Agenda

➢ Why is “low power” important?
➢ How does cache contribute to the power consumption of a processor?
➢ What are some design challenges for low power caches?
➢ Existing solutions for low power cache design
➢ Researched ultra-low-voltage cache design solution for many-core systems
  ○ Detailed architectural overview of the design
  ○ Performance comparison with existing solutions
Background

The **cost/performance ratio** of computing systems have seen a steady decline due to advances in:
- Integrated circuit technology
- Architectural improvements in CPU design

At the same time, more and more emphasis is given to the minimization of environmental footprint of computing systems. In particular, **energy efficiency** is a key concern in future many-core (1000+ processors) system design.
Among all processor components, the cache consumes about 40% of the total area and 20% of power.

Supply voltage scaling can reduce system energy consumption at the cost of performance and reliability. (Especially with the current trend toward many-core processors and deep nanometer designs)
Existing Solutions

- Cache designs that intrinsically consume less energy.
- Employ advanced fault tolerance techniques to maintain reliability under low supply voltage.
Existing Solutions (cont.)

- **Redundant Memory**
  Excessive area and power overheads when cell-failure probability increases

- **Cache Line Disabling**
  This is only effective if the cell-failure probability is low

- **ECC (error correcting codes)**
  E.g. extended hamming codes, multi-bit clustered ECC
Cache Fault Analysis

- **Soft cache fault** is generally caused by random noises or particle strikes. It occurs transiently and is recoverable through error-detecting/correcting codes.

- **Hard cache fault** is persistent memory cell fault that occurs as a result of chip manufacturing defects or device aging. Error correction codes can also be used to correct this failure but can only correct single-bit errors.
Cache Fault Analysis (cont.)

Fig. 1. SRAM cell failure probability VS. $V_{DD}$. 
Cache Fault Analysis (cont.)

Fig. 2. Distribution of the number of persistent faulty cells in a cache line: (a) analysis and (b) simulation.
Researched Solution

Cache Line Disabling

+ Double Error Correcting

Triple Error Detecting Codes

(1-bit error correction for hard errors and 1-bit error correction for soft errors)
Researched Solution (cont.)
Cache Line Disabling

- On ROM software called Built In Self Test checks for errors during initialization of processor
- Detects **Hard** Cache faults
- Disables Cache lines with more than one fault
- Only works if the number of faulty lines is low
- Requires no additional hardware overhead
Cache Line Disabling (cont.)

- SRAM - Modeled as 3 blocks
  - Memory Cell Array, Address Decoder, and Read/Write Circuit
  - All faults can be detected by checking memory cells
- Two general types of cache errors
  - Single-Cell and Multi-Cell
    - Stuck-At 1/0, Stuck Open, State Transition, Data Retention
    - Cell State Coupling, Multiple Address
Cache Line Disabling (cont.)

- Any BIST with 100% correctness can be used
- Most efficient is 9N with an added Data Retention test
- 9N utilizes ‘Marching’
  - For Word-Oriented SRAM, each march element is a word
    - Introduces Multi-Cell errors
- Writes one march element at a time to each word address, reads it, compares them
- After marching, data retention test runs
- This method performs all checks with the best speed/ROM space ratio
ECC Overview

- Error Detection with Parity Codes
can detect all odd number faults, cannot correct error

- Single-Error Correction Codes
  - product codes (2-D parity check), parity check matrices
  - such coding scheme is commonly referred to as **Hamming codes**
  - cannot simultaneously detect double-bit errors
ECC Overview (cont.)

FIGURE 5.6 Parity check matrices for the code word: $D_1D_2D_3D_4$ with corresponding code bits $C_1C_2C_3$. (a) Parity check matrix where code bit columns are contiguous. (b) Parity check matrix where the code bit columns are in power of two positions. The syndrome (discussed in the text) for (b) points to the bit position in error, if the count of the column is started from 1. The syndrome zero would mean no error.
ECC Overview (cont.)

- Single-Error Correct Double-Error Detect Codes
  - widely used to protect memory cells
  - extend SEC by adding an extra bit that represents the parity over the seven bits of the SEC code word

\[
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\end{bmatrix}
\]
A parity check matrix for a \((N - 1, N - 2m - 2)\) DECTED code is usually expressed in a compact form as:

\[
\begin{bmatrix}
1 & 1 & 1 & K & 1 \\
1 & X & X^2 & K & X^{N-2} \\
1 & X^3 & X^6 & K & X^{3(N-2)}
\end{bmatrix}
\]

where \(X\) is the root of a primitive binary polynomial \(P(x)\) of degree \(m\), \(N = 2^m\), the number of data bits is \(N - 2m - 2\), and number of check bits is \(2m + 1\). The design may have to incur the overhead of extra bits if the number of data bits is fewer than \(N - 2m - 2\).
DECTED (cont.)
Simulation Comparison

![Bar Graph]

- SECDED: 563 mV
- CLD: 371 mV
- VS_ECC: 362 mV
- Proposed: 321 mV
- MS_ECC: 403 mV
Simulation Comparison (cont.)

- (a) Effective Cache Size (MB) vs. Supply Voltage (V)
  - VS_ECC_Disabling
  - Cache Line Disabling
  - Proposed Approach

- (b) Cache Miss Rate vs. Supply Voltage (V)
  - VS_ECC_Disabling
  - Cache Line Disabling
  - Proposed Approach
Simulation Comparison (cont.)

![Simulation Comparison Diagram](image)

- **Y-axis:** Energy Consumption (Normalized)
- **X-axis:** LU, fft, radix, Barnes, Cholesky, Stream
- Legend:
  - SECDED
  - Cache Line Disabling
  - VS_ECC_Disabling
  - Proposed Approach
  - MS_ECC
Simulation Comparison (cont.)
Simulation Comparison (cont.)

![Bar Chart]

- Instruction Per Cycle
- Normalized
- LU, fft, radix, Barnes, Cholesky, Stream
- SECDED, Cache Line Disabling, VS_ECC_Disabling, Proposed Approach, MS_ECC
Simulation Comparison (cont.)

(a) Effective Cache Size (MB)

(b) Cache Miss rate

- Proposed Approach
- VS_ECC_Disabling
- Cache Line_Disabling


Questions?