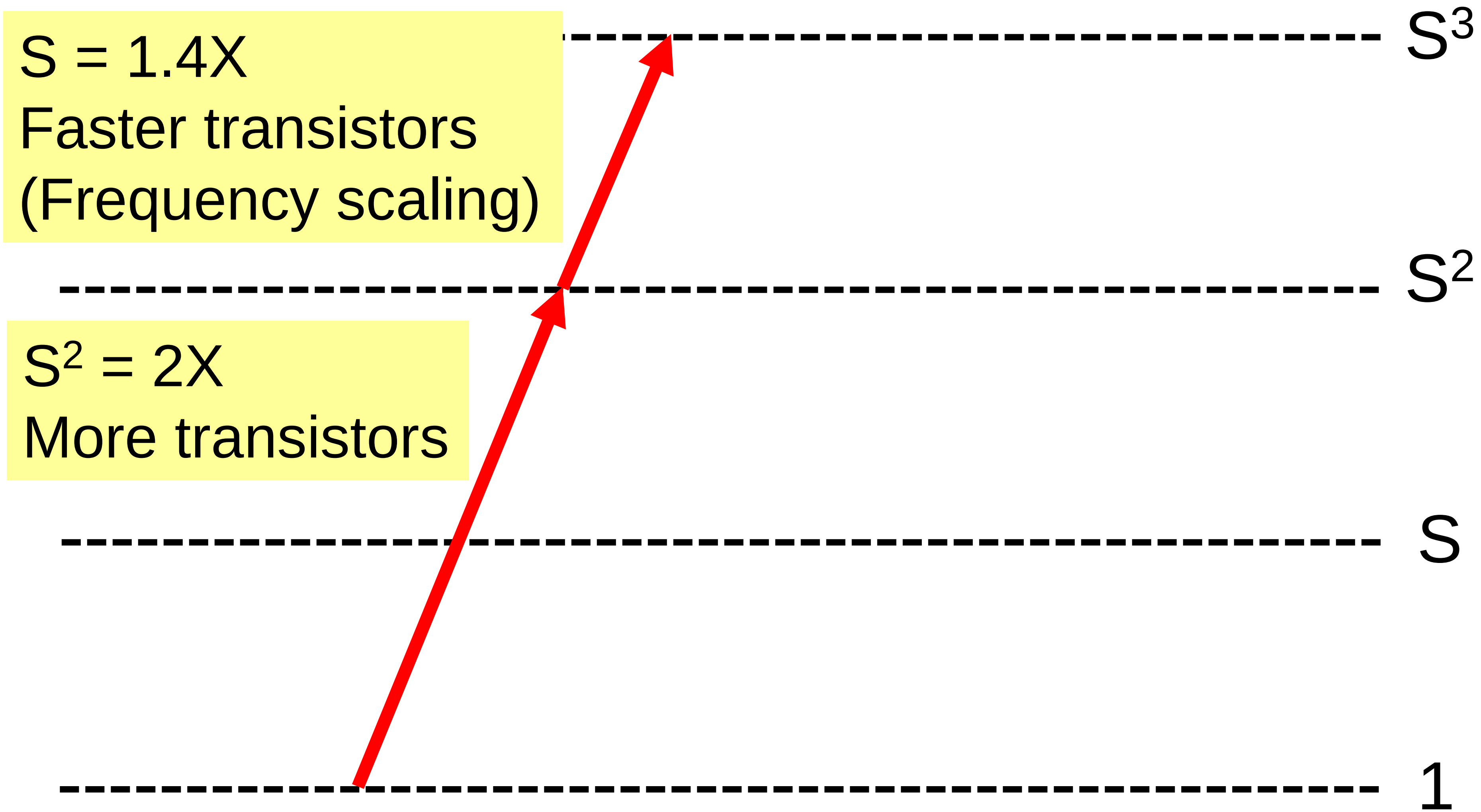
# Scaling 101 – Dennard Scaling



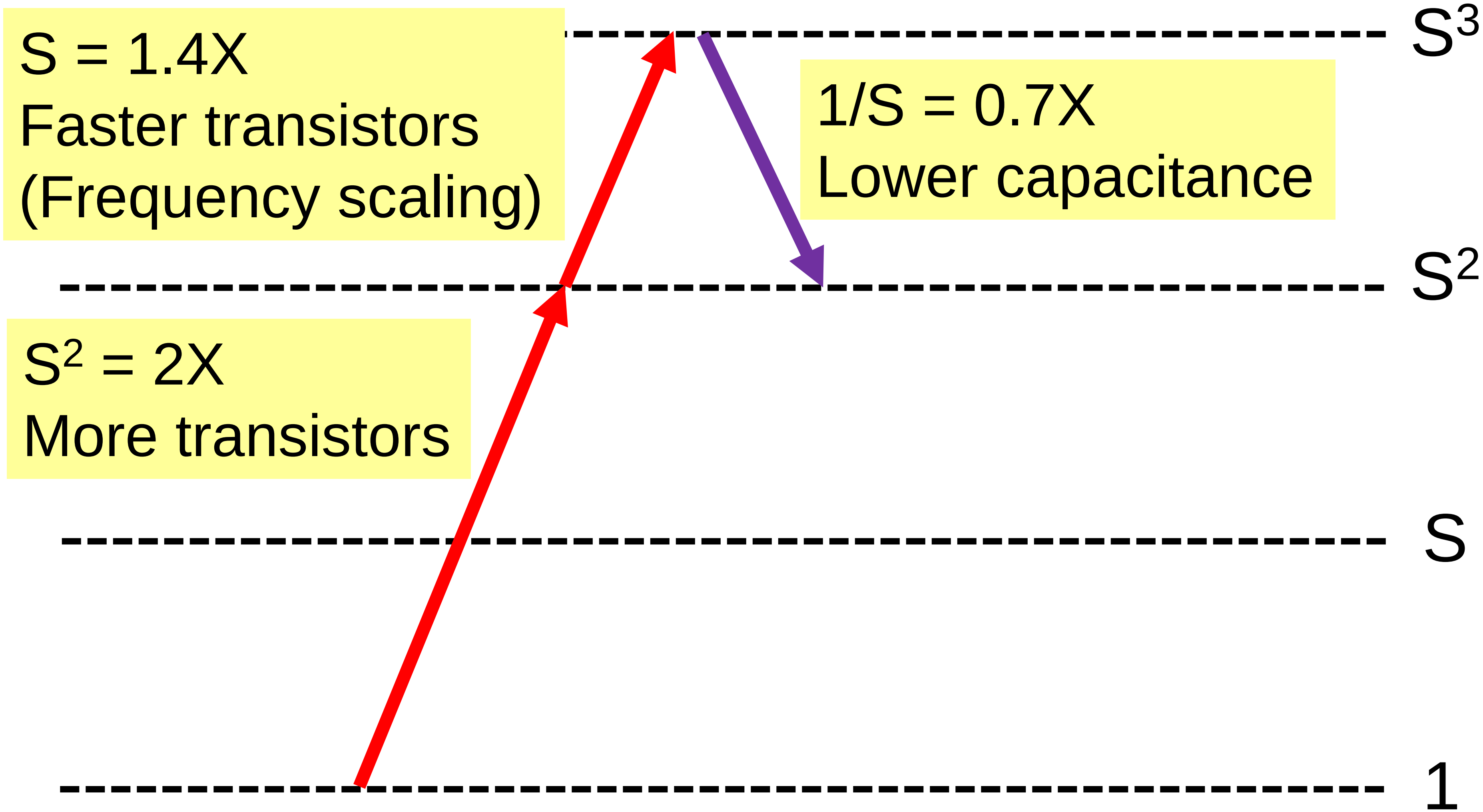
# Scaling 101 – Dennard Scaling



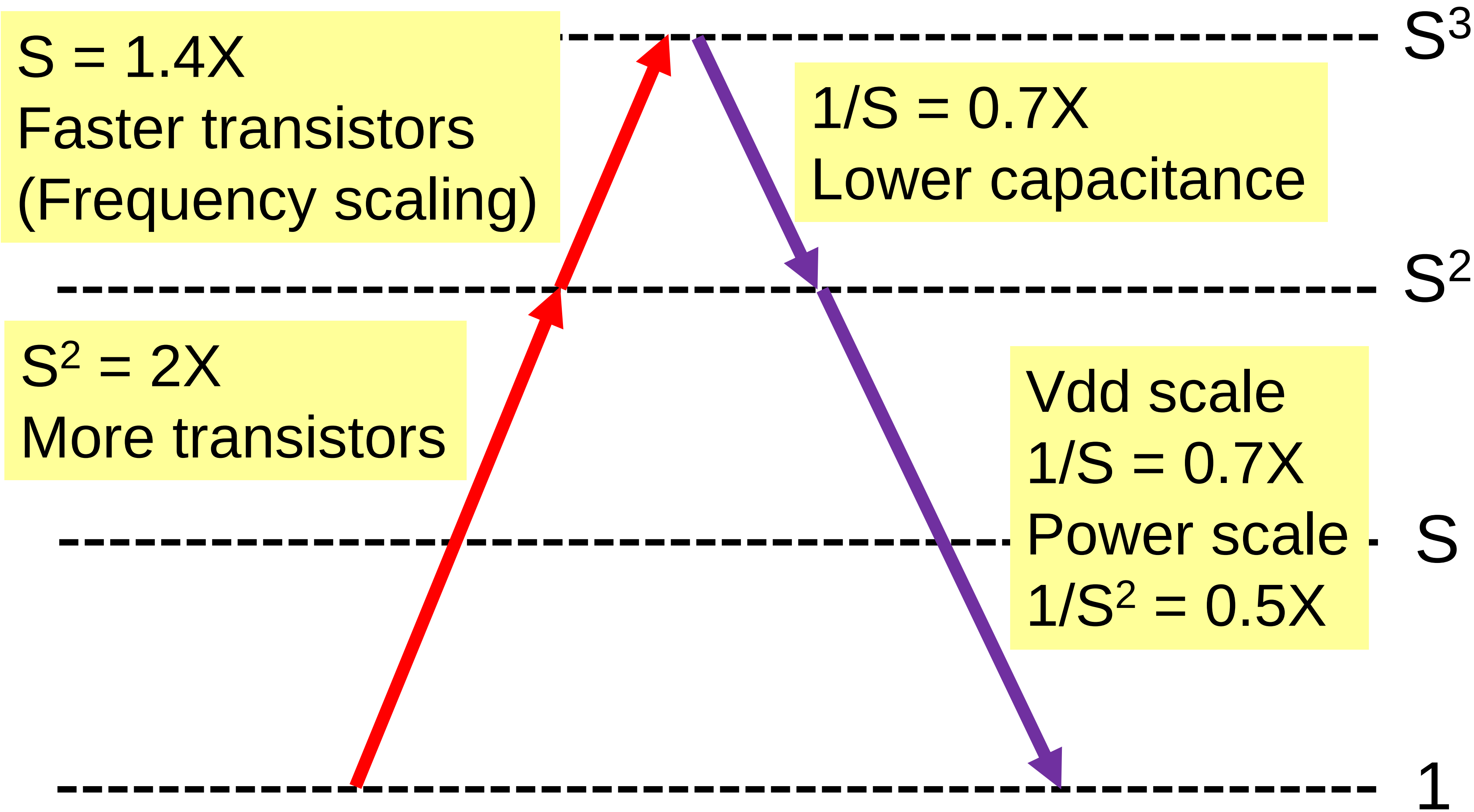
# Scaling 101 – Dennard Scaling



# Scaling 101 – Dennard Scaling

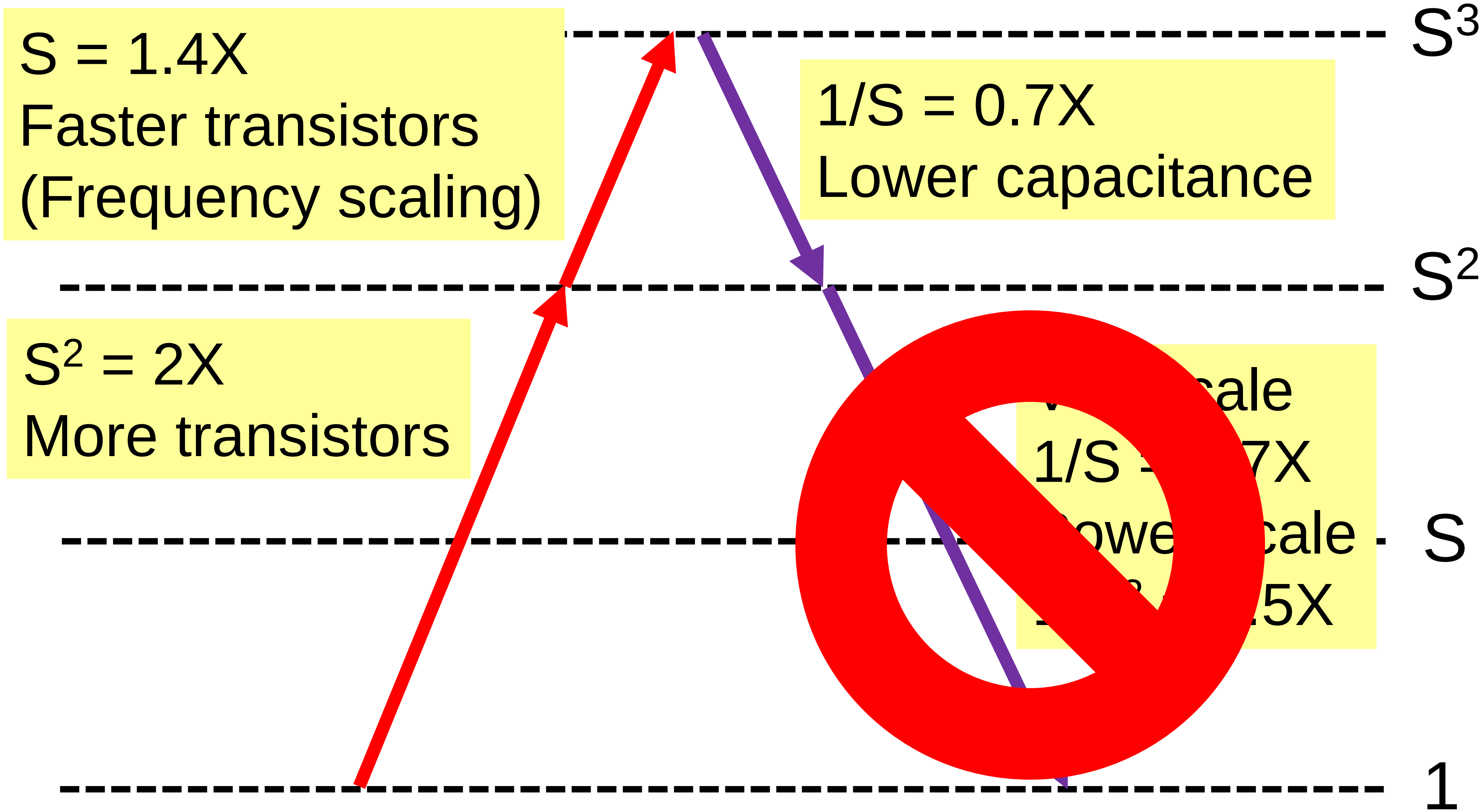


# Scaling 101 – Dennard Scaling

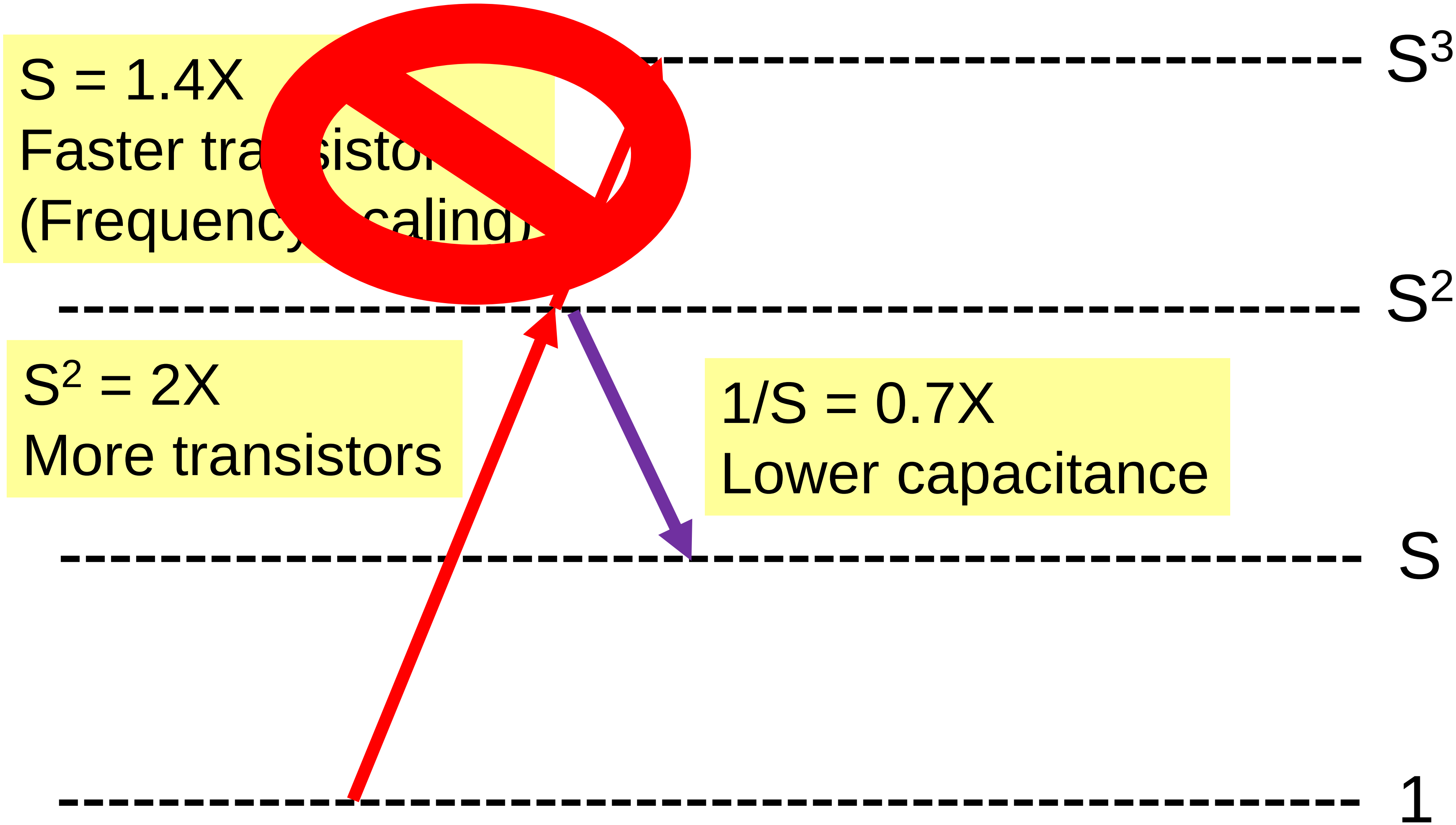


# 2005 The End of Dennard Scaling

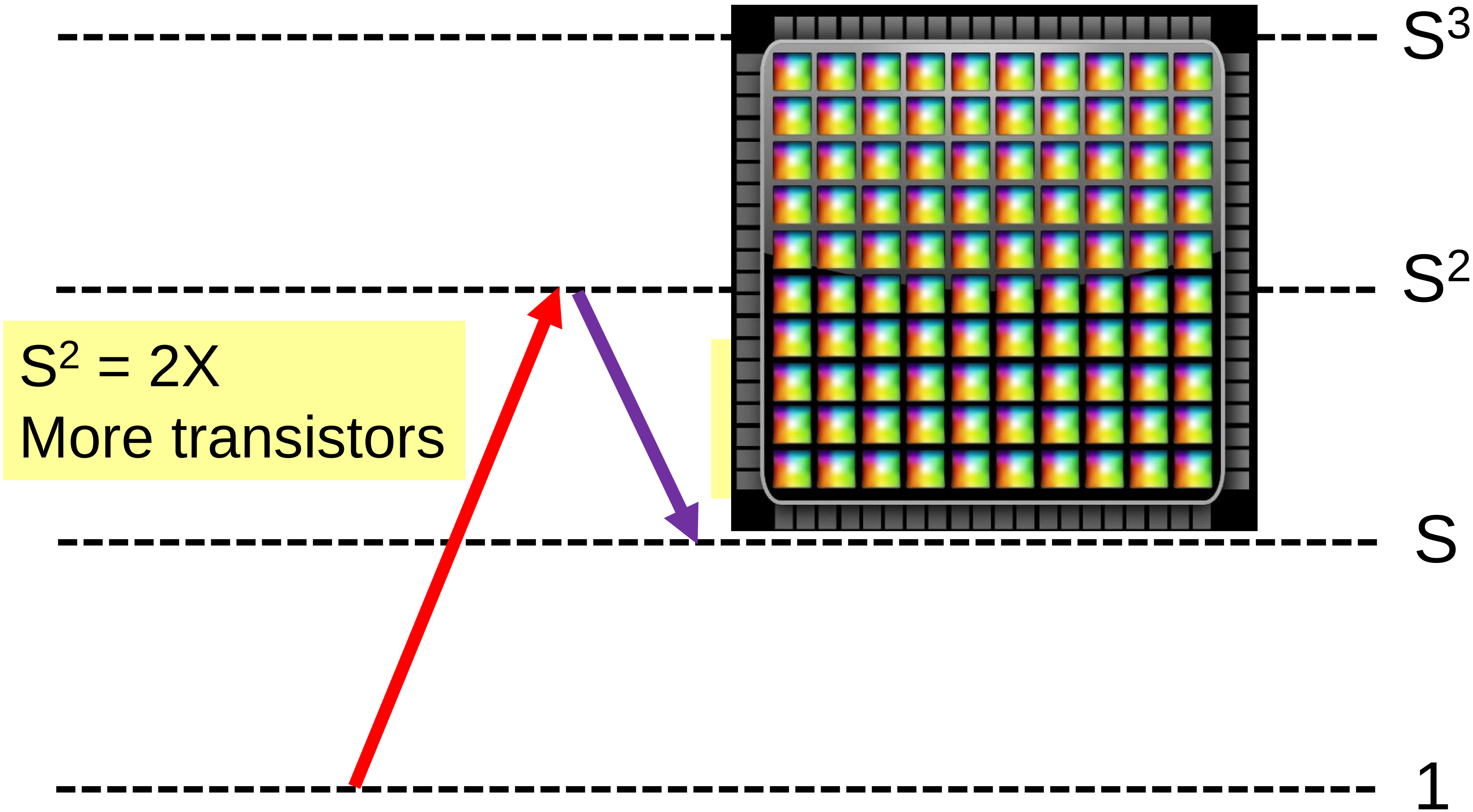
## Threshold Scaling and Leakage



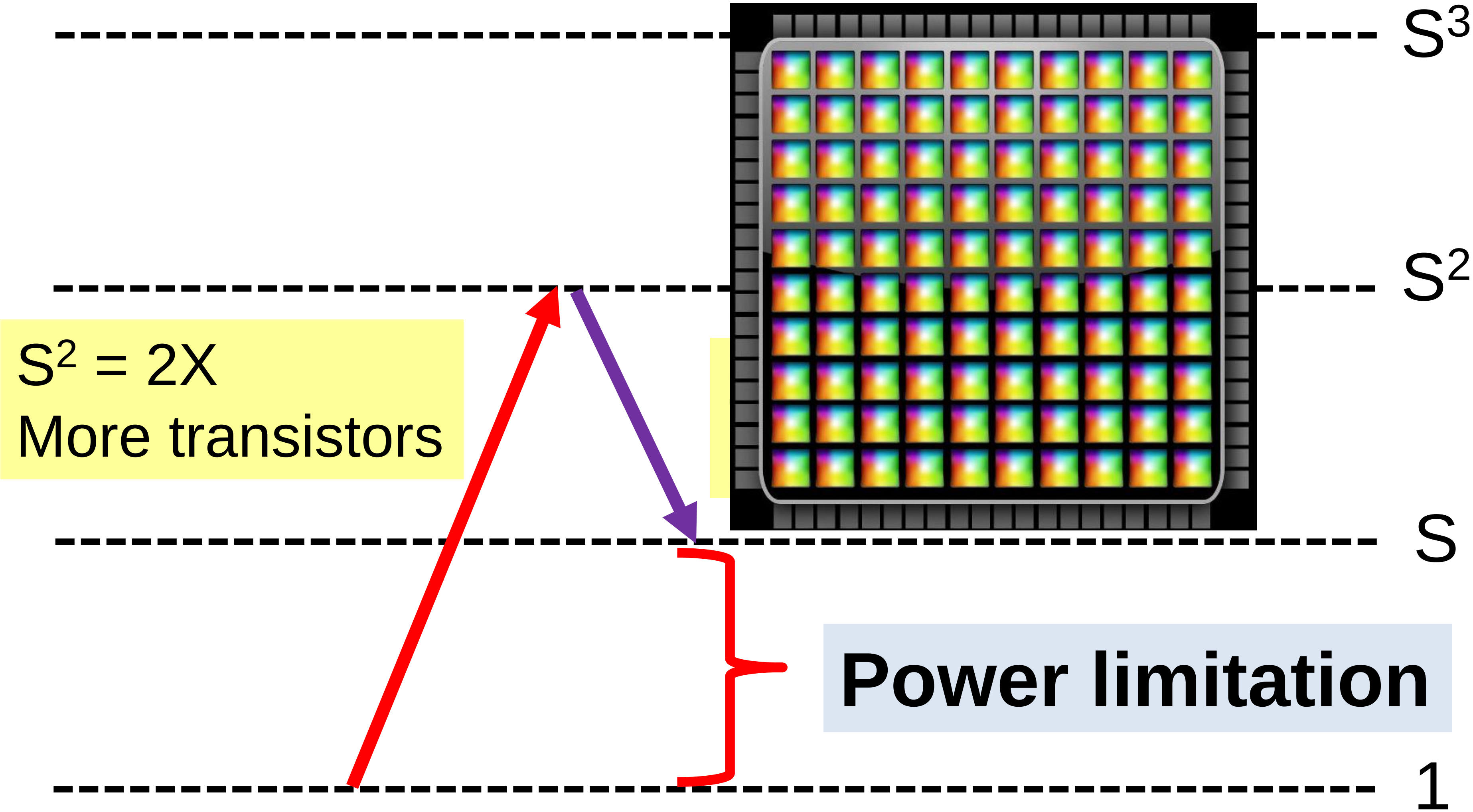
# The End of Frequency Scaling



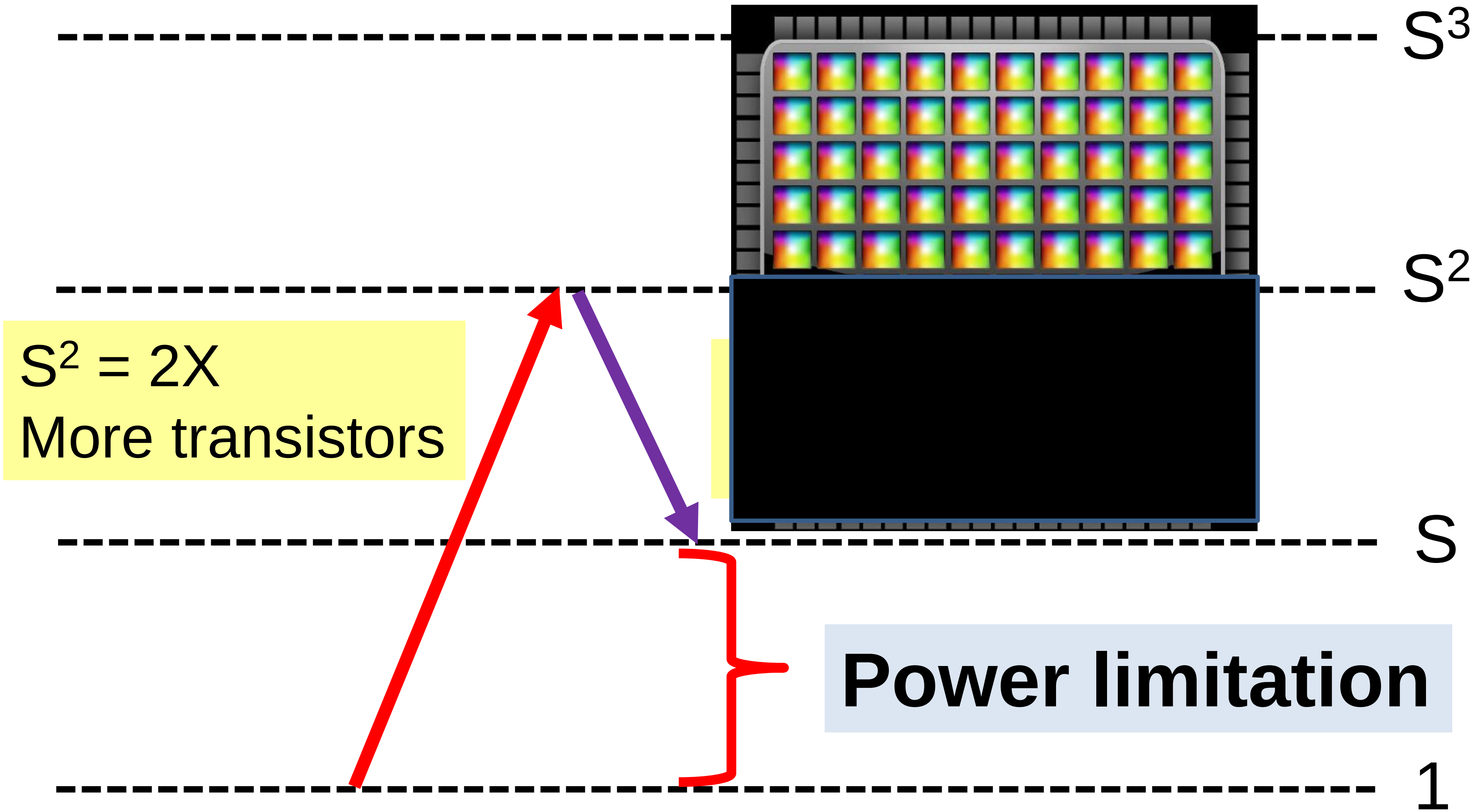
# Moving to Multicore



# Dark Silicon



# Dark Silicon



# The Four Horsemen of Dark Silicon

**Taylor DAC 2012**

- Shrink
- Dim
- Specialize
- Technology magic

  (*Deus Ex Machina*)

# The Four Horsemen of Dark Silicon

**Taylor DAC 2012**

- Shrink
- Dim
- **Specialize**
- Technology magic

  (*Deus Ex Machina*)

# Sources of Energy Inefficiency

| Operation (16-bit operand) | Energy/Op (45 nm) | Cost (vs. Add) |
|---|---|---|
| Add operation | 0.18 pJ | 1X |
| Load from on-chip SRAM | 11 pJ | 61X |
| **Send to off-chip DRAM** | **640 pJ** | **3,556X** |


M. Horowitz, "Computing's Energy Problem (and what we can do about it)," ISSCC Keynote 2014
A. Pedram, S. Galal, S. Richardson, S. Kvatinsky, and M. Horowitz, "Dark Memory and Accelerator-Rich System Optimization in the Dark Silicon Era," IEEE Design & Test (submitted)

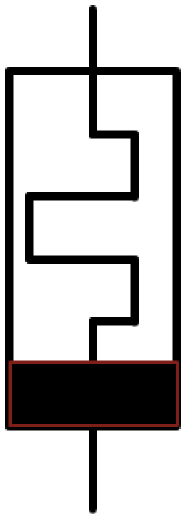
# Dark Memory and Specialization

- Memory system contributes >50% system power
- Memory hierarchy does not solve everything, SRAM is never completely dark
- Specialization increases memory power portion
- Amdahl's law - need to dim memory


A. Pedram, S. Galal, S. Richardson, S. Kvatinsky, and M. Horowitz, "Dark Memory and Accelerator-Rich System Optimization in the Dark Silicon Era," IEEE Design & Test (submitted)

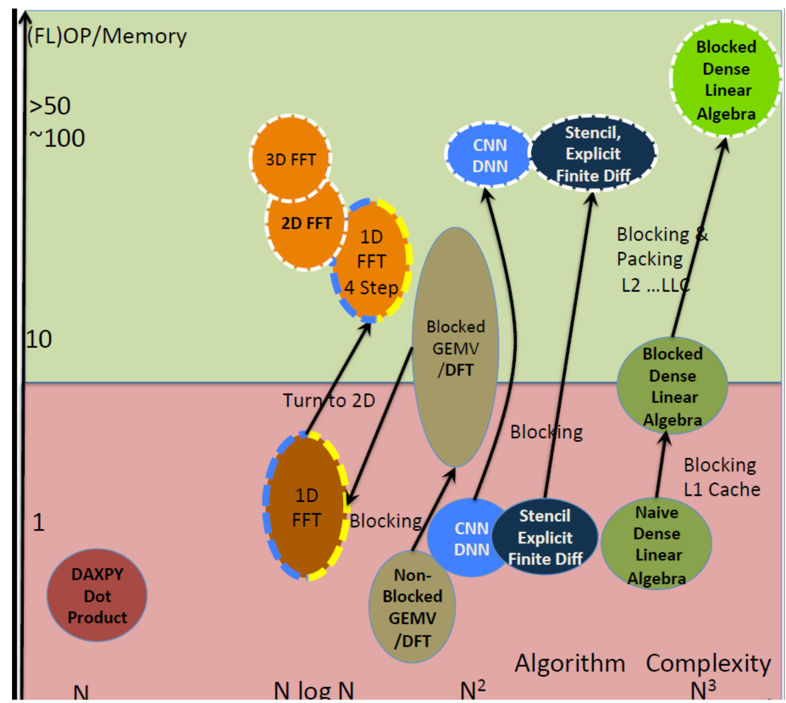
# Will Memristors Light the (Dark) Memory?

- Nonvolatility – low static energy
- Dense memory – short wires
- Still large memory -> relatively long wires, not a fundamental change in energy



# Fundamental Solution – SW-HW

- Minimizing memory accesses – algorithm execution
- High chip-level locality
- Memristive accelerators can help






A. Pedram, S. Galal, S. Richardson, S. Kvatinsky, and M. Horowitz, "Dark Memory and Accelerator-Rich System Optimization in the Dark Silicon Era," IEEE Design & Test (submitted)

# Memristive Accelerators

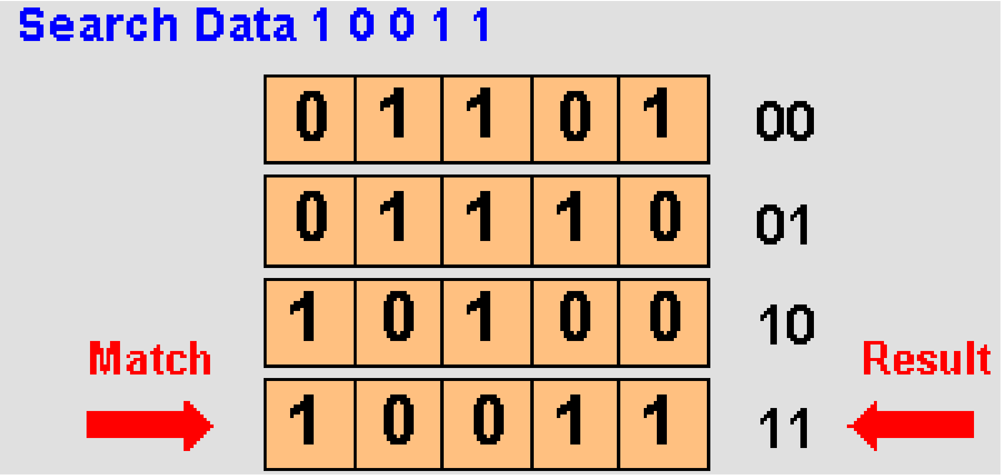
- Resistive Associative Processor (ReAP, Yavits et al. CAL 2015)
- Resistive GP-SIMD (Morad et al., TACO 2016)
- Neuromorphic (Soudry et al. TNNLS 2015)
- Memory Processing Unit (MPU, Kvatinsky et al. TVLSI 2014, TCAS II 2014, Levy et al. MEJ 2014)

# Memristive Accelerators

- **Resistive Associative Processor**
  (ReAP, Yavits et al. CAL 2015)
- Resistive GP-SIMD (Morad et al., TACO 2016)
- Neuromorphic (Soudry et al. TNNLS 2015)
- Memory Processing Unit (MPU, Kvatinsky et al. TVLSI 2014, TCAS II 2014, Levy et al. MEJ 2014)

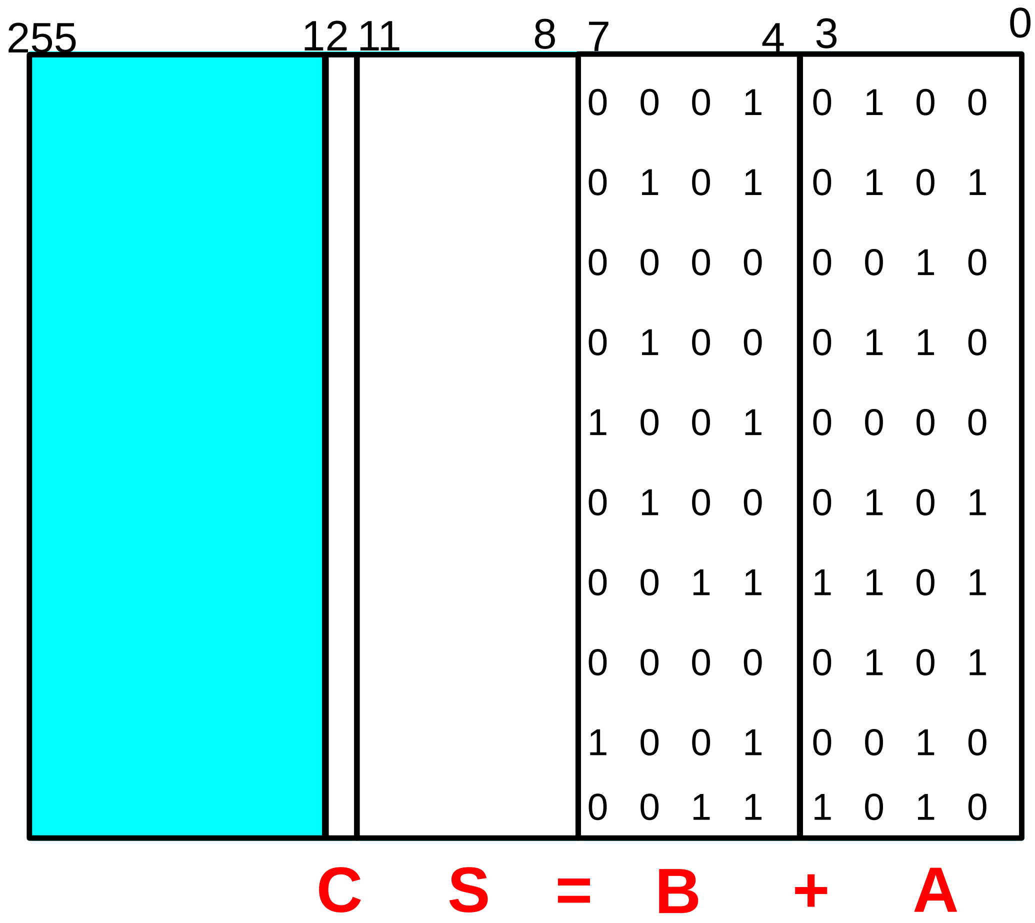
# Associative Processor

- Processing in-memory (PiM), using CAM
- AP is similar to a look-up table
- Computation is a series of "compare" and "write" operation



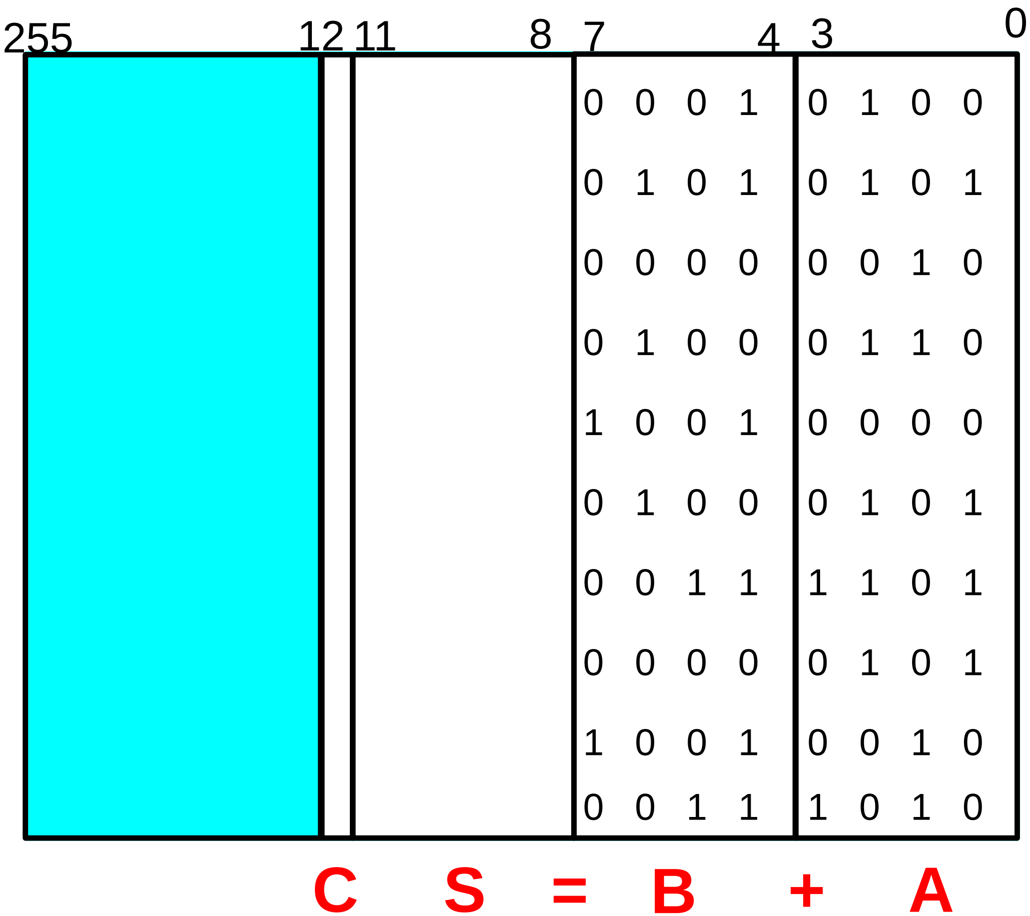
# Example: Associative Vector Addition

## ASSOCIATIVE PROCESSOR: MEMORY MAP



# Example: Associative Vector Addition

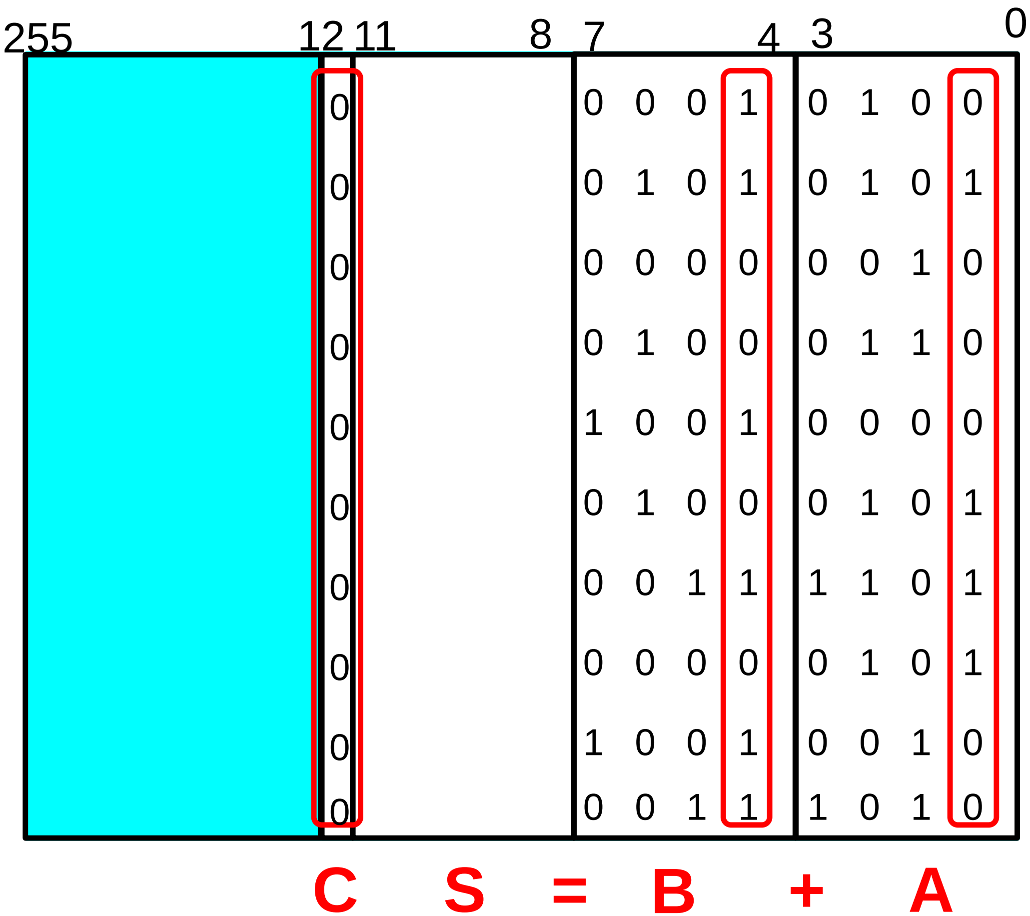
## ASSOCIATIVE PROCESSOR: MEMORY MAP




| cout | s | $c_{in}$ | a | b |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

# Example: Associative Vector Addition

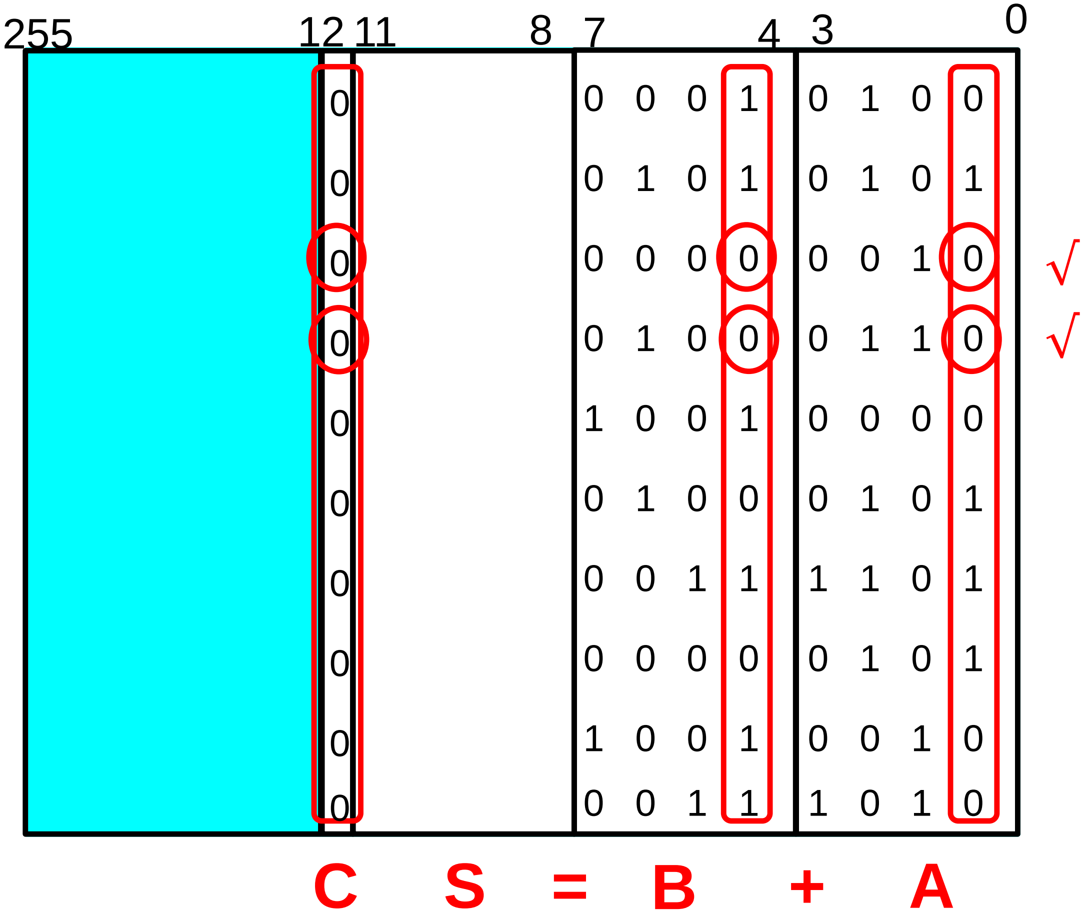
## SELECTING BIT COLUMN 0




| cout | s | $c_{in}$ | a | b |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

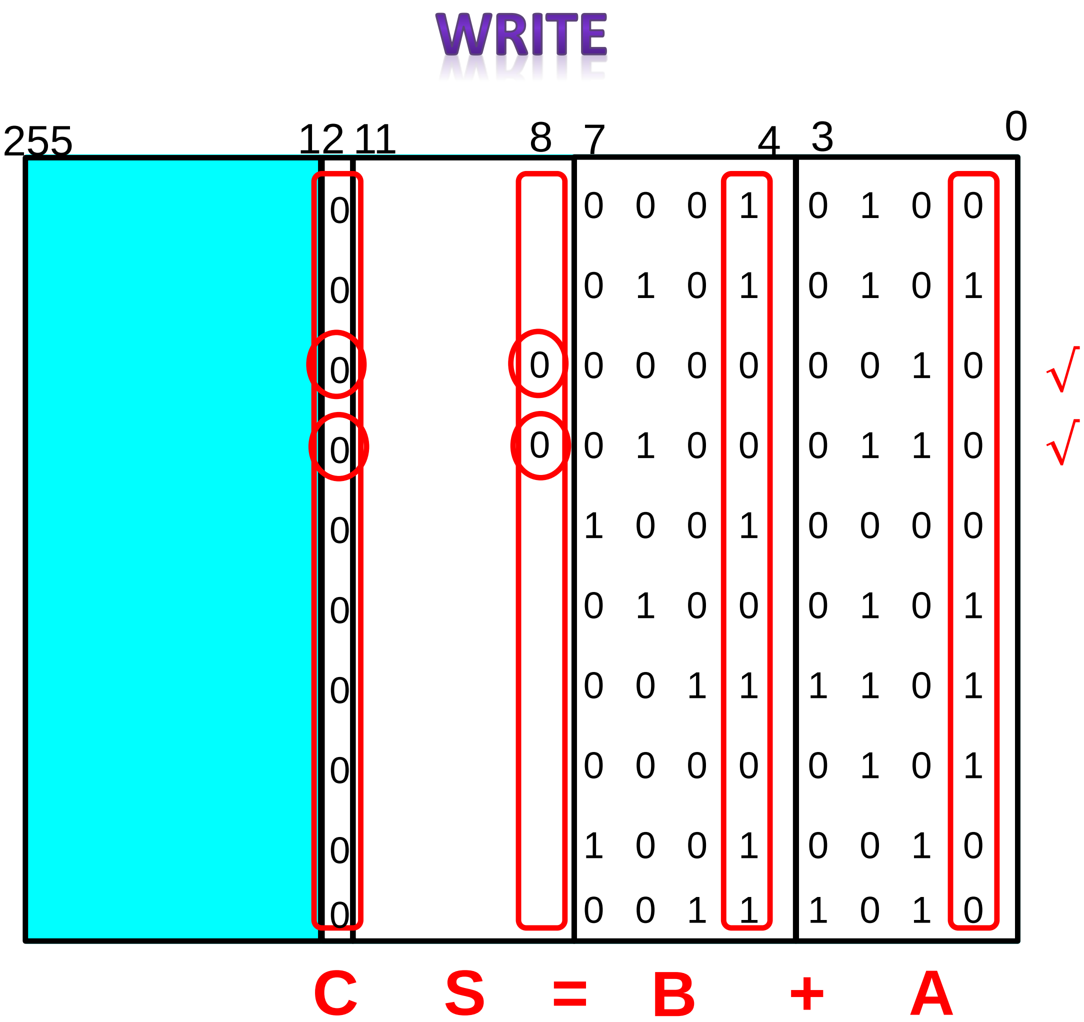
# Example: Associative Vector Addition

COMPARE




| cout | s | $c_{in}$ | a | b |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

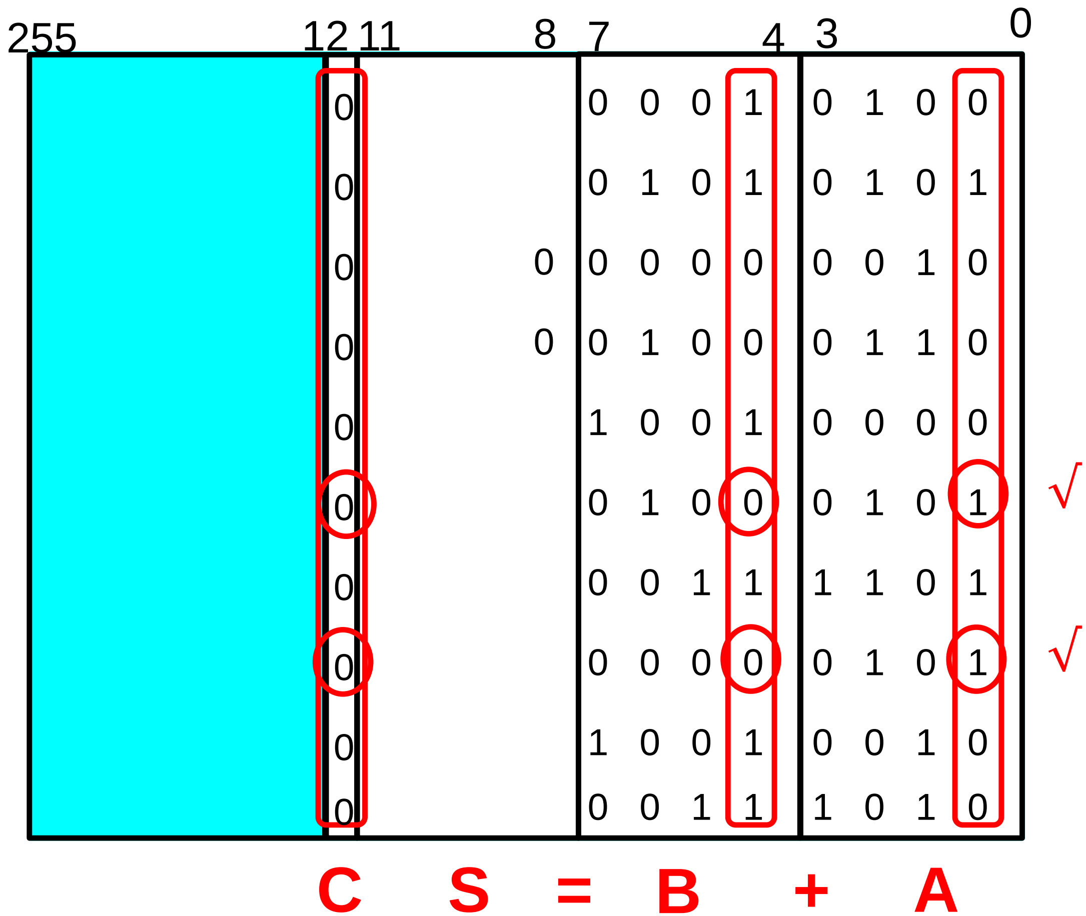
# Example: Associative Vector Addition




| cout | s | $c_{in}$ | a | b |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

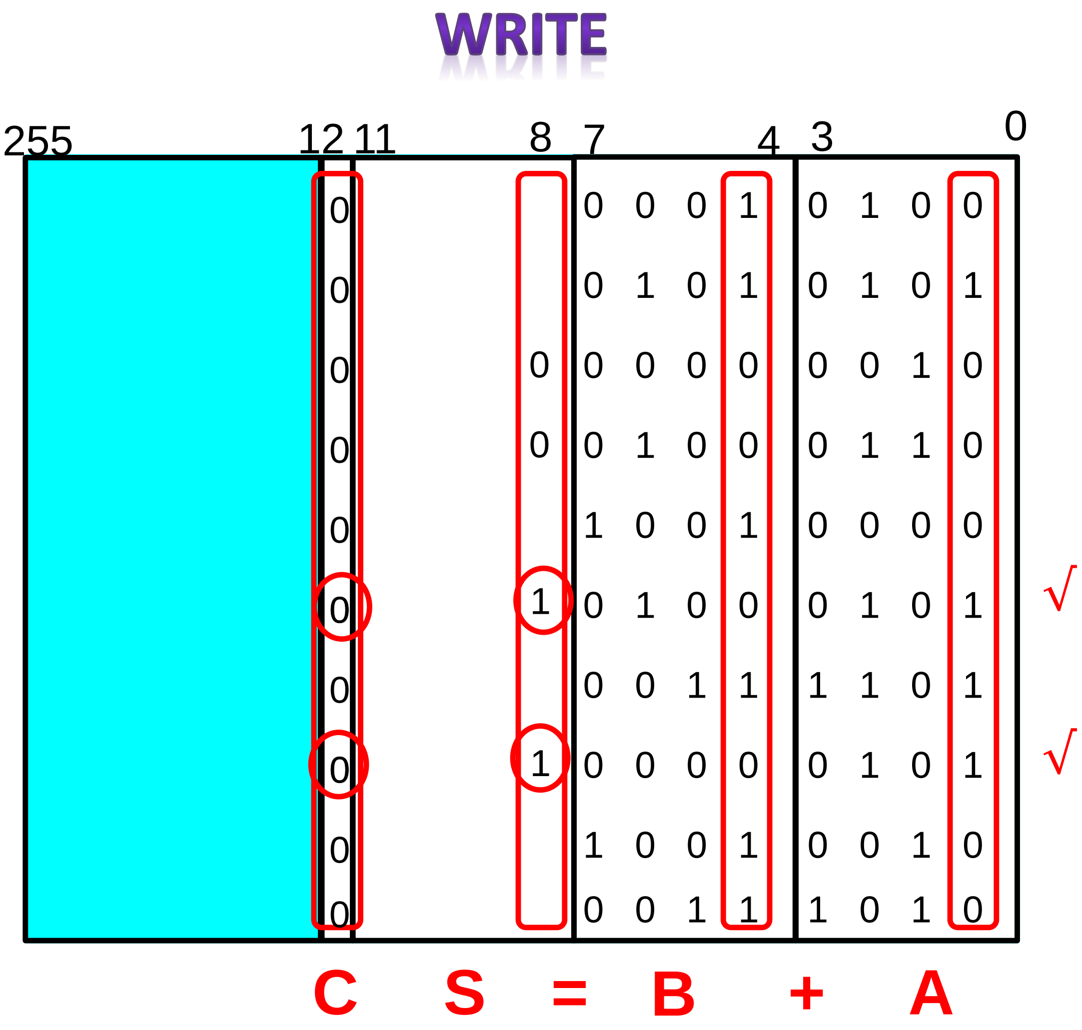
# Example: Associative Vector Addition

COMPARE




| cout | s | $c_{in}$ | a | b |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

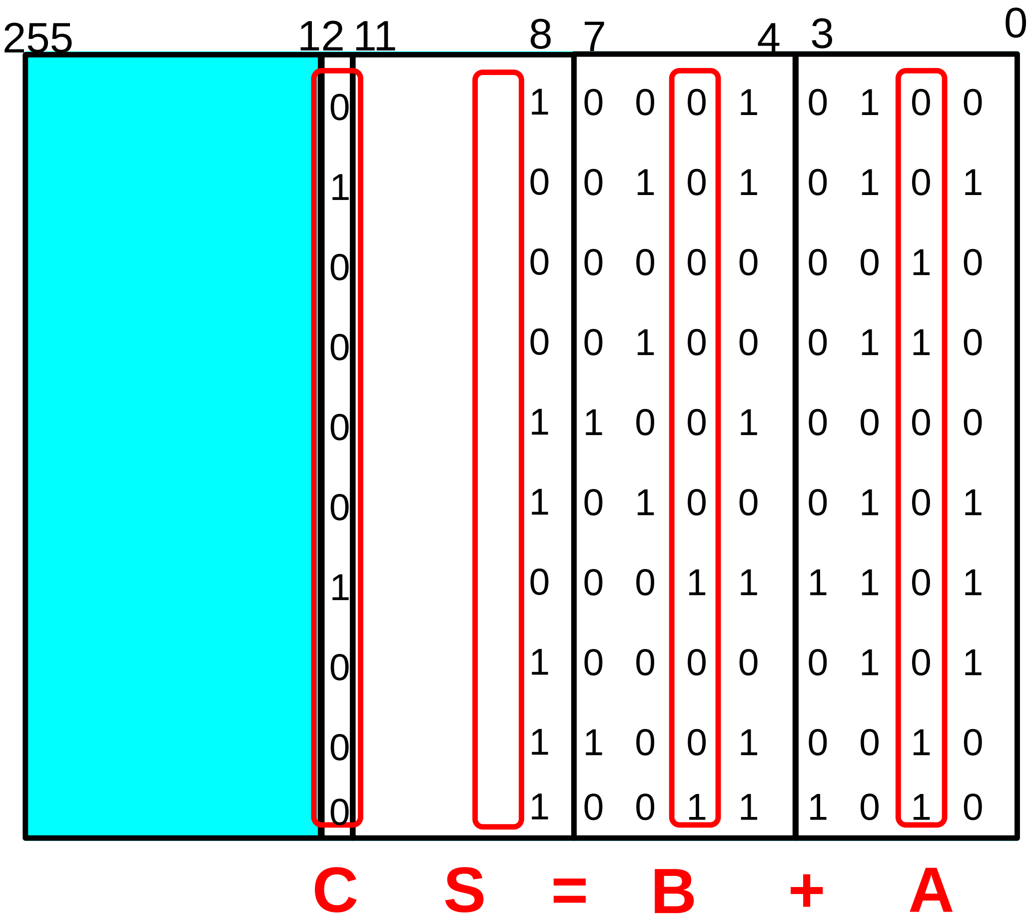
# Example: Associative Vector Addition




| cout | s | $c_{in}$ | a | b |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

# Example: Associative Vector Addition

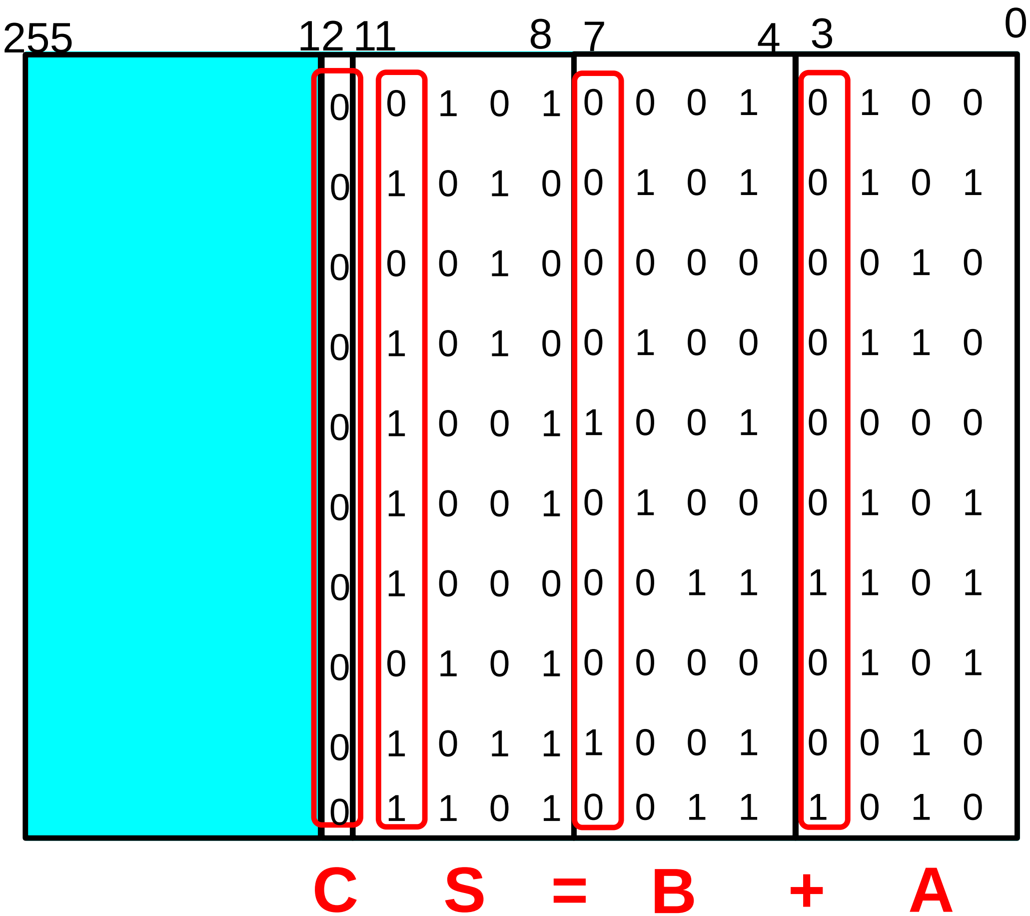
## SELECTING BIT COLUMN 1




| cout | s | $c_{in}$ | a | b |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

# Example: Associative Vector Addition

## END OF COMPUTATION

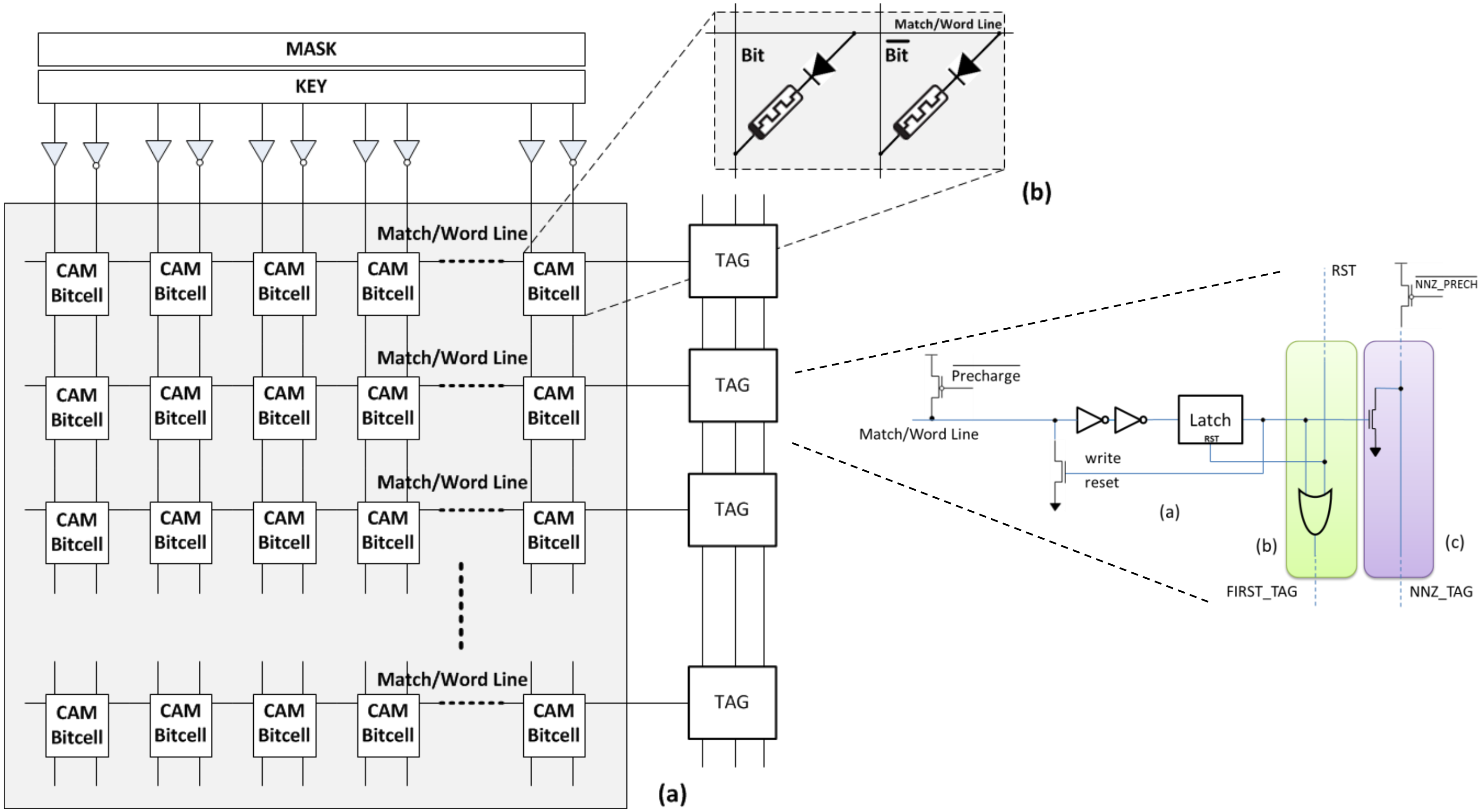



| cout | s | $c_{in}$ | a | b |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

# AP Complexity

- Arithmetic:
  - Fixed point
    - $m$ bit add / sub: $O(m)$ cycles
    - $m$ bit mult/div: $O(m^2)$ cycles
- Pattern match: $O(1)$ cycles
- Finding max/min: $O(1)$ cycles
- Independent of the dataset size:
  The larger the problem, the better the performance of the Associative Processor!

# Resistive Associative Processor

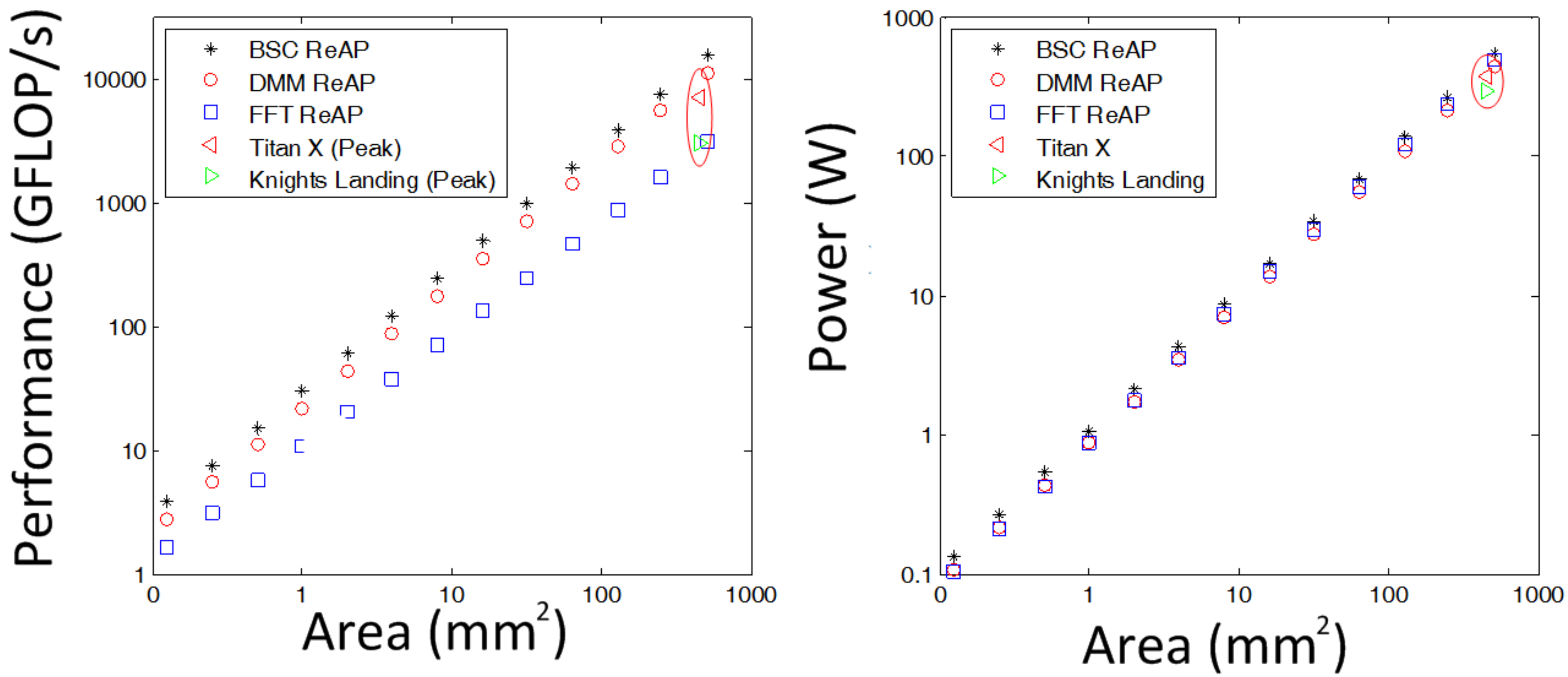



- Converting a memory crossbar into a massively parallel SIMD processor
  - Enabling a 100M PU-AP

# What AP is Good for

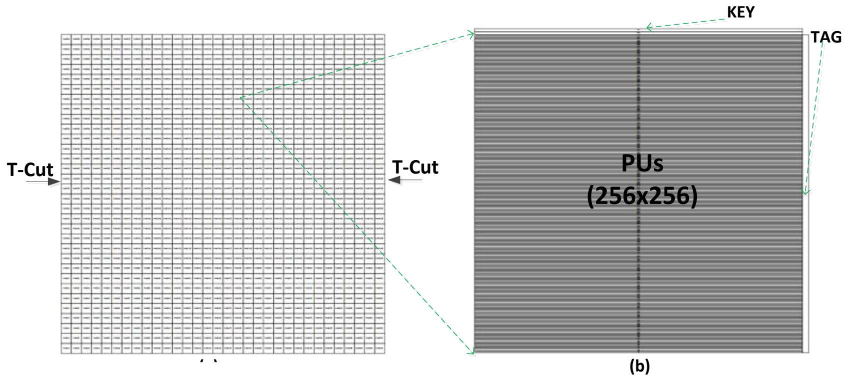
- Dense and sparse linear algebra
- K-means clustering
- Linear SVM classification
- FFT, convolution, feature extraction
- Sequence alignment (Smith-Waterman)
- Graph processing (Dijkstra's shortest path finding)

# Performance and Power Consumption

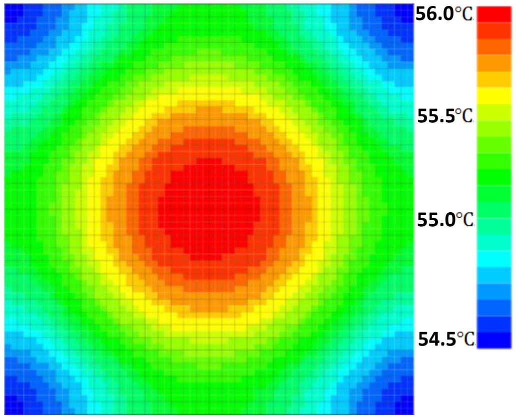



- ReAP size (and consequently performance) are constrained by memristor write energy
- Max Dense Matrix Multiplication performance is 5TFLOPS under this constraint

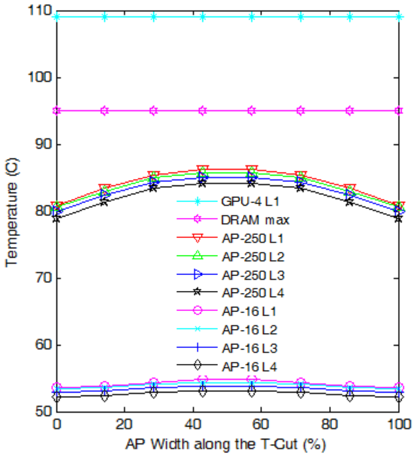
# Thermal View




ReAP Floorplan




ReAP Thermal Map

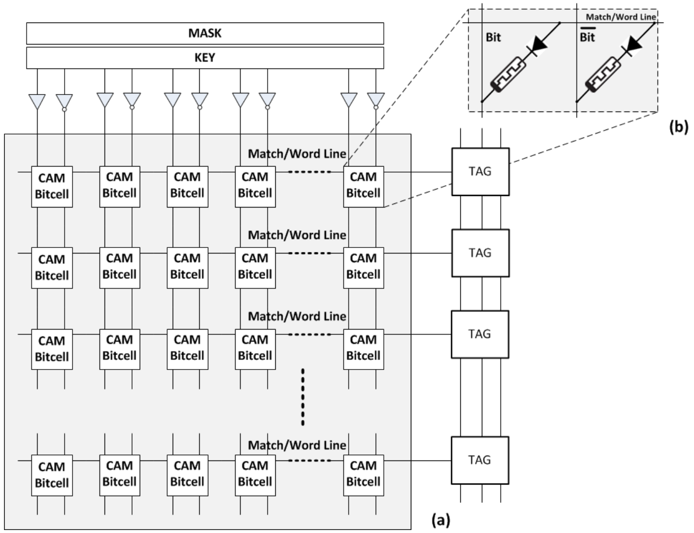



ReAP Temperature (vs. DRAM)

- Temperature and hot spots are the reason 3D integration of CPUs and DRAM is stalling
- AP does not have this problem due to its (almost) uniform thermal distribution

# Summary

- The dark (silicon and memory) age
  - Main source of inefficiency is data movement
- The solution: accelerators and HW-SW awareness
- Memristive accelerators!



# Thanks!

shahar@ee.technion.ac.il