Memory Bus Encoding for Low Power: A Tutorial

Wei-Chung Cheng and Massoud Pedram
University of Southern California
Department of EE-Systems
Los Angeles CA 90089

Outline

• Background
• Memory Bus Encoding Techniques
  – Algebraic codes
  – Permutation codes
  – Probabilistic codes
• Conclusions
Power Dissipation Equation

- \( P \sim V^2 \cdot C \cdot f \cdot N \)
- Low Power Techniques
  - Voltage Scaling
  - Capacitance Reduction
  - Frequency Scaling
  - Switching Activity Reduction
- Memory Modules
  - Fixed
  - High
  - High
  - Memory Bus Encoding

Bus Encoding Example

<table>
<thead>
<tr>
<th>Binary Code</th>
<th>Gray Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1: 0000</td>
<td>B1: 0000</td>
</tr>
<tr>
<td>A2: 0001</td>
<td></td>
</tr>
<tr>
<td>A3: 0010</td>
<td>B2: 0001</td>
</tr>
<tr>
<td>A4: 0011</td>
<td>B3: 0011</td>
</tr>
<tr>
<td></td>
<td>B4: 0010</td>
</tr>
<tr>
<td>SA=6</td>
<td>SA=4</td>
</tr>
</tbody>
</table>
Generic Bus Encoding Architecture

Bus Encoding Taxonomy
- Redundancy
  - Irredundant: Gray, Pyramid
  - Redundant: T0, Bus Invert, Working Zone
- Circuit
  - Non-terminated: TTL, LVCMOS
  - Terminated: RAMBUS, GTL
- Signal Level
  - Level Signaling
  - Transition Signaling
- Location
  - On-chip Bus: Between CPU core and Caches
  - Host Bus: Between Pentium and Chipset
  - Memory Bus: Between Chipset and DRAM
- Address/Data
  - Separated
  - Multiplexed
- Multiplexing
  - Non-multiplexed: SRAM
  - Multiplexed: DRAM
Code Classification

1. Algebraic Codes
   \( c_i \, op \, x \) : \( op \) is a binary operation

2. Permutation Codes
   \( f(c_i) \) : \( f \) is a fixed function

3. Probabilistic Codes
   \( f_x(c_i) \) : \( f_x \) is an application-specific function

1 Algebraic Framework

- Decoding
  - \( c_i \, op \, x \)
- Notation
  - \(<\{x\},op>\)
Bus Invert: \(<\{0,1\},XOR>\)

- Stan, TVLSI 1995
- Extra signal: \(INV\)
  \[ s_i = c_i, \quad \text{if } INV=0 \]
  \[ s_i = c_i \oplus 1, \quad \text{if } INV=1 \]
- Encoding
  - Hamming distance

Partial Bus Invert: \(<\{0,x\},XOR>\)

- Shin et al., ISLPED 1998
  - Bus Partitioning
- Extensions
  - M-redundant Bus Invert
    - Spatial partitioning
  - Interleaving Partial Bus Invert, ICVC 1999
    - Temporal partitioning
Transition Signaling: $\langle c_{i-1}, \text{XOR} \rangle$

- Decoding function
  $s_i = c_i \text{XOR} c_{i-1}$
- Efficient when $c_i$ and $c_{i-1}$ are similar

T0: $\langle s_{i-1}, \text{Add,1} \rangle$

- Benini et al., Great Lakes VLSI Symp. 1997
- Extra signal: INC
  $s_i = s_{i-1} \text{ add 1},$ if INC=1
  $s_i = c_i,$ if INC=0
- Effective for sequential access patterns
- Prediction
**Prediction-based:** \(<\{s_{i-1}\},\text{XOR},1\>\)

- Ramprasad et al., TVLSI 1999
  - Inc-Xor
- Fornaciari et al., CODES 2000
  - Offset-Xor
  - T0-Xor
- For sequential access patterns

**Hybrid Encoding**

- Benini et al, DATE 1998
- Instruction/Data interleaving
- T0 for instructions; Bus Invert for data
- Examples
  - T0_BI
  - Dual_T0
  - Dual_T0_BI
Working Zone: \{a[], ADD\}

- Musoll et al, TVLSI 1998
- Instruction/Data segments
- Offset
- One-hot coding

Comparison

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Binary</th>
<th>Ø Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus Invert</td>
<td>{0,1} XOR</td>
<td></td>
</tr>
<tr>
<td>Partial Bus Invert</td>
<td>{0,x} XOR</td>
<td></td>
</tr>
<tr>
<td>Transition Signaling</td>
<td>{c_{i-1}} XOR</td>
<td></td>
</tr>
<tr>
<td>T0</td>
<td>{c_{i-1}} ADD_1</td>
<td></td>
</tr>
<tr>
<td>Inc-Xor</td>
<td>{c_{i-1}} XOR_1</td>
<td></td>
</tr>
<tr>
<td>Working Zone</td>
<td>{x[]} ADD</td>
<td></td>
</tr>
</tbody>
</table>
2. Permutation Codes

- Fixed function: \( f(c_i) \)
- Irredundant
- Do not need the previous word \( s_{i-1} \) or \( c_{i-1} \)
- Examples
  - Gray code
  - Pyramid code
- For sequential access patterns

Gray Code

- Su et al., ISLPED 1995
- Only one transition between consecutive words
- For address busses
Pyramid Code

- Cheng et al., ISPLED 2000
- For multiplexed DRAM address busses
- No transition between consecutive words
- 50% switching activity reduction

3. Probabilistic Code

- Given a program trace
- Statistics Information
  - First-order: \( f(c_i) \)
  - Second-order (pair-wise): \( f(c_{i-1}, c_i) \)
- Examples
  - Static analysis
    - Limited-weight code, Beach code, Clustered and Discretized code
  - Dynamic analysis
    - Adaptive, Codebook-based
Limited Weight Code

- Stan et al., TVLSI 1997
- $K$-limited code
- First-order analysis
- First-order encoding

Beach Code

- Benini et al., TVLSI 1998
- Second-order analysis
- First-order encoding
Entropy-reduced Framework

- Ramprasad et al., TVLSI 1999
- Functions
  - $F$: predict
    - Identity
    - Increment
  - $f1$: error
    - Xor
    - Difference
  - $f2$: entropy
    - Invert
    - Probability (pbm)
    - Value (vbm)
- Examples
  - Second-order analysis; First-order encoding
  - $\langle s_i, f_i \rangle$, $\text{xor} \cdot f(c_i)$

<table>
<thead>
<tr>
<th>Code-name</th>
<th>$F$</th>
<th>$f1$</th>
<th>$f2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>xor-pbm</td>
<td>identity</td>
<td>xor</td>
<td>pbm</td>
</tr>
<tr>
<td>inc-xor</td>
<td>increm</td>
<td>xor</td>
<td>identity</td>
</tr>
</tbody>
</table>

Transition Encoding

- Benini et al, DAC 1999
- Second-order analysis
- Second-order encoding
  - Encode transitions instead of words
- Static
  - Exact
  - Clustered
  - Discretized
- Adaptive
Codebook

- Komatsu et al., Great Lakes VLSI Symp. 1999
- Second-order analysis
- Second-order encoding
- Sort and encode dynamically

More Recent Work

- Coupling-driven encoding
  - Kim et al., ICCAD 2000
  - Sotirsadis et al., ICCAD 2000
Conclusions

- Encoding can reduce the switched capacitance on a bus

- Different types of codes have been proposed, each applicable to a particular type of bus and data access pattern
  - Algebraic codes
    - \( \langle x \rangle \text{.op} \)
  - Permutation codes
  - Probabilistic codes
    - Analysis/Encoding
      - First-order
      - Second-order
    - Static vs. Adaptive