Analog Decoding

JASS'05

St. Petersburg

<Pavol.Hanus@tum.de>

Contents

- **Introduction**
  - History, Motivation, Advantages
- **Basics**
  - Analog Implementation of the Log-likelihood Algebra
- **Analog Decoder**
  - Graph based, Trellis based
- **Analog Turbo Decoder**
  - Interleaver as Cross-connecting Network, “Flooding"
- **Conclusion**
  - Results, Open Problems, Future Work

Historic Overview

- **Idea**
  - J. Hagenauer 1997 (TUM)
- **Implementation Aspects**
  - A. Loeliger, F. Lustenberger 1998 (ETH Zürich)
- **First Functioning Decoder Chip**
  - Matthias Mörz 1998/99 (TUM, Bell Labs)
- **Research Teams**
  - TUM, University of Utah, University of Alberta, University of Toronto, ETH Zürich, Politecnico di Torino, Università di Padova
  - Industry, Companies

Motivation

- **World of „digital“ transmission is analog**
  - Information is transmitted in form of analog waves
- **Soft-in/Soft-out algorithms (SISO) use analog values**
  - SOVA, APP, LDPC, …
  - Only analog is really soft
- **Speed**
  - Decoding speed limited only by the settling time
- **Small Area**
- **Low Power Consumption**
- **Simple Circuit Design**
  - Repetition of a few elementary circuits

Motivation for using Log-likelihood Algebra

A-posteriori LLR

$$L(x \mid y) = L_c y + L(x)$$

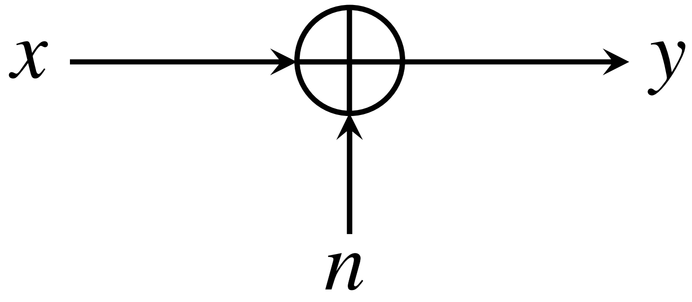
x : transmitted symbol

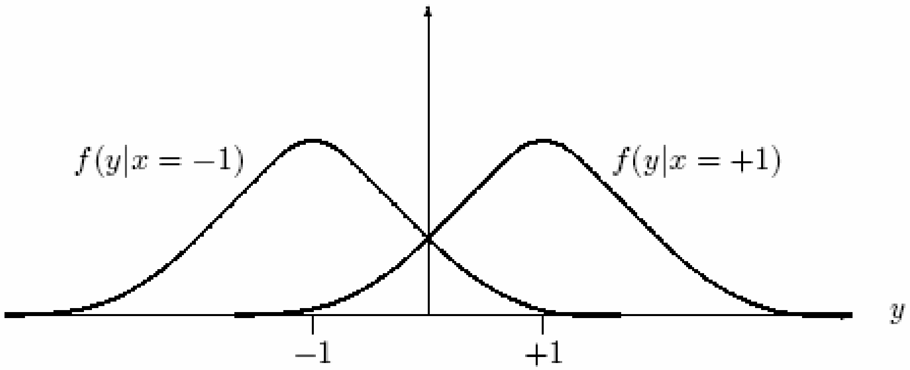
y : Matched Filter-output

Channel-state Information (Gaussian Fading Channel)

$$L_c = 4a \frac{E_s}{N_o}$$

a : Fading-Amplitude

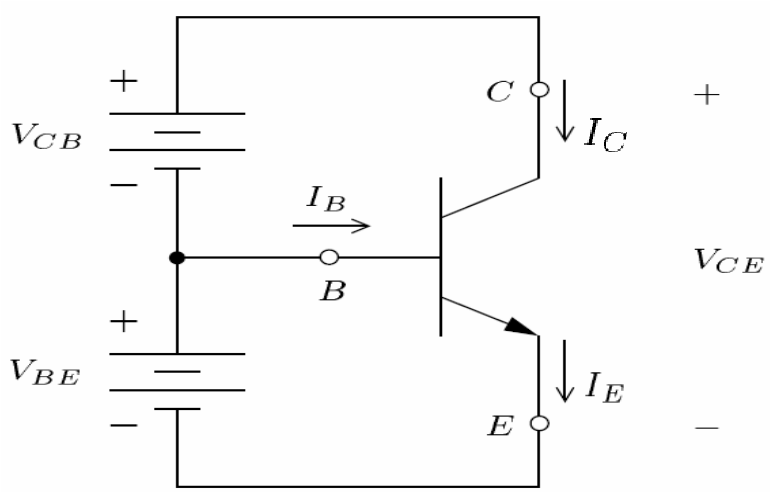




Binary Log-likelihood Algebra Quantities using Analog Circuits (Voltages and Currents)

Quantity	Formula	Analog *(V,I)*
LLRatio	$L(X) = \ln \frac{P_X(x=0)}{P_X(x=1)}$	**?**
Probability	$P_X(x=0) = \frac{1}{1+e^{-L(X)}}$ $P_X(x=1) = \frac{e^{-L(X)}}{1+e^{-L(X)}}$	**?**
Soft-bit	$\lambda(X) = E\{X\} = \tanh\left[\frac{L(X)}{2}\right]$ $= (+1)P_X(x=0) + (-1)P_X(x=1)$	**?**

Basic Circuits: Bipolar Junction Transistor (NPN-BJT)

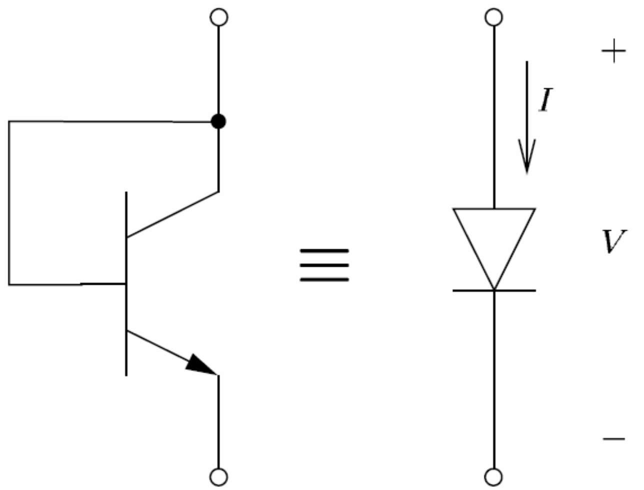


BJT in forward active mode

$$I_C = I_S\left(e^{V_{BE}/V_T} - 1\right) \approx I_S \cdot e^{V_{BE}/V_T}$$

$$V_T = kT/q \approx 26mV \text{ [Thermal Voltage]}$$

$$I_S \approx 10^{-16} A \text{ [Saturation Current]}$$



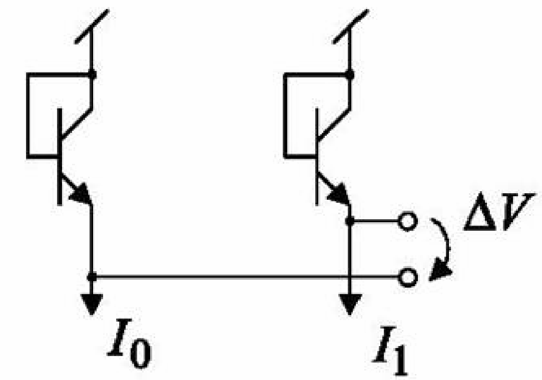
BJT as a PN Diode

$$I = I_S \cdot e^{V/V_T}$$

$$V = V_T \ln\left(I/I_S \right)$$

Basic Circuits: Transistor Pairs

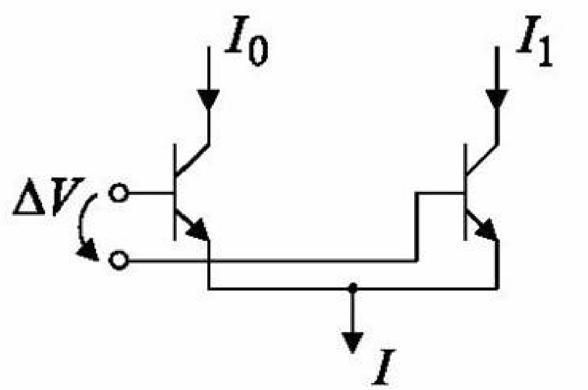
Pair of Diodes



$$\frac{\Delta V}{V_T} = \frac{V_0 - V_1}{V_T} = 2\tanh^{-1}\left(\frac{\Delta I}{I}\right) = L(X)$$

P=>L

Differential Pair



$$\frac{I_1}{I} = \frac{e^{-\Delta V / V_T}}{1 + e^{-\Delta V / V_T}} = P_X(x=1)$$

$$\frac{I_0}{I} = \frac{1}{1 + e^{-\Delta V / V_T}} = P_X(x=0)$$

$$\frac{\Delta I}{I} = \frac{I_0 - I_1}{I} = \tanh\left(\frac{1}{2}\frac{\Delta V}{V_T}\right) = \lambda(X)$$

L=>P

Binary Log-likelihood Algebra Quantities using Analog Circuits (Voltages and Currents)

Quantity	Formula	Analog *(V,I)*
LLRatio	$L(X) = \ln \frac{P_X(x=0)}{P_X(x=1)}$	$L(X) = \frac{\Delta V}{V_T}$
Probability	$P_X(x=0) = \frac{1}{1+e^{-L(X)}}$ $P_X(x=1) = \frac{e^{-L(X)}}{1+e^{-L(X)}}$	$P_X(x=i) = \frac{I_i}{I}$
Soft-bit	$\lambda(X) = E\{X\} = \tanh\left[\frac{L(X)}{2}\right]$ $= (+1)P_X(x=0) + (-1)P_X(x=1)$	$\lambda(X) = \frac{\Delta I}{I}$

Binary Log-likelihood Algebra Operations using Analog Circuits (Voltages and Currents)

BOX-Plus

$$L(X_3) = L(X_1) \boxplus L(X_2) = \ln \frac{P_X(x_3=0)}{P_X(x_3=1)} =$$

$$= \ln \frac{(P_X(x_1=0)P_X(x_2=0) + P_X(x_1=1)P_X(x_2=1))}{(P_X(x_1=0)P_X(x_2=1) + P_X(x_1=1)P_X(x_2=0))}$$

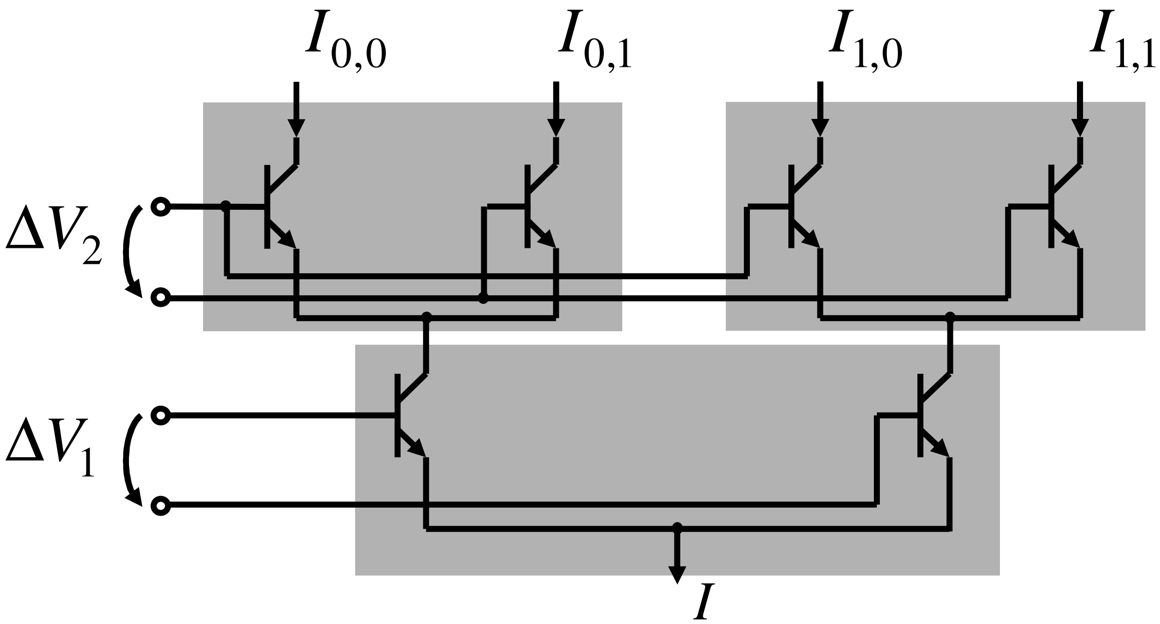
L-Values Addition

$$L(X_3) = L(X_1) + L(X_2) = \ln \frac{P_X(x_1=0)}{P_X(x_1=1)} + \ln \frac{P_X(x_2=0)}{P_X(x_2=1)} =$$

$$= \ln \frac{(P_X(x_1=0)P_X(x_2=0))}{(P_X(x_1=1)P_X(x_2=1))}$$

Basic Circuits: Probability Multiplication

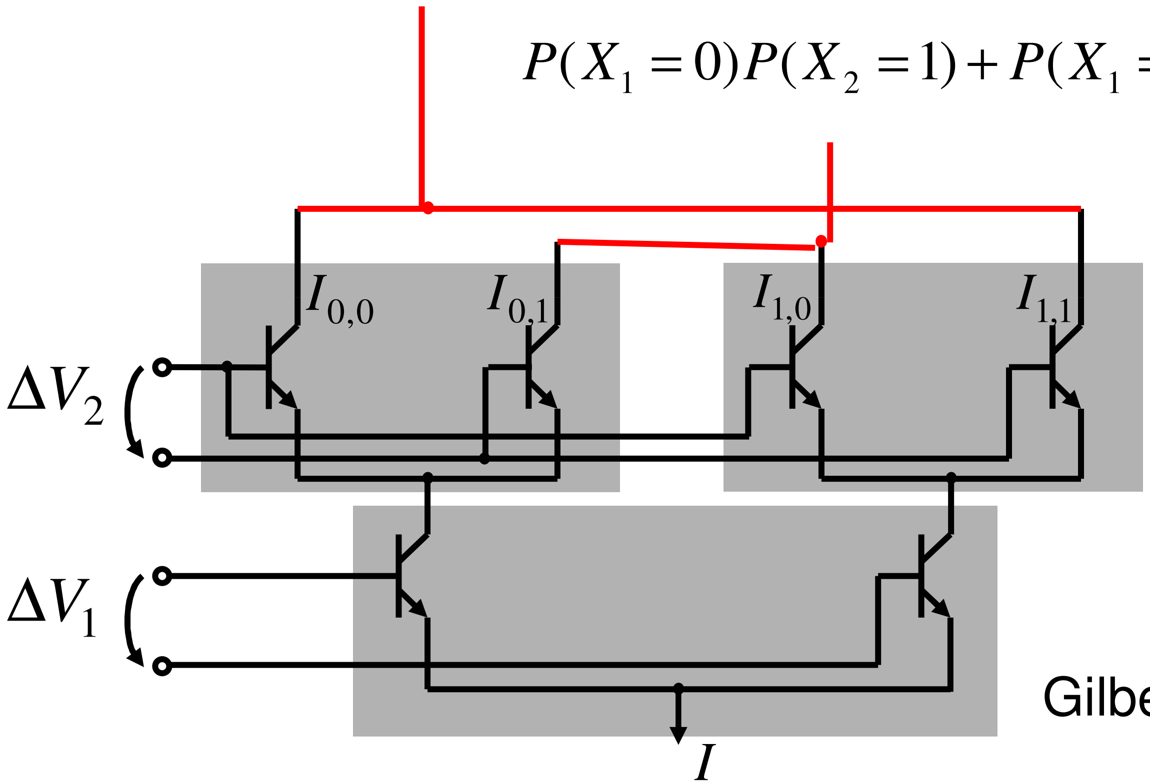
$$\frac{\Delta V_i}{V_T} = L(X_i) \qquad \frac{I_{i,j}}{I} = P_X(x_1 = i) \cdot P_X(x_2 = j)$$



Basic Circuits: Probability Addition

$$P(X_1 = 0)P(X_2 = 0) + P(X_1 = 1)P(X_2 = 1) = \frac{I_{00} + I_{11}}{I}$$

$$P(X_1 = 0)P(X_2 = 1) + P(X_1 = 1)P(X_2 = 0) = \frac{I_{01} + I_{10}}{I}$$

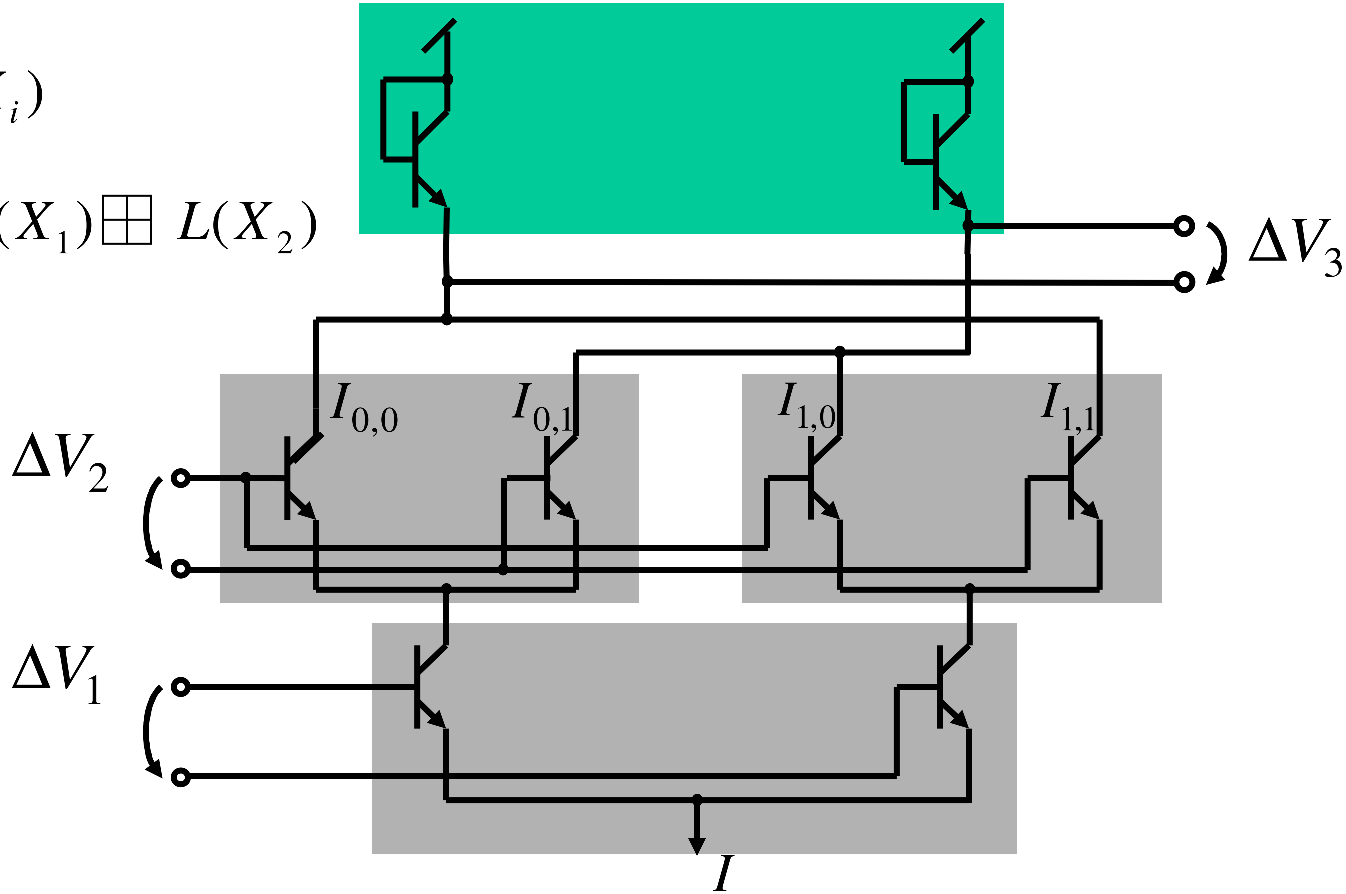


Gilbert Multiplicator

Basic Circuits: BOX-Plus

$$\frac{\Delta V_i}{V_T} = L(X_i)$$

$$L(X_3) = L(X_1) \boxplus L(X_2)$$

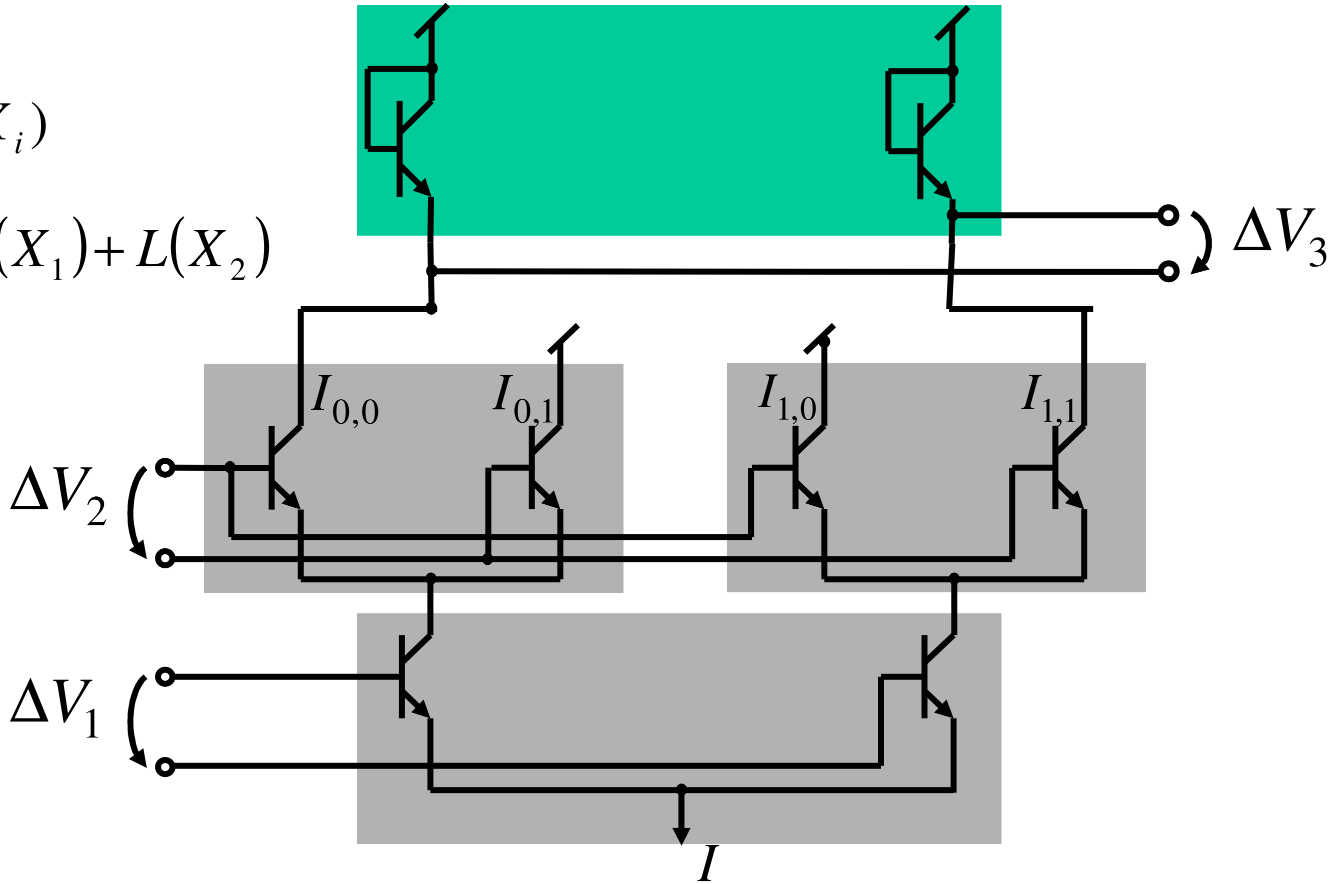


$$\frac{\Delta V_3}{V_T} = 2 \cdot \tanh^{-1}\left[\tanh\left(\frac{\Delta V_1}{2V_T}\right) \cdot \tanh\left(\frac{\Delta V_2}{2V_T}\right)\right]$$

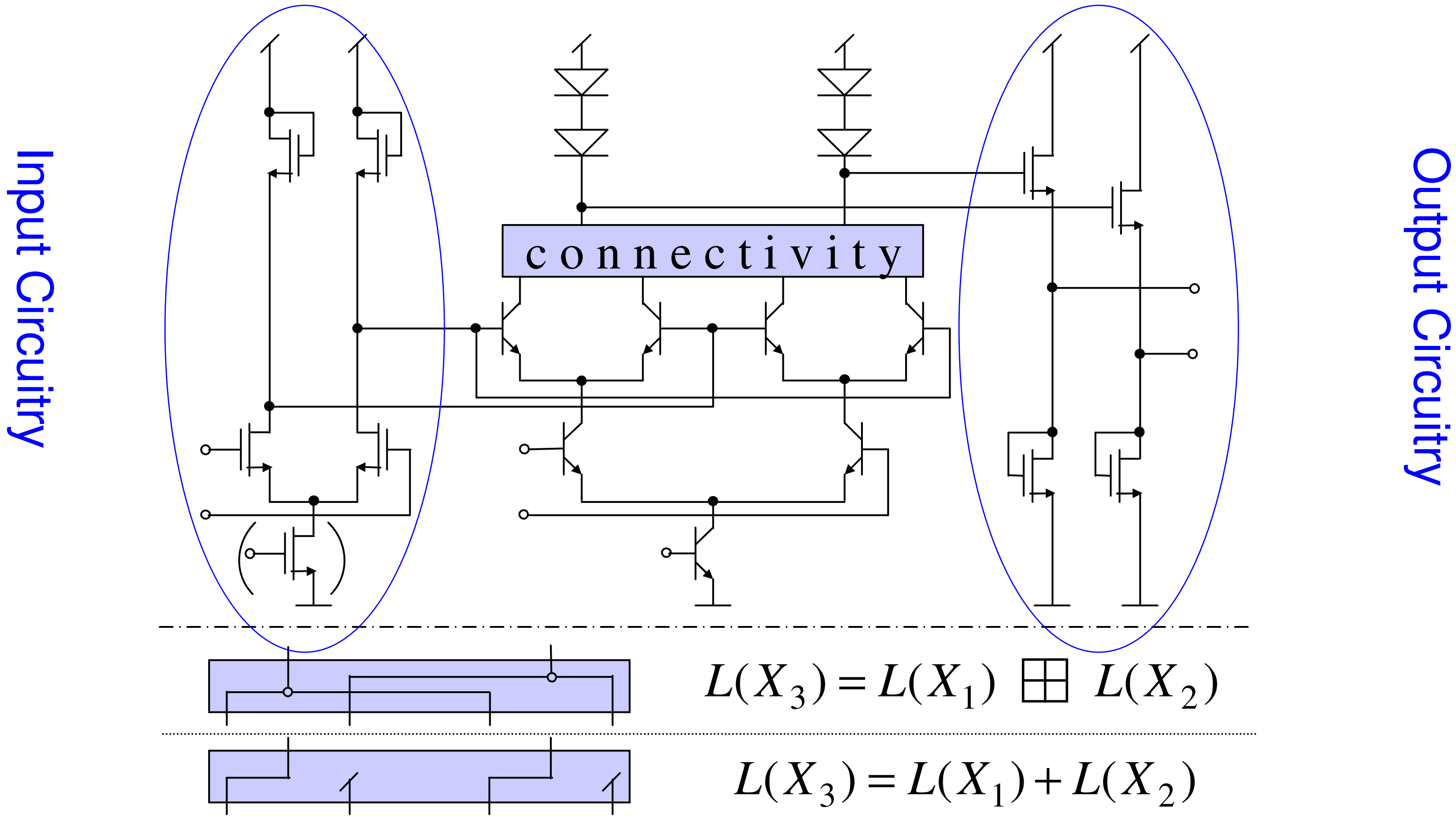
Basic Circuits: L-values Addition

$$\frac{\Delta V_i}{V_T} = L(X_i)$$

$$L(X_3) = L(X_1) + L(X_2)$$



Decoder Core Building Block with I/O

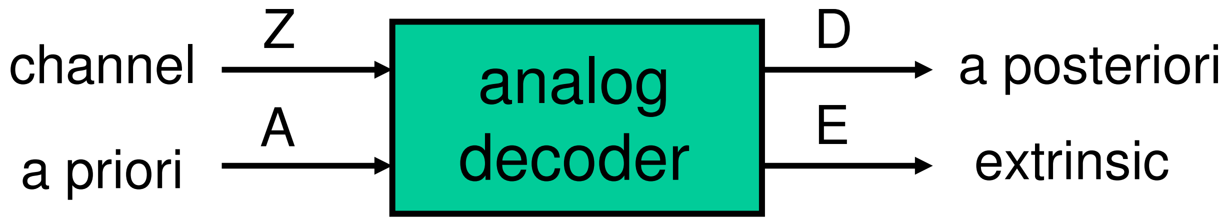


$$L(X_3) = L(X_1) \boxplus L(X_2)$$

$$L(X_3) = L(X_1) + L(X_2)$$

Soft-in / Soft-out Decoder (only analog is really soft)

Component decoder used in a turbo scheme

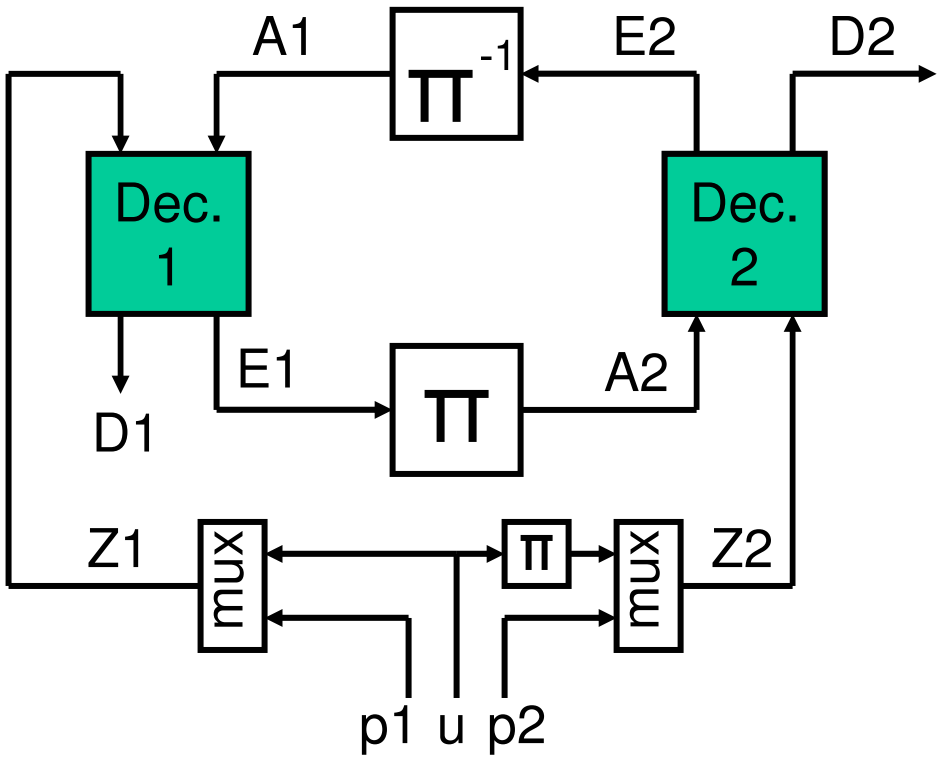


All values are Log-likelihood ratios (real numbers)

- **Digital implementation works with quantized values**
- **Analog Decoder works with voltages and currents**
- **Analog sliding window decoder as component decoder**

Analog Turbo Decoder
(soft-in, soft-out, soft-time)

Example Parallel Concatenated Codes:



Log-likelihood ratios

D: a posteriori decoder output

D = Z + A + E

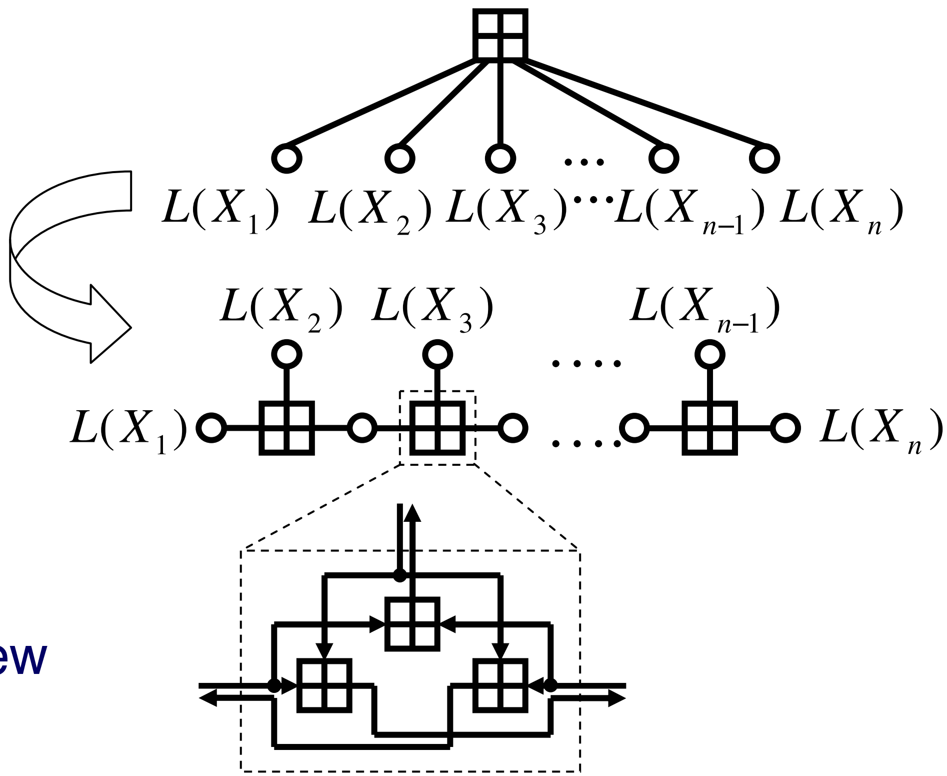
Z: channel output

A: a priori input

E: extrinsic input

Check Node Decoder

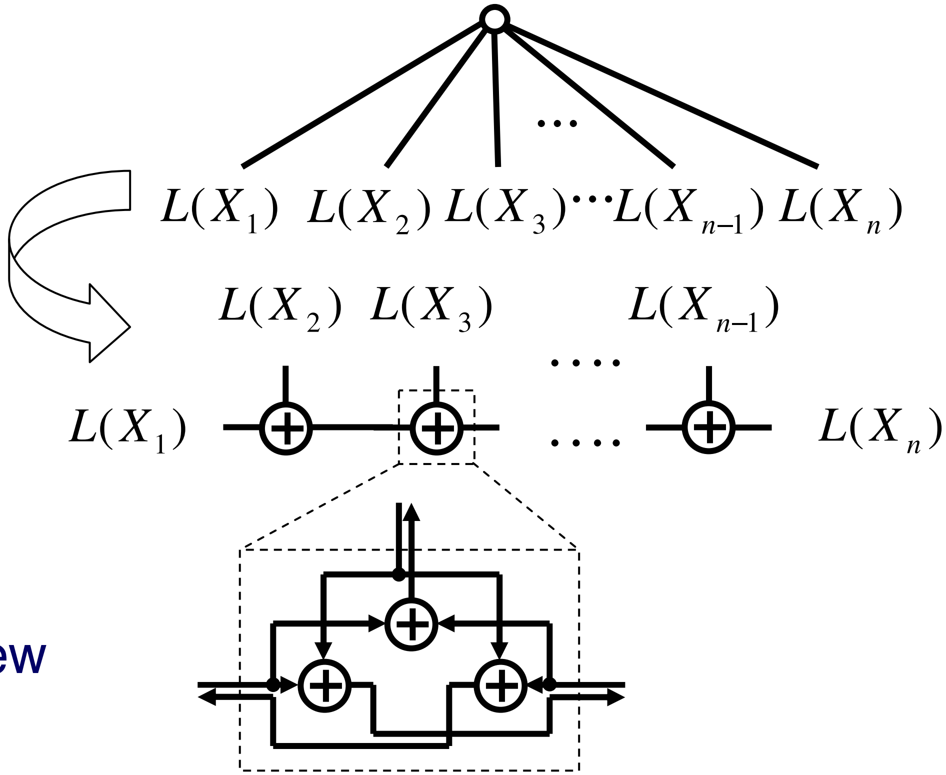
$$L(X_i) = \sum_{\substack{j=1 \\ j \neq i}}^{n} \boxplus \, L(X_j)$$



Detailed view

Variable Node Decoder

$$L(X_i) = \sum_{\substack{j=1 \\ j \neq i}}^{n} L(X_j)$$



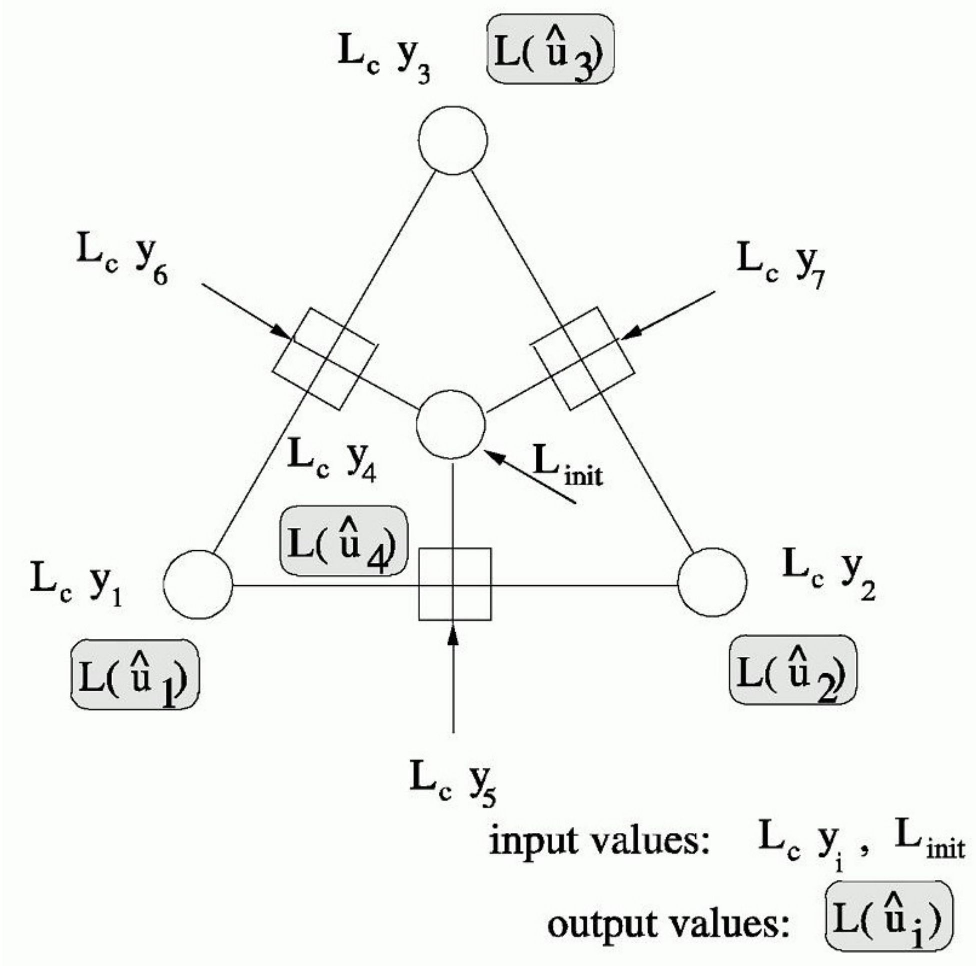
Detailed view

Tanner Graph Based Decoder

(7,4) Hamming Code

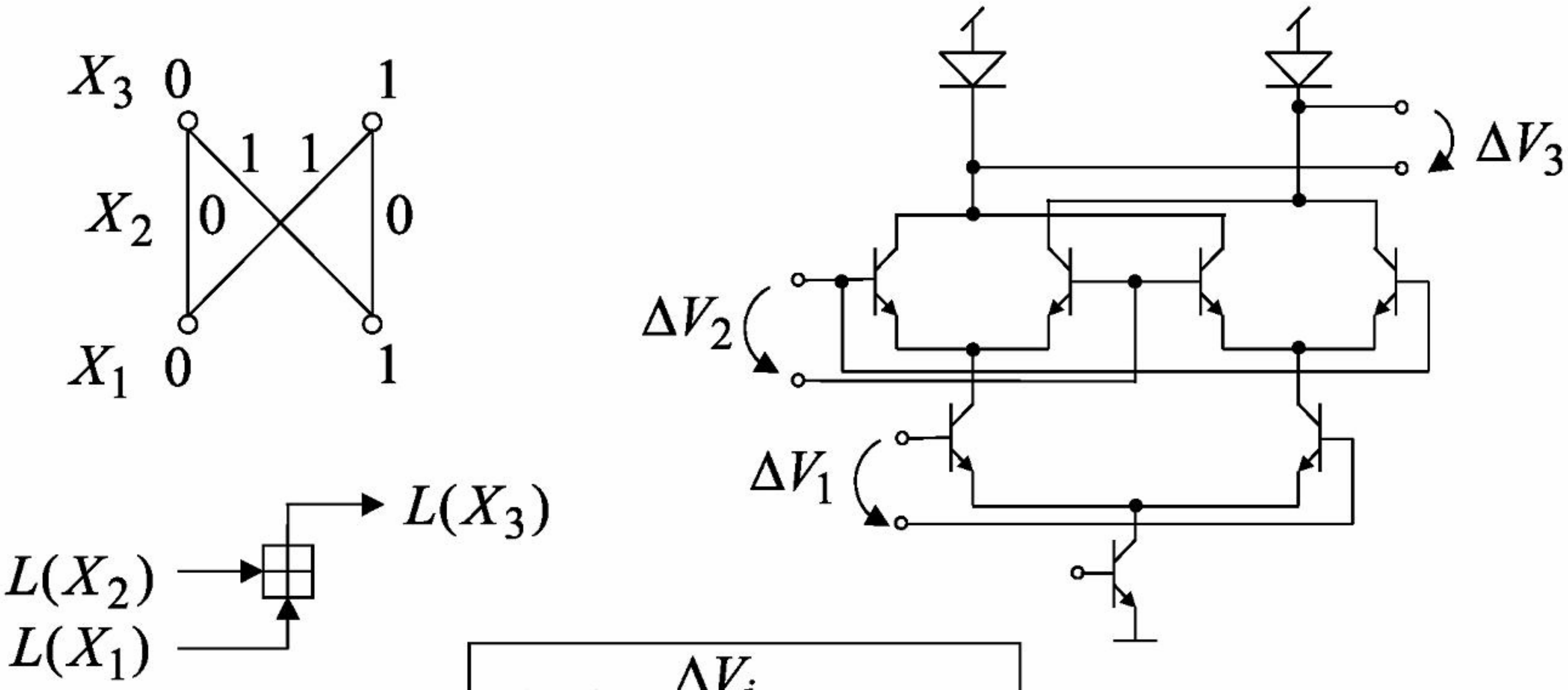
Parity Check Matrix

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$



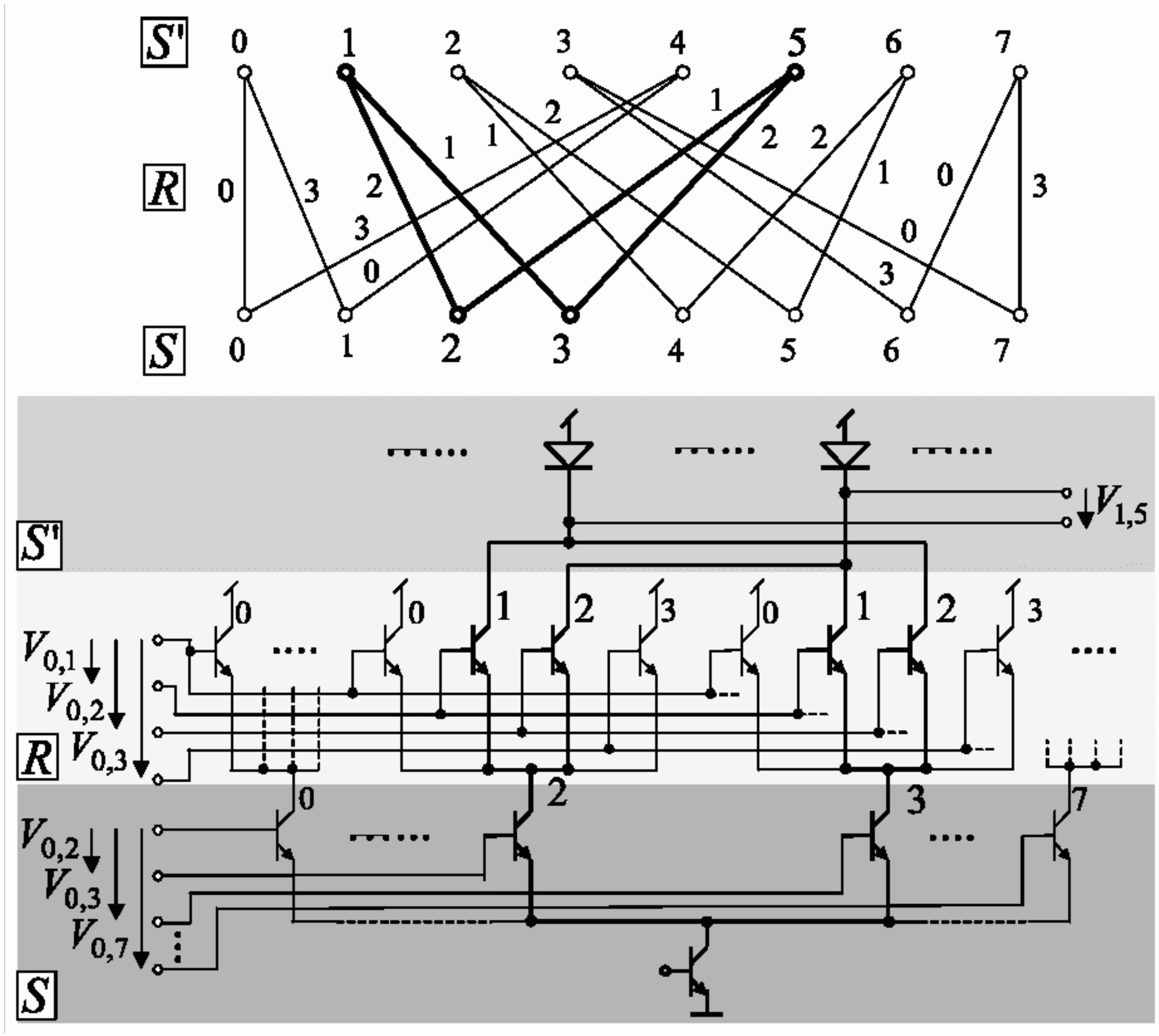
Trellis Based Decoder

Special case: M=1 (memory)



$$L(X_i) = \frac{\Delta V_i}{V_T}, \quad i = 1,2,3$$

Trellis Based Decoder: Structure



S' – 2^M Transistors

R – $2^M 2^N$ Transistors

S – 2^M Transistors

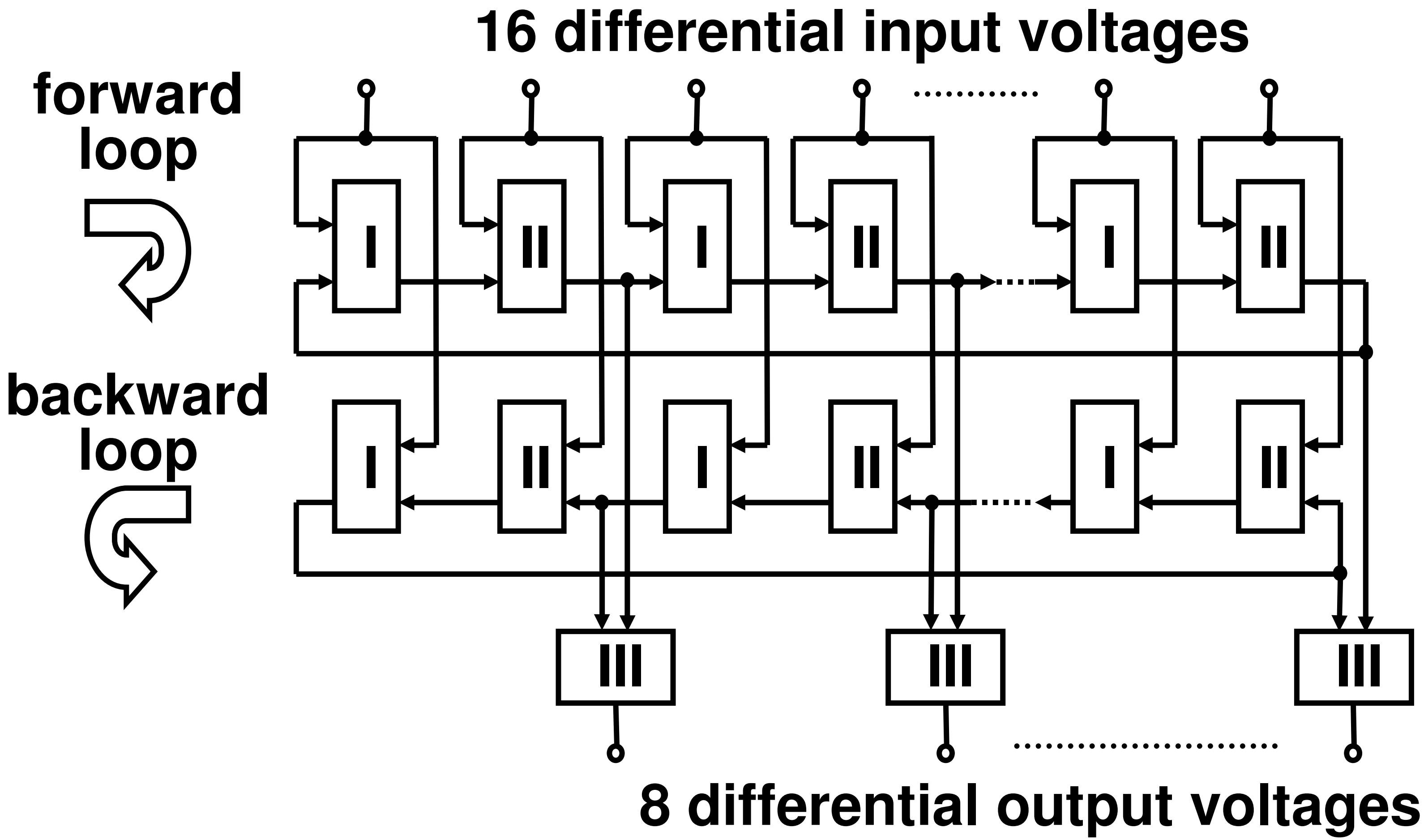
M=3

$g^{(1)}(D)=1+D^2+D^3$

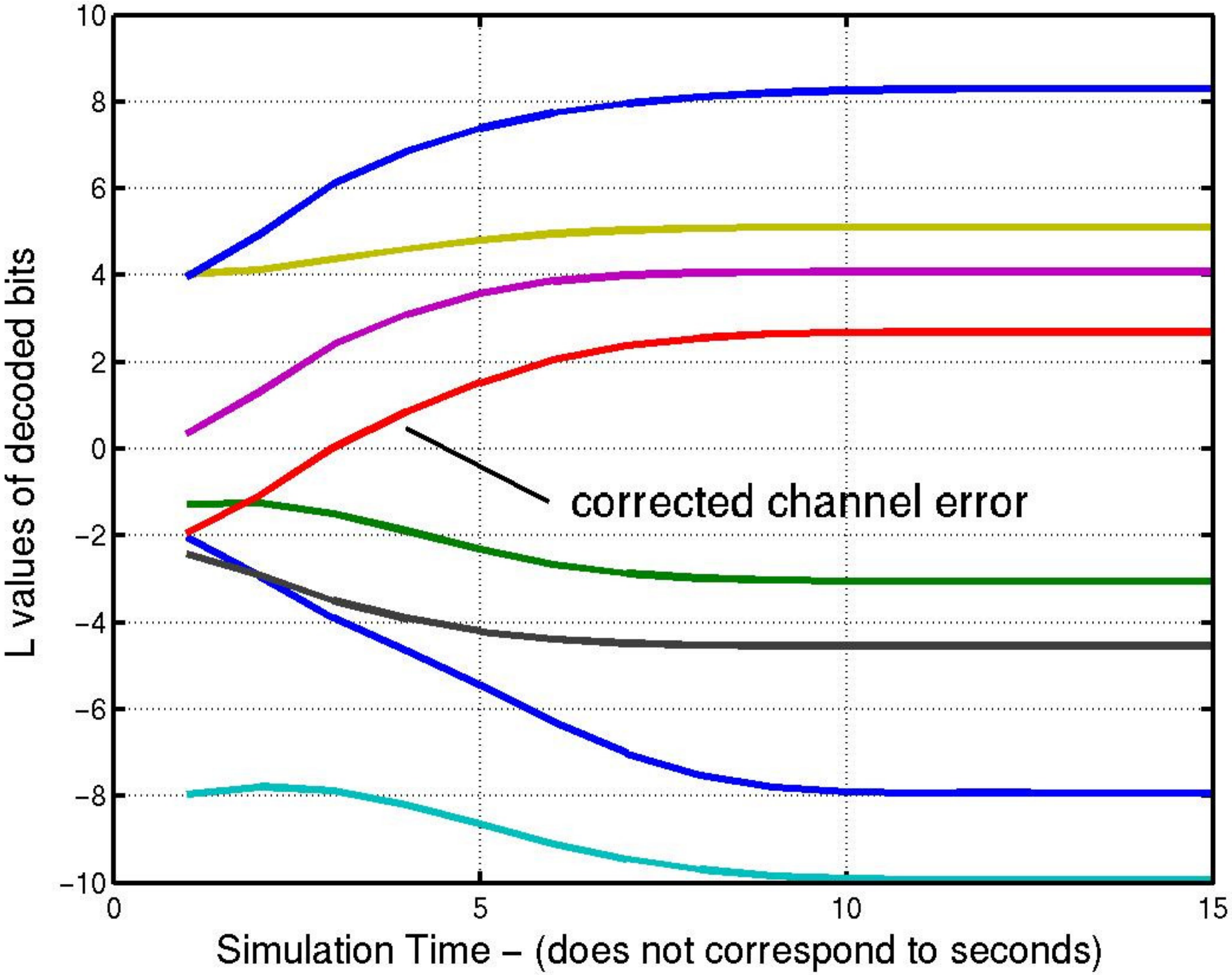
$g^{(2)}(D)=1+D+D^3$

Decoder Network

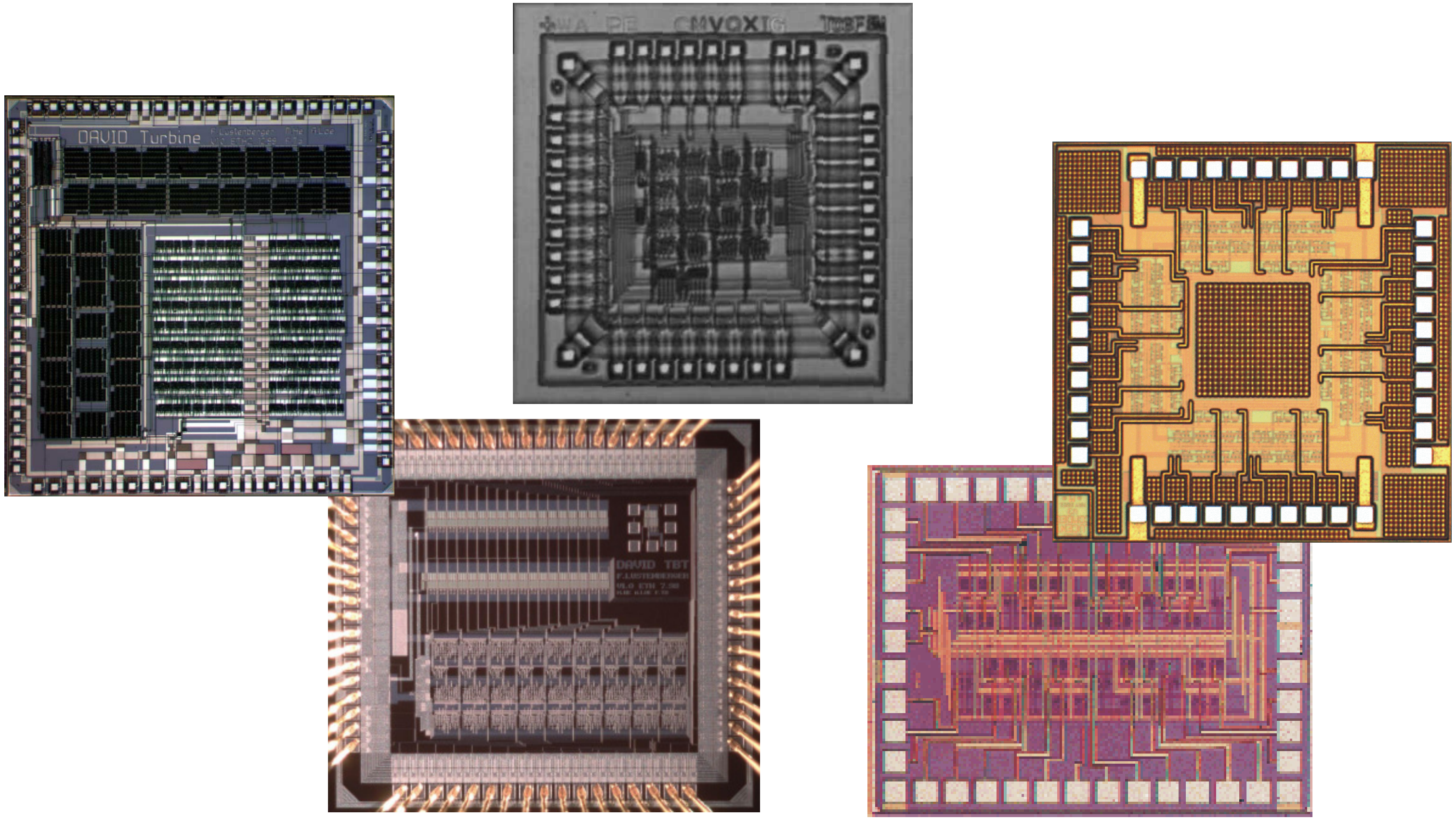
(16,8) tailbiting code, APP Decoder



Simulation Results - Settling Time



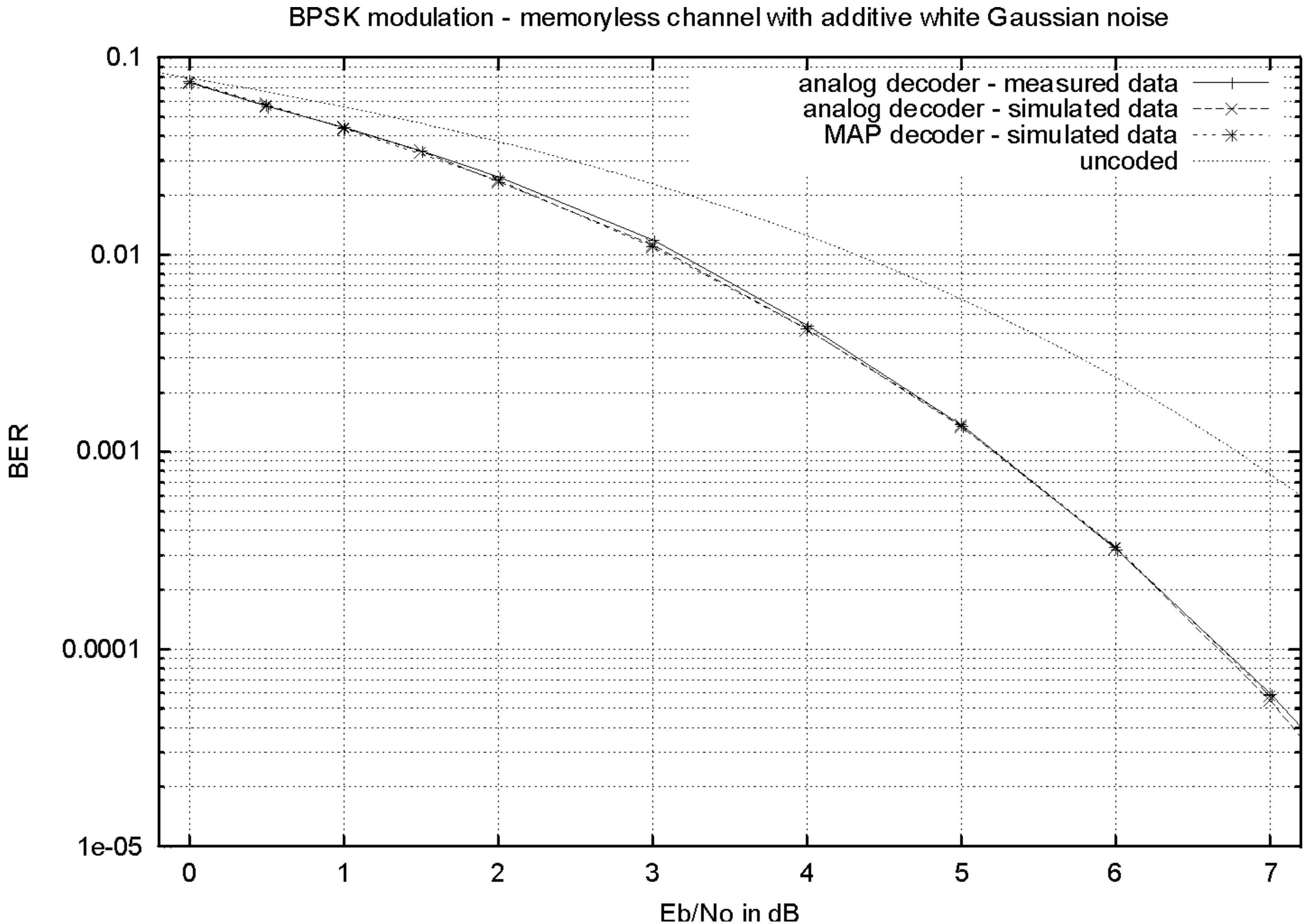
Implementation Examples



Implementation Examples

	TUM Lucent		ETH Endora	University of Utah
code	(16,8,3), m=1		(18,9,5), m=2	(8,4,4) ext. Hamming
process	0.25μm BiCMOS	IBM6HP	0.8μm BiCMOS	0.5μm CMOS
chip area	1.68mm^2	1.64mm^2	7.28mm^2	2.25mm^2
transistor # bipolar # nMOS # pMOS total # per inf.bit&state	 441 356 - 49.8	 54.4	 940 - 650 44.17	 - n.a. n.a. n.a.
supply	3.3V	3.3V	5V	n.a.
bias per block	80μA (nMOS) 200μA (bipolar)	 400μA	 200μA or 50μA	 10μA
power consumption	20mW	150mW	98mW @ 200μA 50mW @ 50μA	3.8mW
power per inf.bit&state	 1.25mW	 9.4mW	2.72mW @ 200μA 1.38mW @ 50μA	 0.32mW
measurements				
channel	AWGN	AWGN	BSC	AWGN
speed	160Mbit/s	10 Gbit/s	100Mbit/s	10Mbit/s
BER	yes	tbd.	n.a.	n.a.

Simulation Results (BER)



Conclusions

Open Questions

- Place interleaver cross-connecting network on chip
- Frame synchronization
- „Timing recovery“
- D/A conversion of the input values
- Storage of analog values

Usage

- Optical Transmission Systems
- Magnetic Recording

Thank you!

Any Questions?