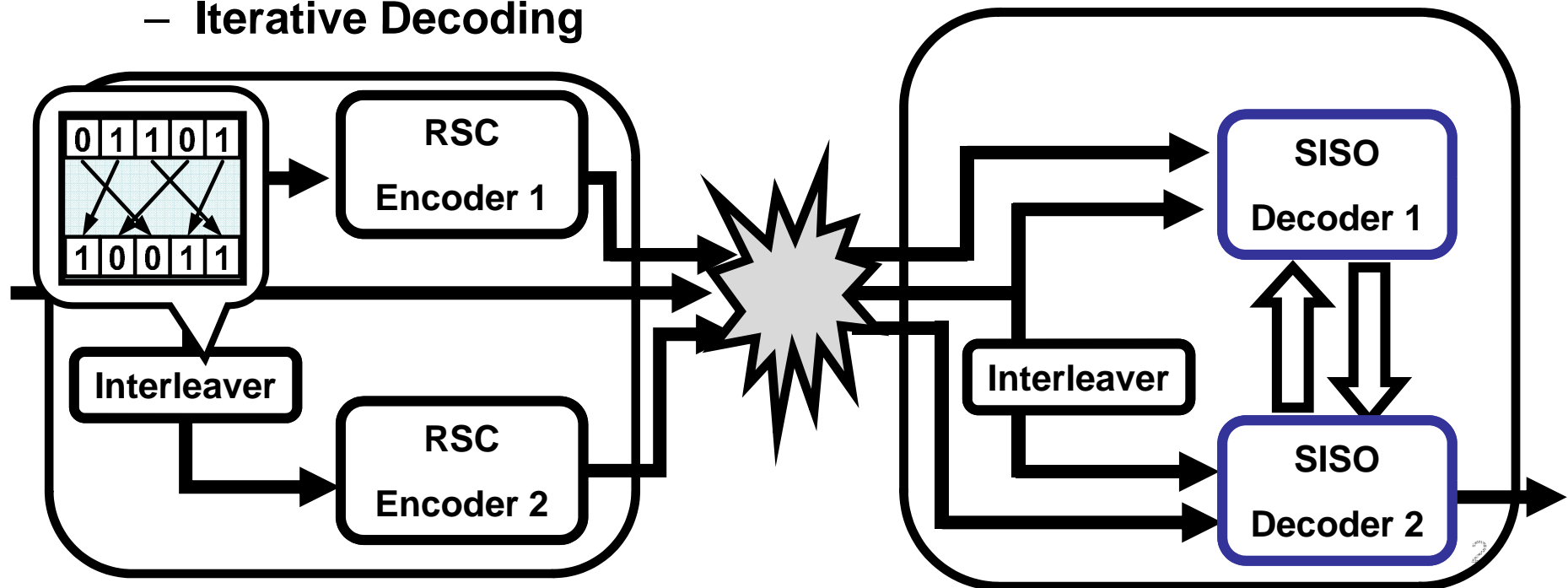# Duo-Binary Circular Turbo Decoder Based on Border Metric Encoding for WiMAX

Ji-Hoon Kim and In-Cheol Park

Division of Electrical Engineering

KAIST

Introduction to Turbo Codes

- ## Introduced in 1993
  - ### Error Correcting Performance within 0.5dB of Shannon limit
  - ### Widely Used in W-CDMA, CDMA2000, and WiMAX
  - ### Non-Uniform Interleaver
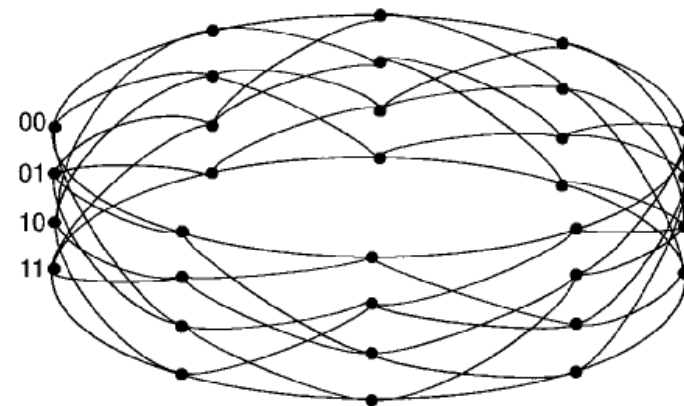 - ### Iterative Decoding

Turbo Codes for WiMAX

- **Double-Binary Turbo Codes**
  - **Better Convergence**
  - **Larger Minimum Distances**
  - **Reduced Latency**

- **Circular Coding**
  - **a.k.a Tail-Biting**
  - **No Tail Bits**
  - **Avoid Spectrum Waste**

< CTC Encoder for WiMAX >
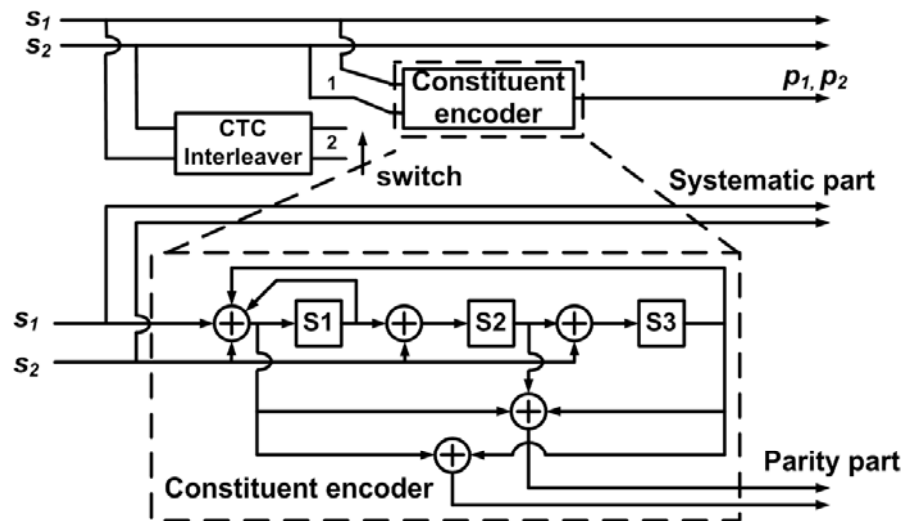
< 4-State Trellis for Circular Coding >

3

Max-log-MAP for Double-Binary

Z (Source Bits)

11/00/01/10   7
01/10/11/00   6
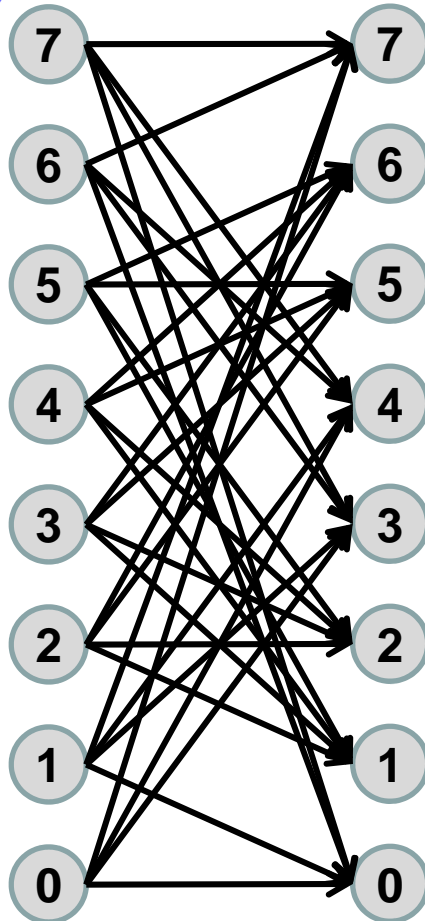11/00/01/10   5
01/10/11/00   4
10/01/00/11   3
00/11/10/01   2
10/01/00/11   1
00/11/10/01   0

$$\tilde{\alpha}_k(s_k) \cong \max_{s_{k-1} \in A}\left[\tilde{\alpha}_{k-1}(s_{k-1}) + \tilde{\gamma}_k(s_{k-1} \rightarrow s_k)\right]$$

$$\tilde{\beta}_k(s_k) \cong \max_{s_{k+1} \in B}\left[\tilde{\beta}_{k+1}(s_{k+1}) + \tilde{\gamma}_{k+1}(s_k \rightarrow s_{k+1})\right]$$

$$\tilde{\gamma}_k(s_k \rightarrow s_{k+1}) = \ln\left[P(\mathbf{y}_k \mid \mathbf{x}_k) \cdot P(u_k = z)\right]$$

$$= \frac{L_c}{2}(x_k^{s_1} y_k^{s_1} + x_k^{s_2} y_k^{s_2} + x_k^{p_1} y_k^{p_1} + x_k^{p_2} y_k^{p_2}) + L_{e,IN}^{(z)}$$

$$\Lambda_k^{(z)} \cong \max_{(s_k \rightarrow s_{k+1}, z)}\left[\tilde{\alpha}_k(s_k) + \tilde{\gamma}_{k+1}(s_k \rightarrow s_{k+1}) + \tilde{\beta}_{k+1}(s_{k+1})\right]$$

$$- \max_{(s_k \rightarrow s_{k+1}, 00)}\left[\tilde{\alpha}_k(s_k) + \tilde{\gamma}_{k+1}(s_k \rightarrow s_{k+1}) + \tilde{\beta}_{k+1}(s_{k+1})\right]$$
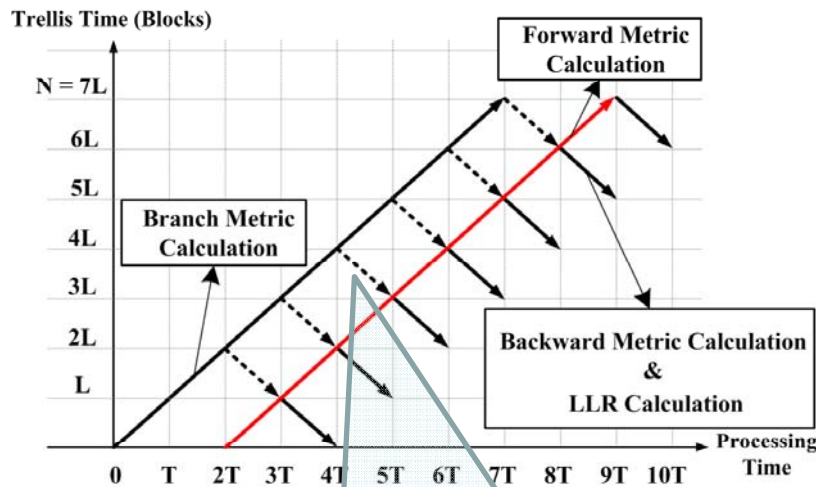
Avoid "Complex Metric Calculation" with Minimum Overhead

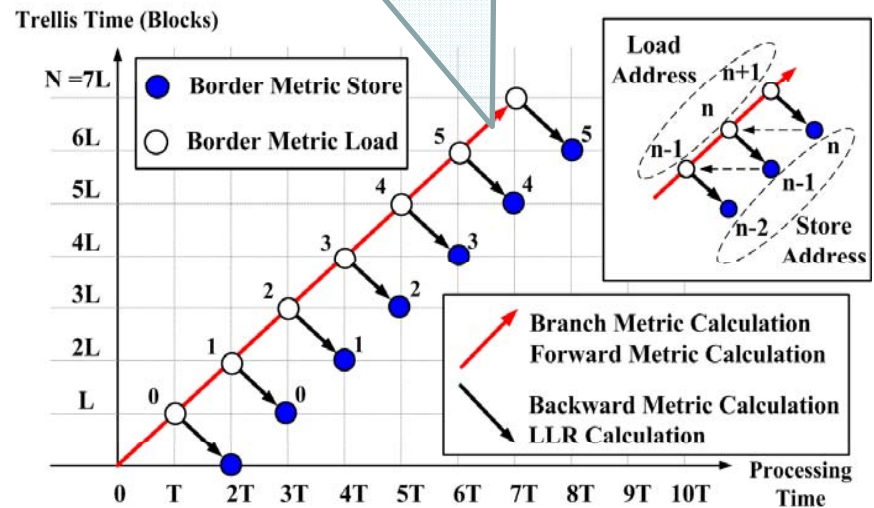# Sliding Window
For Non-Binary Turbo Decoder

- **Border Metric with the values of previous Iteration**
  - **Avoid complex dummy calculation**

Small Window Size, Large Frame Size → Huge Border Memory!

Complex Dummy Metric Calculation @ Border

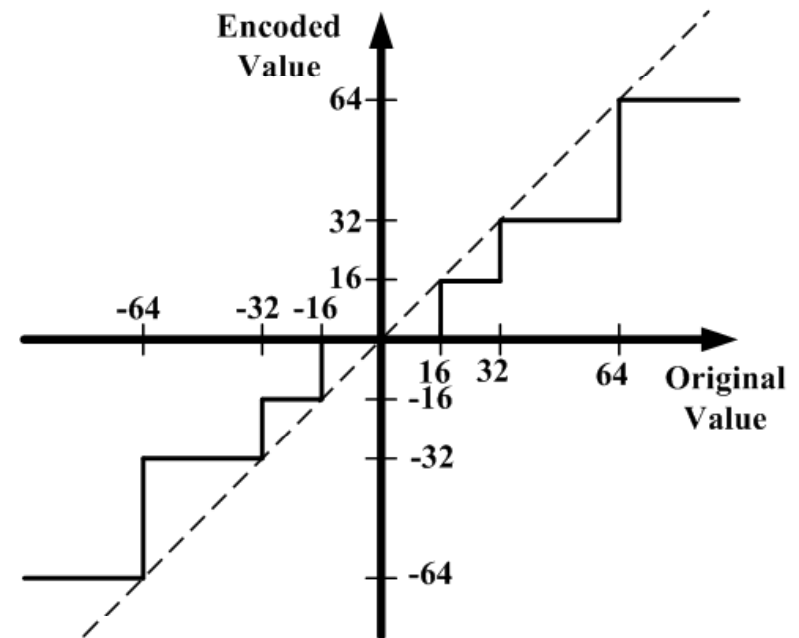**Efficient for Circular Coding**

Proposed Border Metric Encoding

- **Allow only a few values for the Border Metric**
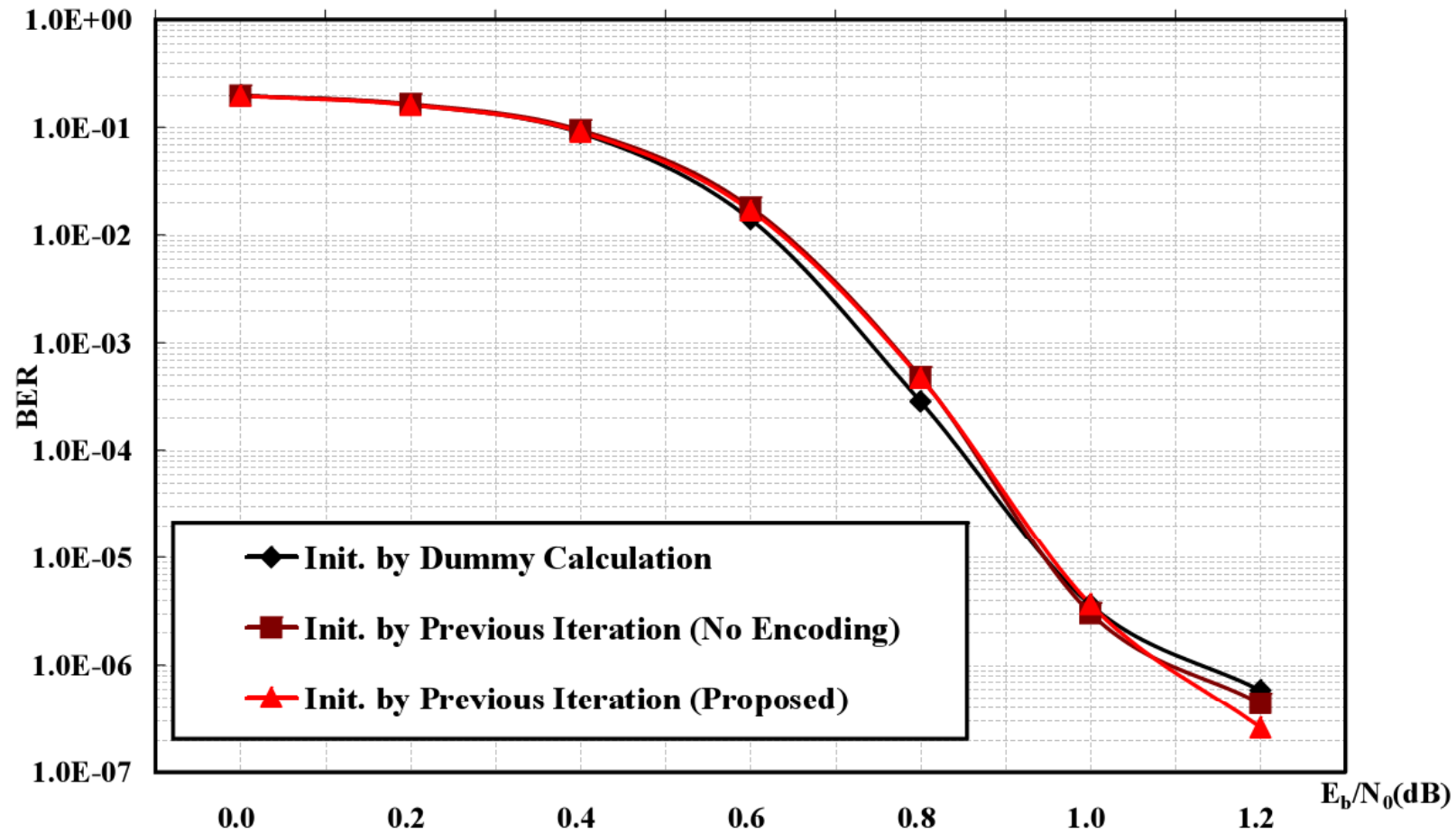
Any Special
Simple Encoding?

Flooring the Original Value
to the **Closest Power-of-Two Number**

Encoding Scheme	Encoded Values
4-bit Encoding	±256, ±128, ±64, ±32, ±16, ±8, ±4, 0 (15 values)

BER Performance Comparison

- **Almost No Performance Degradation!**

Proposed Dedicated Interleaver

- ## Accumulator-Based Interleaver
  - ### Small Area due to Simple Hardware

Step 1 : Switch alternate couples

for $i = 0 \ldots N{-}1$

if $(i_{mod\,2} == 1)$ let $(A_i, B_i) \rightarrow (B_i, A_i)$ (i.e., switch the couple)

Step 2 : Inter-symbol permutation
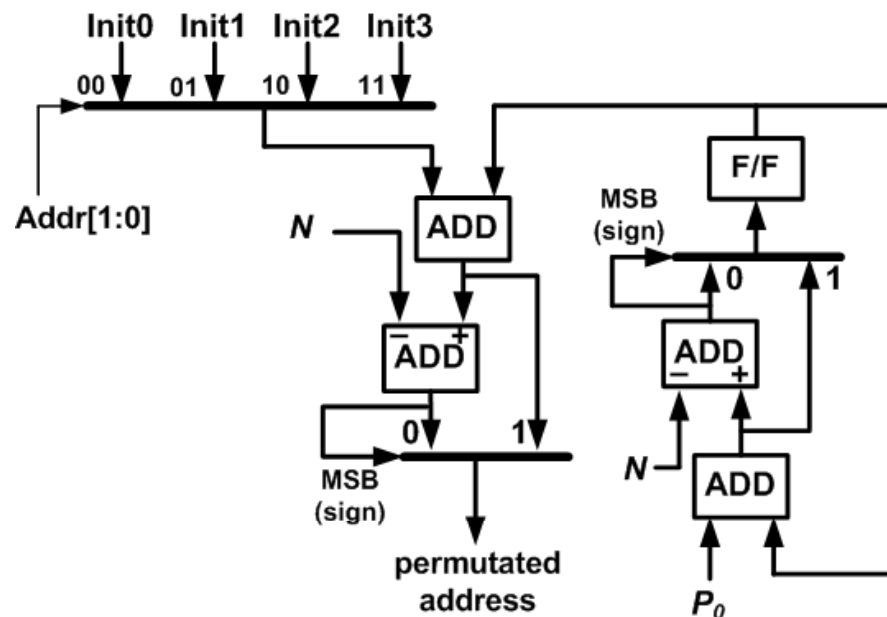
for $i = 0 \ldots N{-}1$

switch $i \bmod 4$:

case 0 : $P(i) = (P_0{\cdot}i + 1)_{mod\,N}$

case 1 : $P(i) = (P_0{\cdot}i + 1 + N/2 + P_1)_{mod\,N}$

case 2 : $P(i) = (P_0{\cdot}i + 1 + P_2)_{mod\,N}$
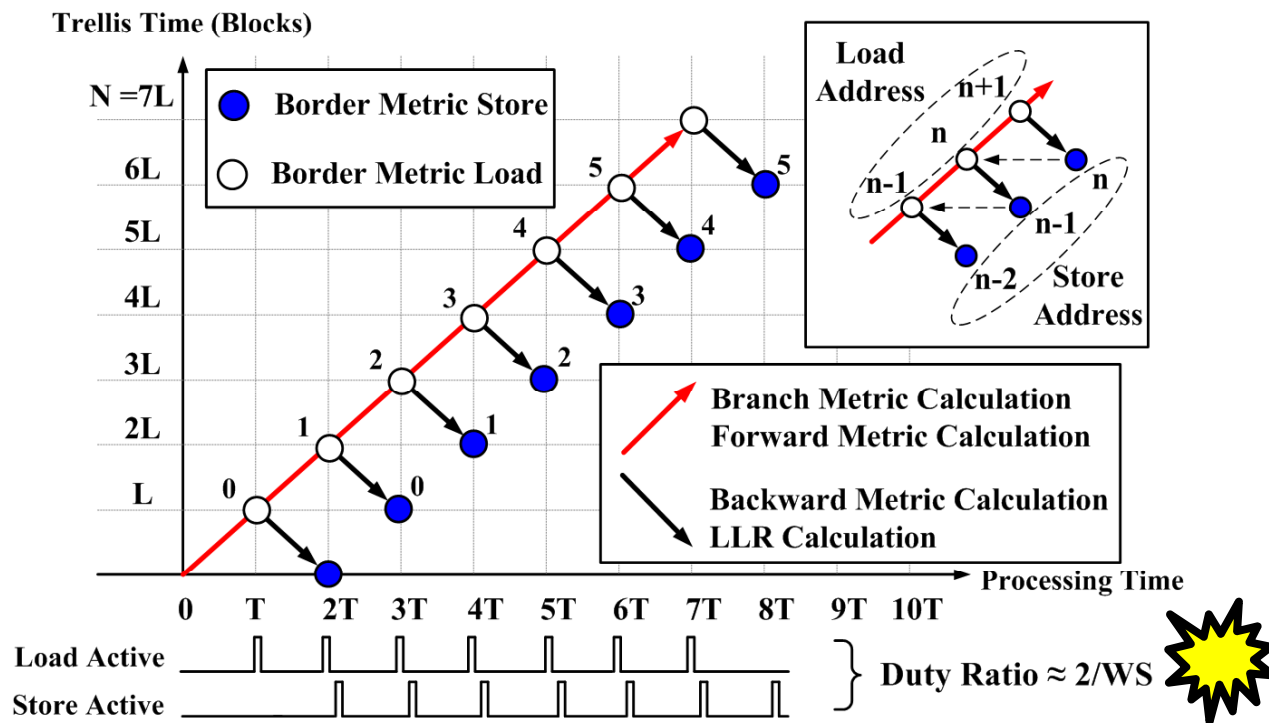
case 3 : $P(i) = (P_0{\cdot}i + 1 + N/2 + P_3)_{mod\,N}$

$$P(i) = (P_0{\cdot}i + Init)_{mod\,N}$$
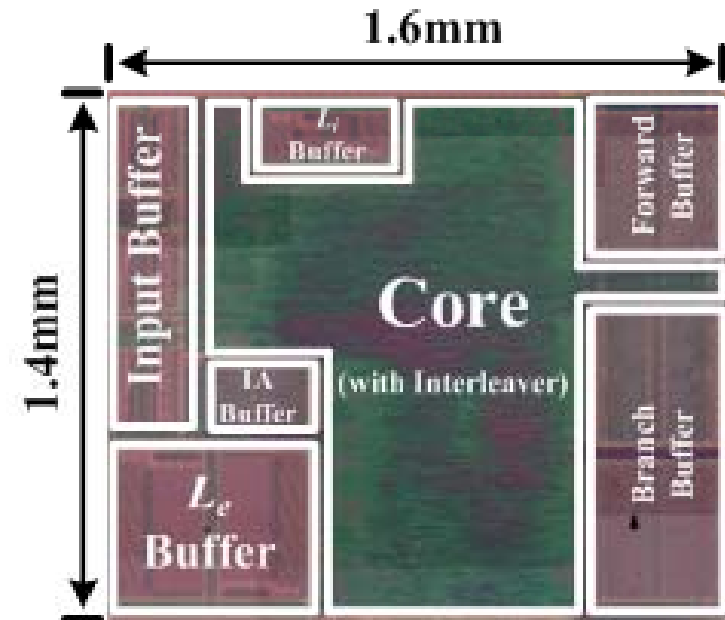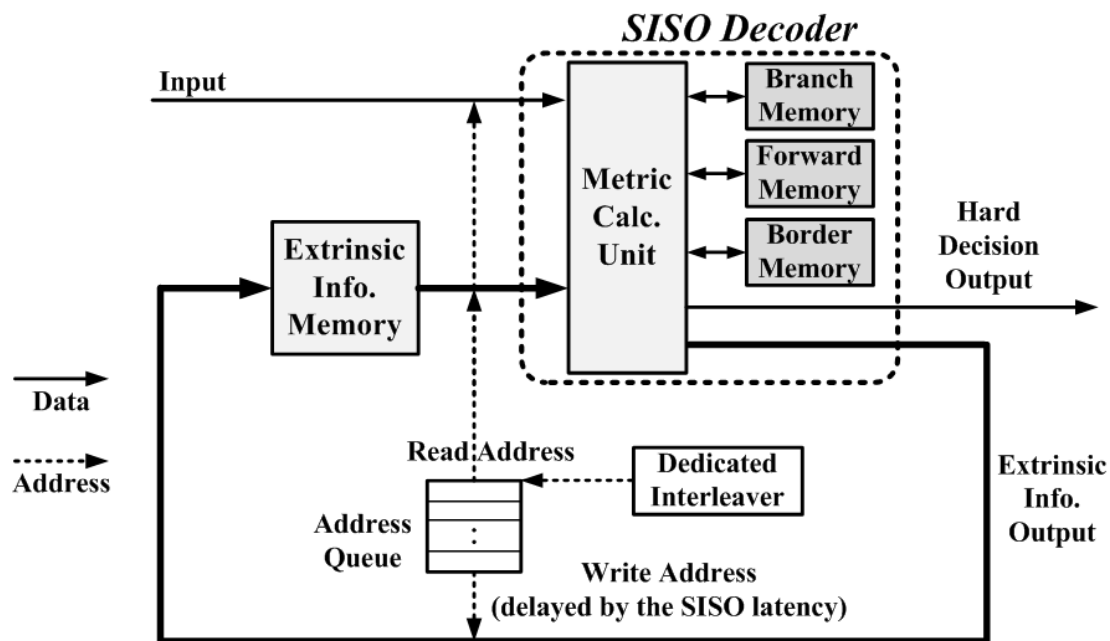$$= [(P_0{\cdot}i)_{mod\,N} + (Init)_{mod\,N}]_{mod\,N}$$

8

Key to Low-Power Consumption

- ## Small-Sized Border Memory
  - **By Border Metric Encoding**

- ## Infrequent Access to Border Memory
  - **Only one load/store for processing one Window**

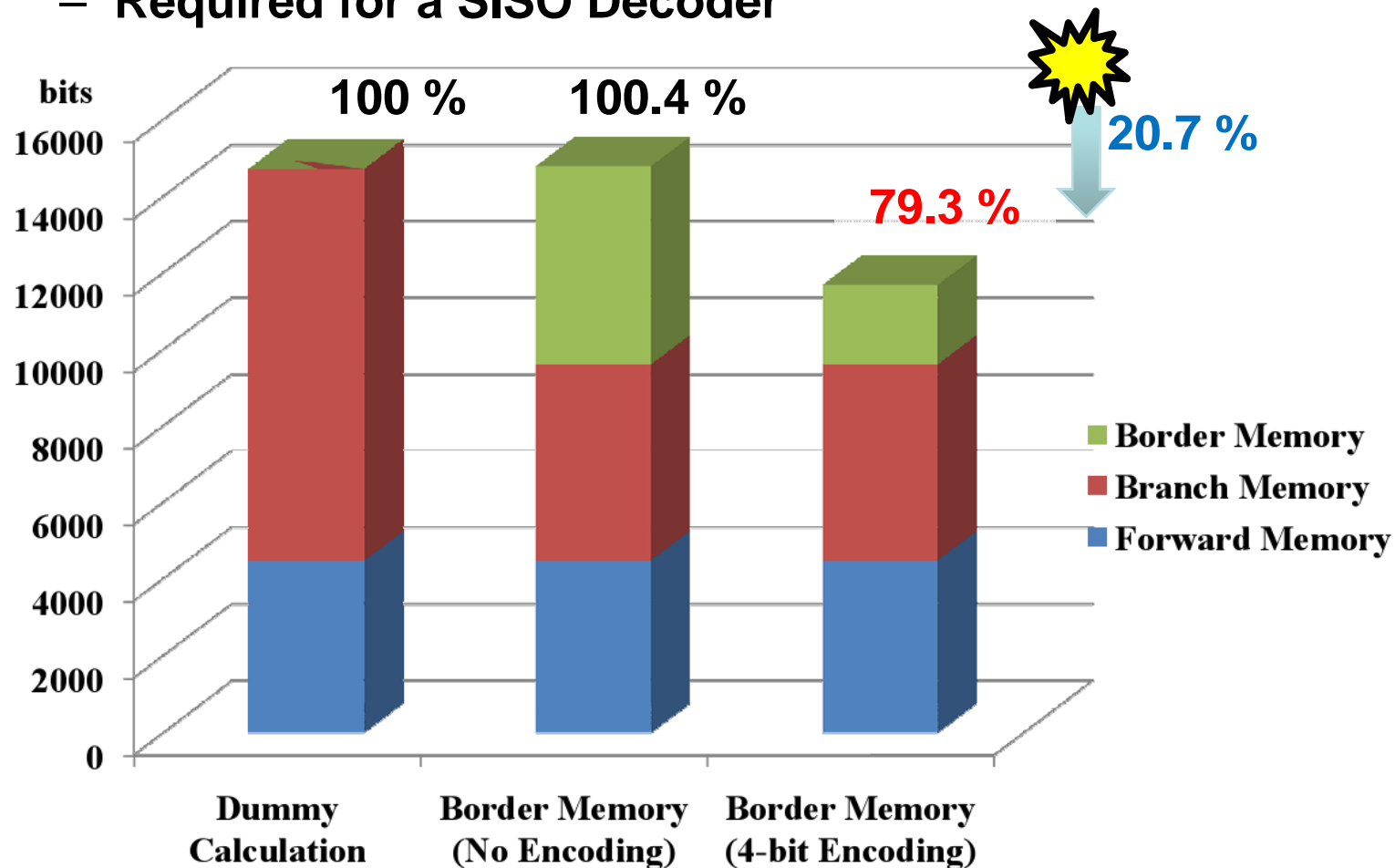# Implementation Results

- **Time-Multiplex Architecture**

SISO Algorithm	Max-log-MAP (Duo-Binary)
Window Size	32
Gate Count	65k

Operating Frequency	200 MHz
Iteration	8 (Fixed)
Data Rate	24.26 Mbps

Memory Size Comparison

- **Single-Port SRAM Size**
  - **Required for a SISO Decoder**

100 %   100.4 %   20.7 %   79.3 %

- Border Memory
- Branch Memory
- Forward Memory

Dummy Calculation | Border Memory (No Encoding) | Border Memory (4-bit Encoding)

11

Energy Consumption Comparison

- **For a SISO Decoder @ 1.2dB**
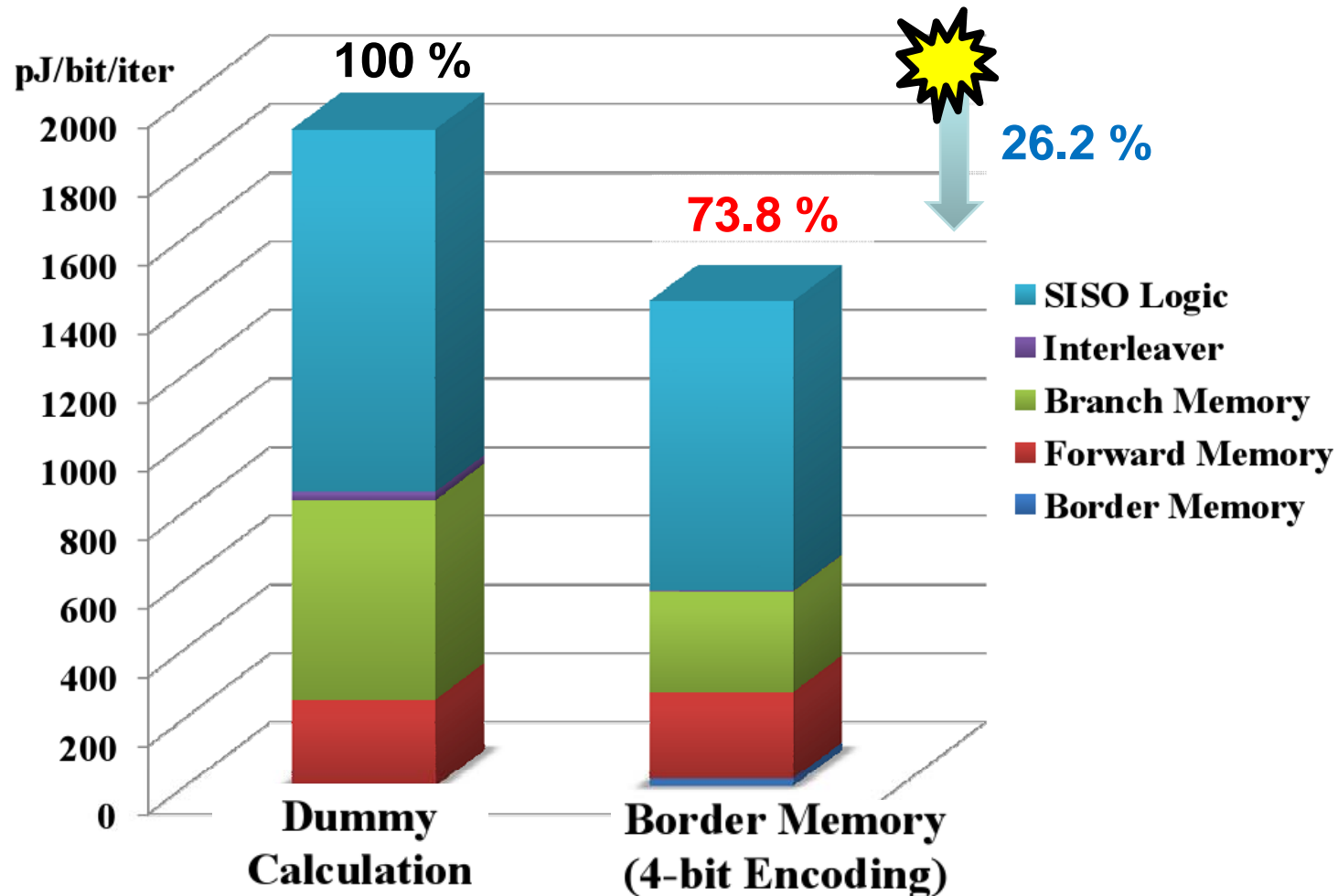
Conclusion

- **Border Metric Encoding is Proposed**
  - Avoid Complex Dummy Calculation
  - Effective for non-binary Turbo Codes

- **Dedicated Hardware Interleaver is Proposed**
  - Generate Interleaved Addresses *on-the-fly*

- **CTC Decoder for WiMAX is Designed**
  - Based on ..
    - Border Metric Encoding
    - Dedicated Hardware Interleaver

13