

## Turbo Product Code Tutorial

### IEEE 802.16 Presentation Submission Template (Rev. 8)

Document Number:

IEEE 802.16t-00/01

Date Submitted:

2000-05-01

Source:

David Williams

Advanced Hardware Architectures, Inc.

2365 NE Hopkins Ct.

Pullman, WA 99163

Voice: 509-334-1000

Fax: 509-334-9000

E-mail: [davew@aha.com](mailto:davew@aha.com)

Venue:

IEEE 802.16 Session #7 Monday night tutorial session (2000-05-01)

Base Document:

none

Purpose:

Provide information on the state of TPC technology

Notice:

This document has been prepared to assist IEEE 802.16. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.

Release:

The contributor grants a free, irrevocable license to the IEEE to incorporate text contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by IEEE 802.16.

IEEE 802.16 Patent Policy:

The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures (Version 1.0) <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, if there is technical justification in the opinion of the standards-developing committee and provided the IEEE receives assurance from the patent holder that it will license applicants under reasonable terms and conditions for the purpose of implementing the standard."

Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:r.b.marks@ieee.org> as early as possible, in written or electronic form, of any patents (granted or under application) that may cover technology that is under consideration by or has been approved by IEEE 802.16. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/letters>.

# **Turbo Product Code Tutorial**

**May 1 2000**

**David Williams  
509-334-1000  
davew@aha.com**

# Agenda

- ◆ **Forward Error Correction Tutorial**
- ◆ **Turbo Product Codes**
- ◆ **Galaxy Simulation SW demonstration**
- ◆ **Open Discussion**
- ◆ **Summary and Conclusions**

# Forward Error Correction

# Typical System Design Goals

- ◆ Maximize data rate
- ◆ Maximize data reliability
- ◆ Minimize required transmission energy
- ◆ Minimize required bandwidth
- ◆ Minimize system complexity (cost)

**Forward Error Correction can be instrumental in helping meet these goals!**

# Definitions

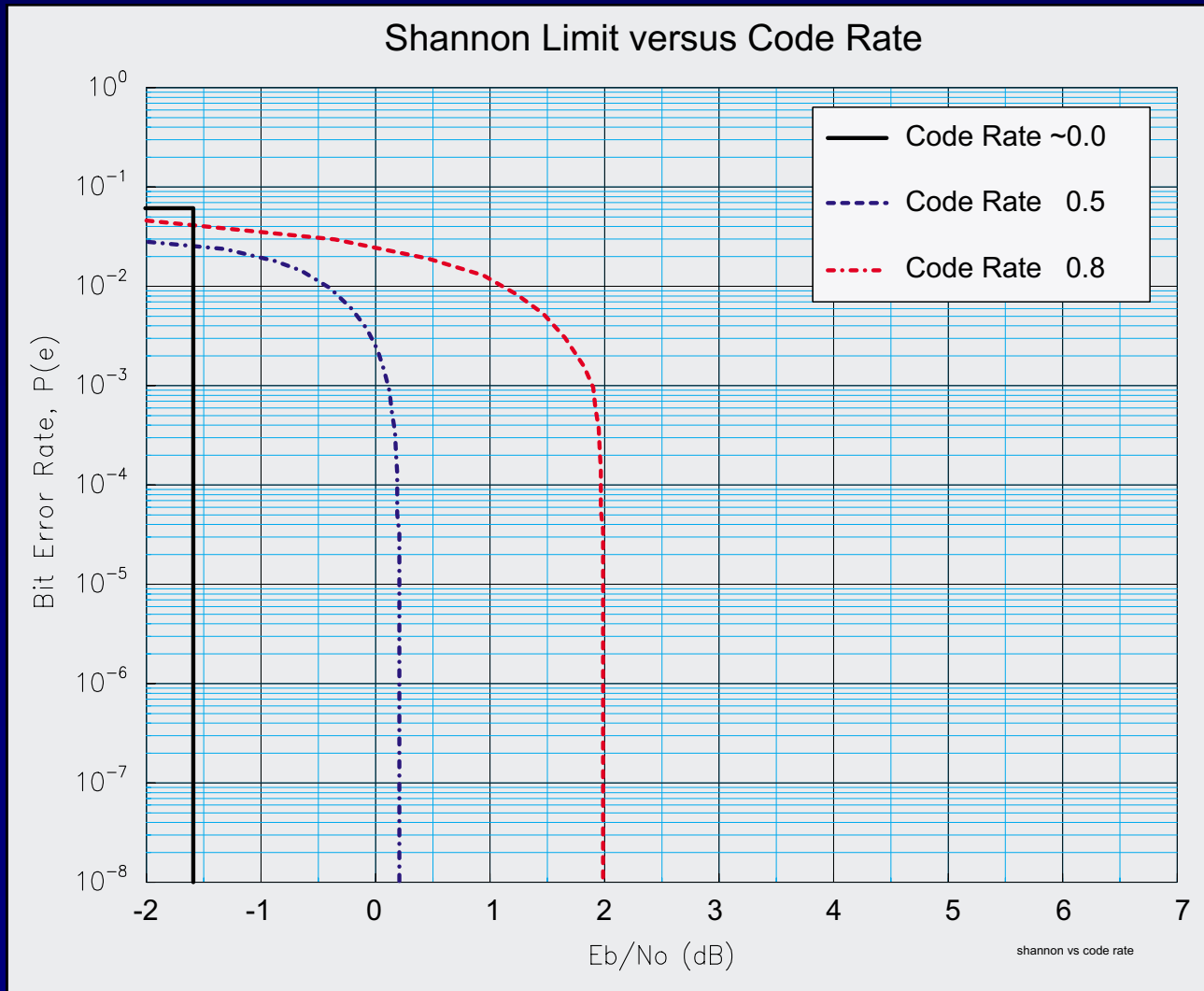
- ◆ **Forward Error Correction Coding**
  - The addition of redundancy to a message through encoding prior to transmission
- ◆ **Code Rate**
  - Ratio of data bits / total bits transmitted
- ◆ **Shannon Channel Capacity**
  - Theoretical bound for channel capacity for a given modulation and code rate

# Error Correction Capabilities

- ◆ The minimum distance or “Dmin” of an error correction code characterizes the codes strength.
- ◆ The error correcting capability, “t”, of a code is defined as the maximum number of guaranteed correctable errors per codeword:

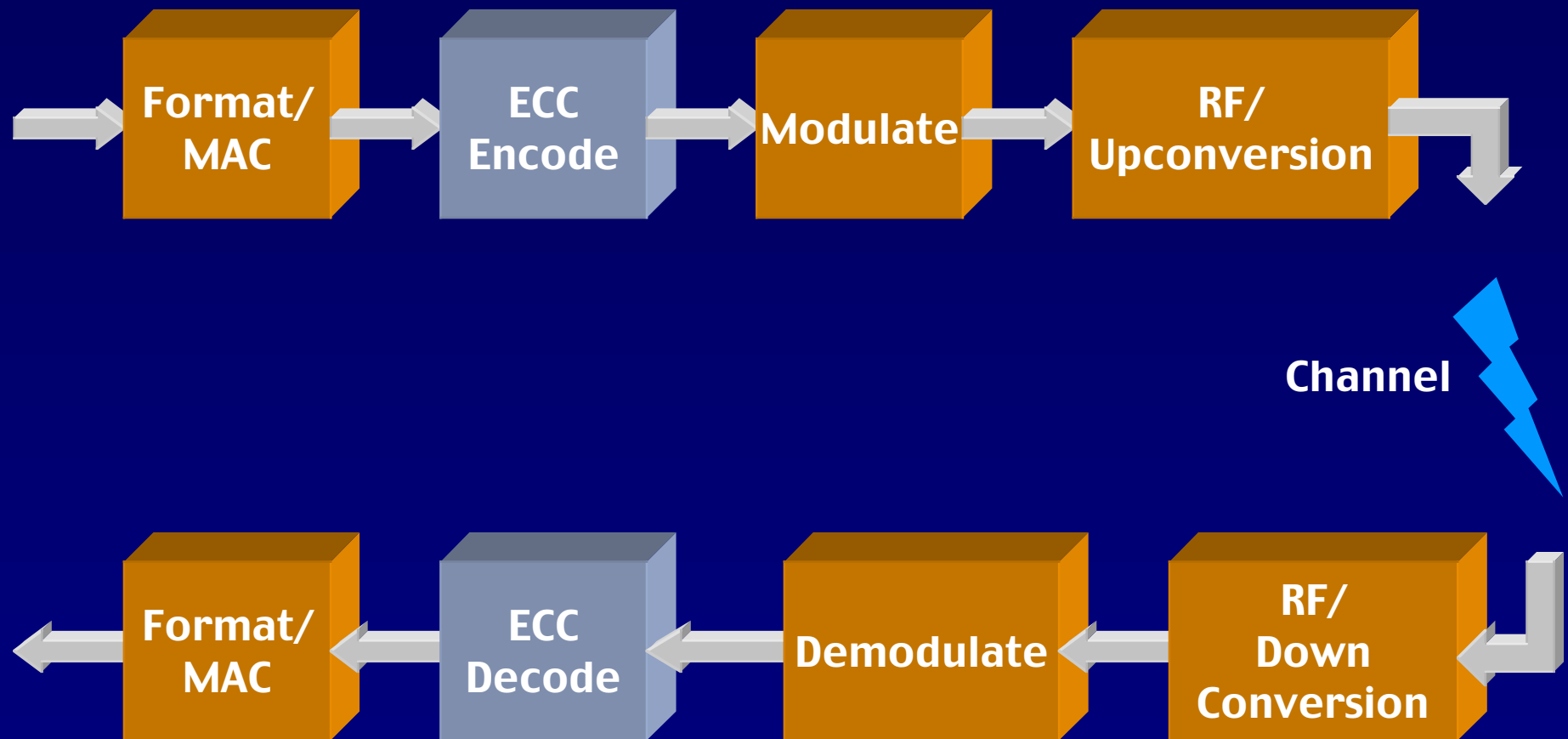
$$t = \left\lfloor \frac{D_{\min} - 1}{2} \right\rfloor$$

# Shannon Capacity vs. Code Rate





# Error Correction Coding



# Why Use Error Correction?

*Designers can choose between levels of improved data reliability, reduced systems costs or increases in range*



**3 dB of coding gain can:**

- reduce the required bandwidth by 50% or
- increase data throughput by a factor of 2 or
- increase range by 40% or
- reduce antenna size by 30% or
- reduce transmitter power by a factor of 2

**Bottom line...reduced cost or increased value**

# Error Correction Codes

- ◆ **Block Codes**
  - Hamming
  - BCH
  - Reed–Solomon
- ◆ **Convolutional Codes (Viterbi)**
- ◆ **Trellis Codes**
- ◆ **Concatenated**
  - Viterbi/Reed–Solomon
- ◆ **Turbo Convolutional Codes (TCC)**
- ◆ **Turbo Product Codes (TPC)**

# Hamming Codes

- ◆ **Linear and systematic**
- ◆ **Specified as  $(n,k)$** 
  - **$n$  is the encoded number of bits**
  - **$k$  is the number of information bits**
- ◆ **Example:  $(7,4)$** 
  - **$d_1 d_2 d_3 d_4 e_1 e_2 e_3$**
- ◆ **Easy to both encode and decode**

# Hamming Codes

- ◆ **Hamming codes**
  - **$D_{min} = 3$**
- ◆ **Extended Hamming codes add a parity bit**
  - **$D_{min} = 4$**
- ◆ **TPCs utilize extended Hamming codes and parity codes to construct Product Codes**
  - **$D_{min} \gg$  than constituent codes**
- ◆ **Iterative decoding enhances performance**
  - **Turbo Product Codes**

# Turbo Product Code History

- ◆ **Product codes first described by Elias (1954)**
- ◆ **Iterative decoding of product codes described by Tanner (1981), Lin & Costello (1983) and others**
- ◆ **Although described in the literature, commercial implementation had to wait for a cost efficient SISO decoder algorithms**
- ◆ **Efficient SISO algorithms have made commercially viable TPCs possible**
- ◆ **AHA4501 introduced in November 1998**

# Turbo Product Codes

- ◆ Turbo Product Codes (TPCs) are based on block codes, not convolutional codes
- ◆ TPCs are built on a 2 or 3 dimensional arrays of extended Hamming codes
  - Encoding is done in a single iteration
  - Minimum distance of a 2-D product code is square of constituent code; for a 3-D code, cubed
  - Thus minimum distance is:
    - ✓ 16 for 2-Dimensional codes
    - ✓ 64 for 3-Dimensional codes

# 2D Product Code Example

- ◆ **(8,4) x (8,4) Code**
  - Code is systematic
  - D represents input data
  - E represents ECC bits
- ◆ **3D codes follow the same concept, but in three dimensions**

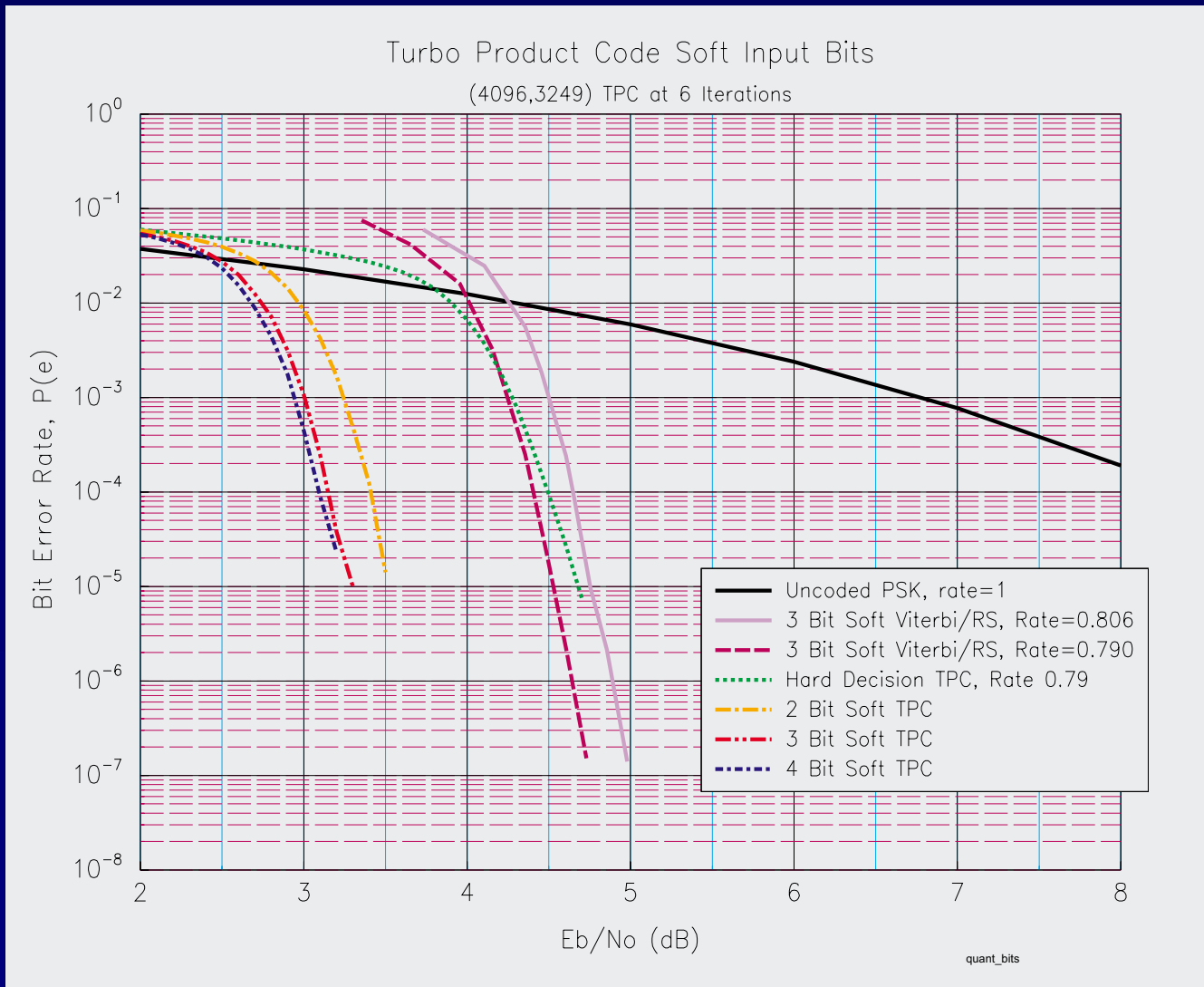
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E



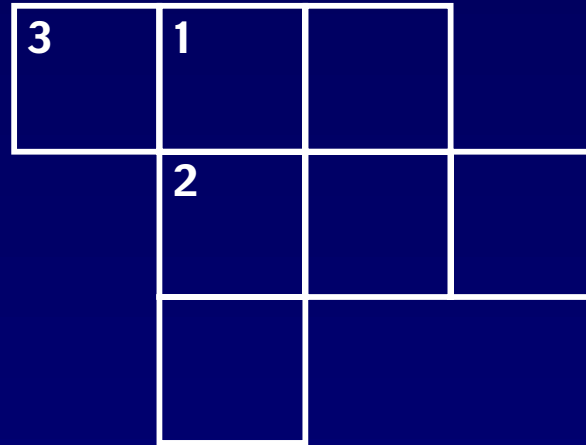
# Hard vs Soft Decision Decoding

- ◆ TPCs will work with hard decision or soft decision decoding
- ◆ Two bit soft decision decoding typically picks up 2 dB of additional coding gain
- ◆ Additional soft bits can add about 1/2 dB more
- ◆ Soft decoder input requires a “soft metric”
  - BPSK/QPSK proportional to I/Q voltage
  - Higher order constellations require LLR computation

# Soft vs. Hard Decision Decoding



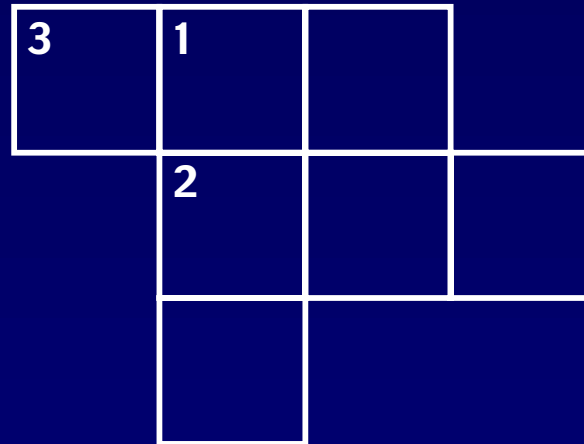
# Iterative Decoding



**Down**

**1. Family pet**

# Iterative Decoding



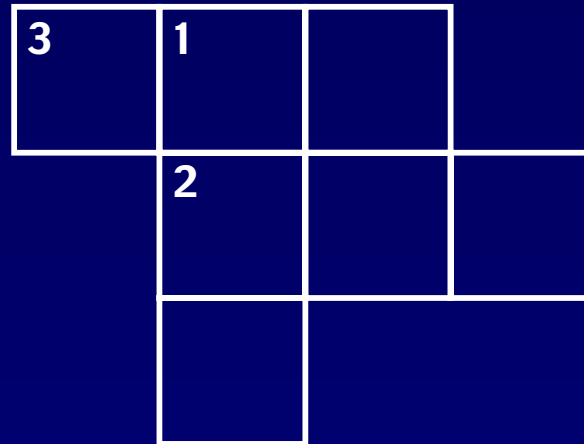
## Down

1. Family pet

## Across

2. Advanced Hardware Architectures

# Iterative Decoding



## Down

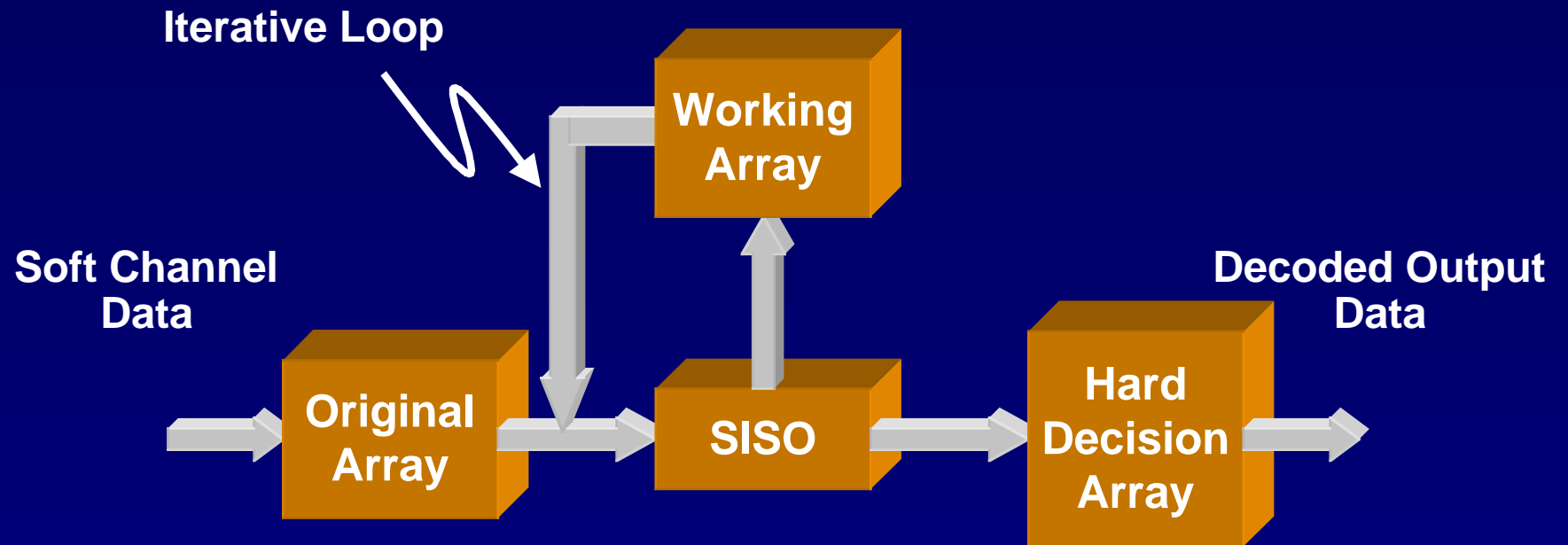
1. Family pet

## Across

2. Advanced Hardware Architectures

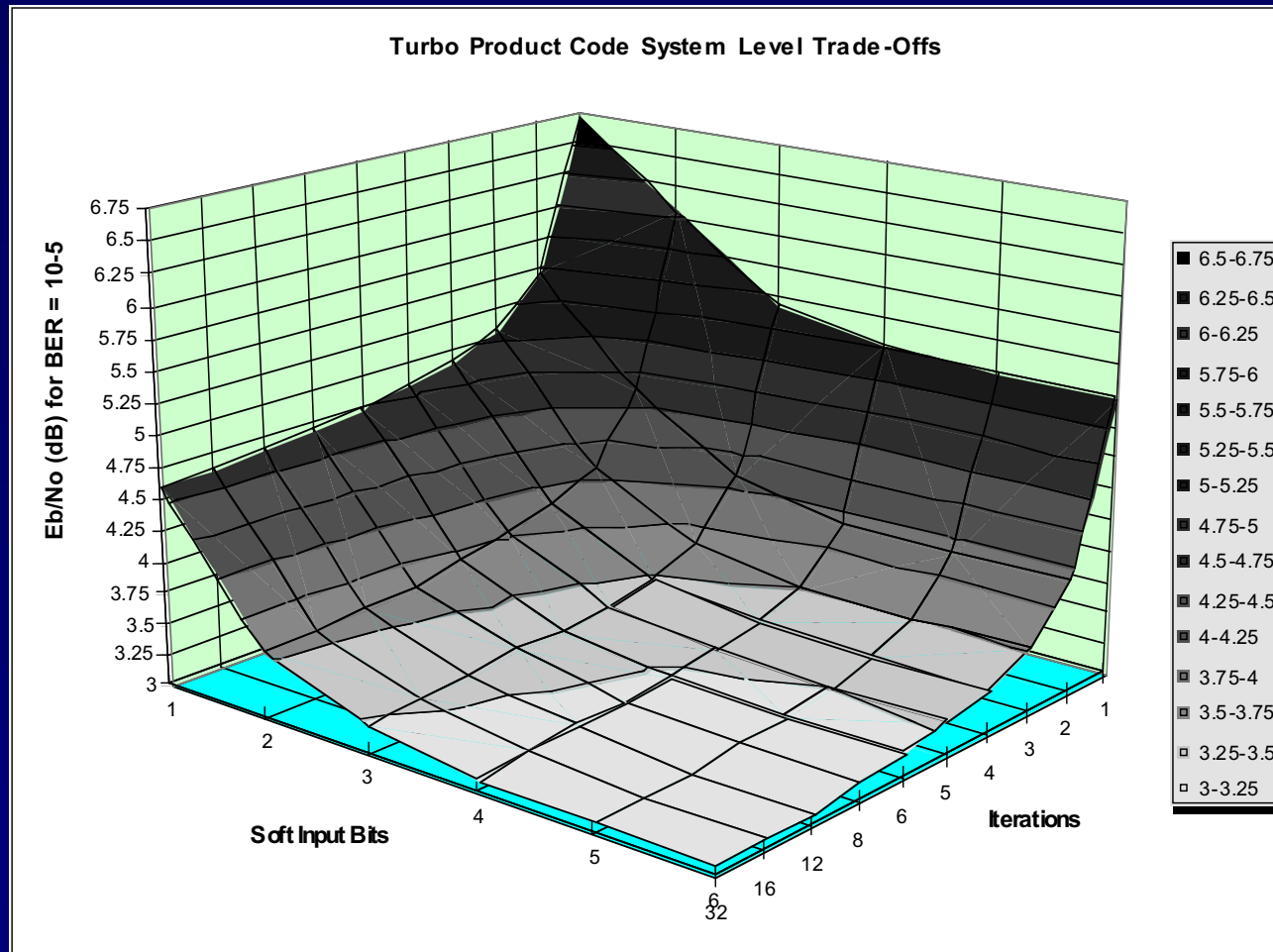
3. Cyclical Redundancy Check

# TPC Decoding



Each Iteration Decodes all Rows, then all Columns

# System Level Tradeoffs



(64,57)x(64,57) Rate 0.793 Code

# Burst Error Performance

- ◆ **Given that extended Hamming codes can only correct a single bit error how well can a TPC handle a burst of errors?**



# Burst Error Performance

- ◆ Product code array is the key!

D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E

# Burst Error Performance

- ◆ Product code array is the key!

D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E

# Burst Error Performance

- ◆ Product code array is the key!

D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
D	D	D	D	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E
E	E	E	E	E	E	E	E

# Burst Error Performance

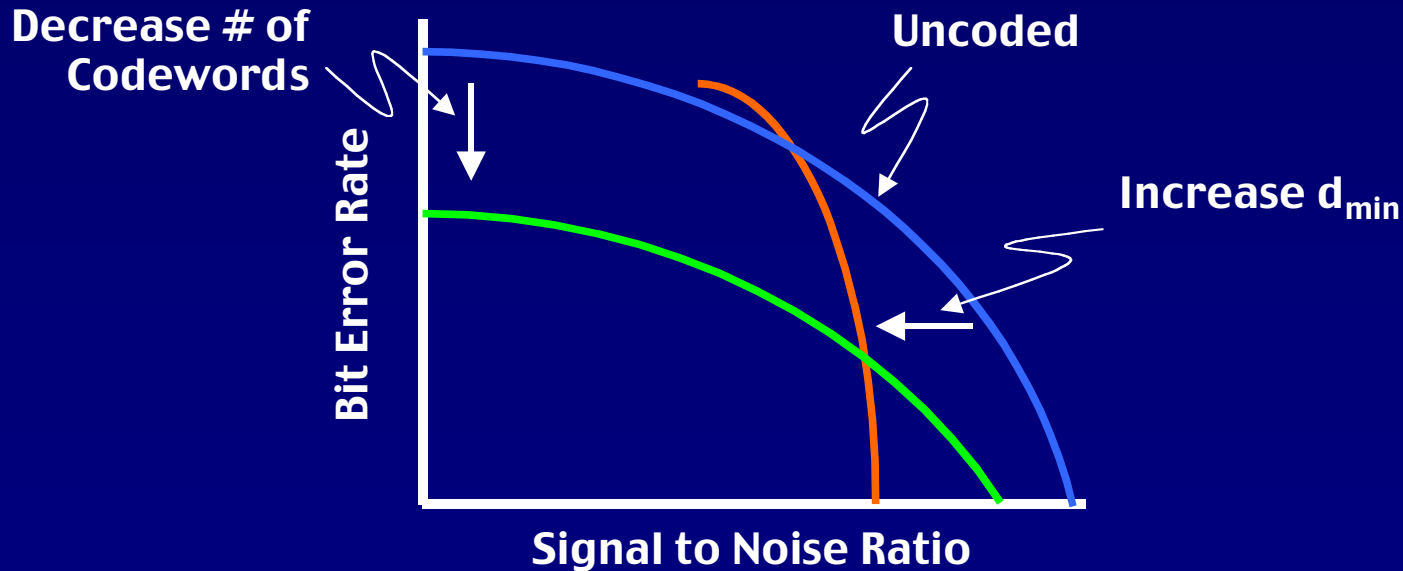
- ◆ **Data scrambling enhances burst error performance**
- ◆  **$(64,57)^2$  code performance (helical scrambling)**
  - **Tolerates 384 burst errors in each code block**
  - **This burst represents 9.4% of the bits in the block**
  - **A 128 bit burst event every block shifts the BER curve by only 1.2 dB**
  - **AWGN channel model used in addition to burst**

# Burst Error Performance

- ◆ **3 D codes provide even more performance**
- ◆ **(16,11)<sup>3</sup> code performance (helical scrambling)**
  - **Handles 1024 burst errors in each code block**
  - **This burst represents 25% of the bits in the block**
  - **A 128 bit burst event every block shifts the BER curve by only 0.5 dB**
  - **For example, BER < 10<sup>-6</sup> at Eb/No of 2 dB with a 128 bit burst error event every block**

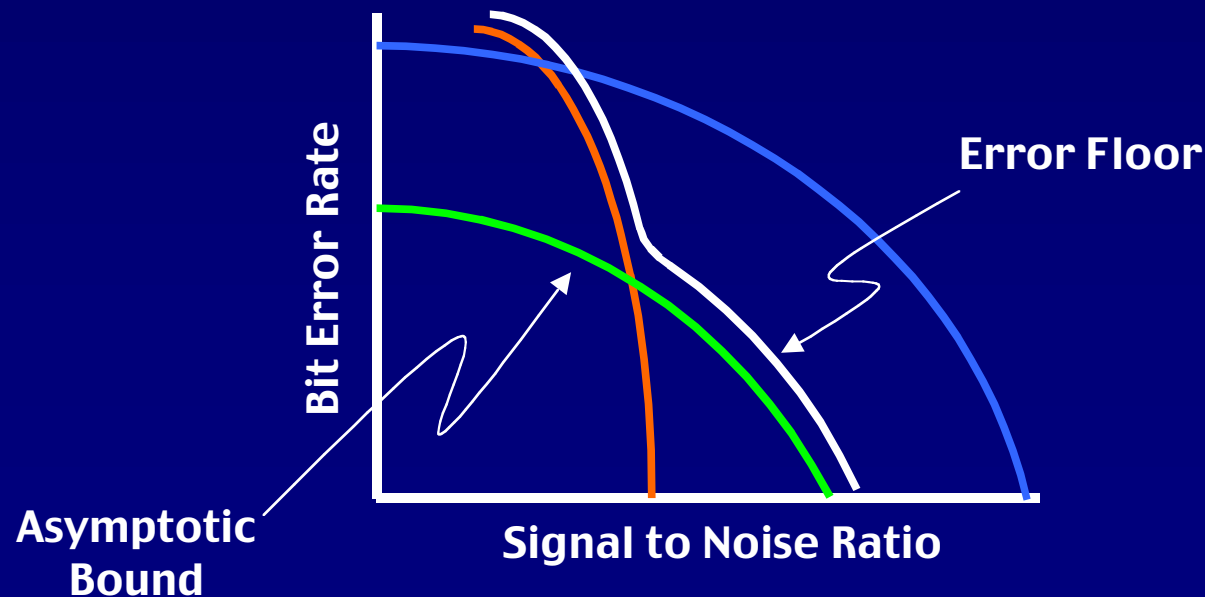
# Asymptotic Bounds

- ◆ Minimum Distance ( $d_{\min}$ ) Determines Slope of Asymptotic Bound
- ◆ Number of Codewords at  $d_{\min}$  Determine Position of Bound



# TCC Distance Structure

- ◆ Decreases Number of Codewords at  $d_{\min}$
- ◆ However,  $d_{\min}$  is not High (Especially After Puncturing)
- ◆ Can Cause Characteristic 'Error Floor'

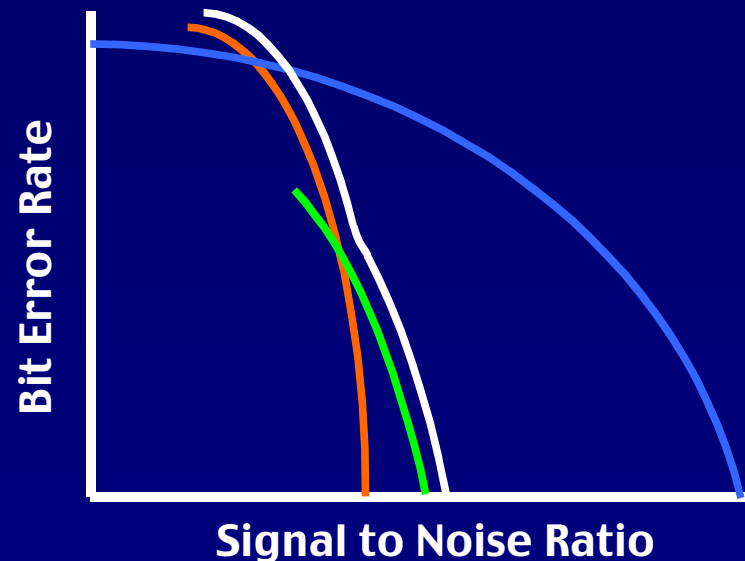


# TPC Distance Structure

- ◆ Distance Properties of Product Code are Equal to the Product of the Constituent Codes

$(32,26)^2$  Product Code  
 $d_{\min} = 16$

$(16,11)^3$  Product Code  
 $d_{\min} = 64$



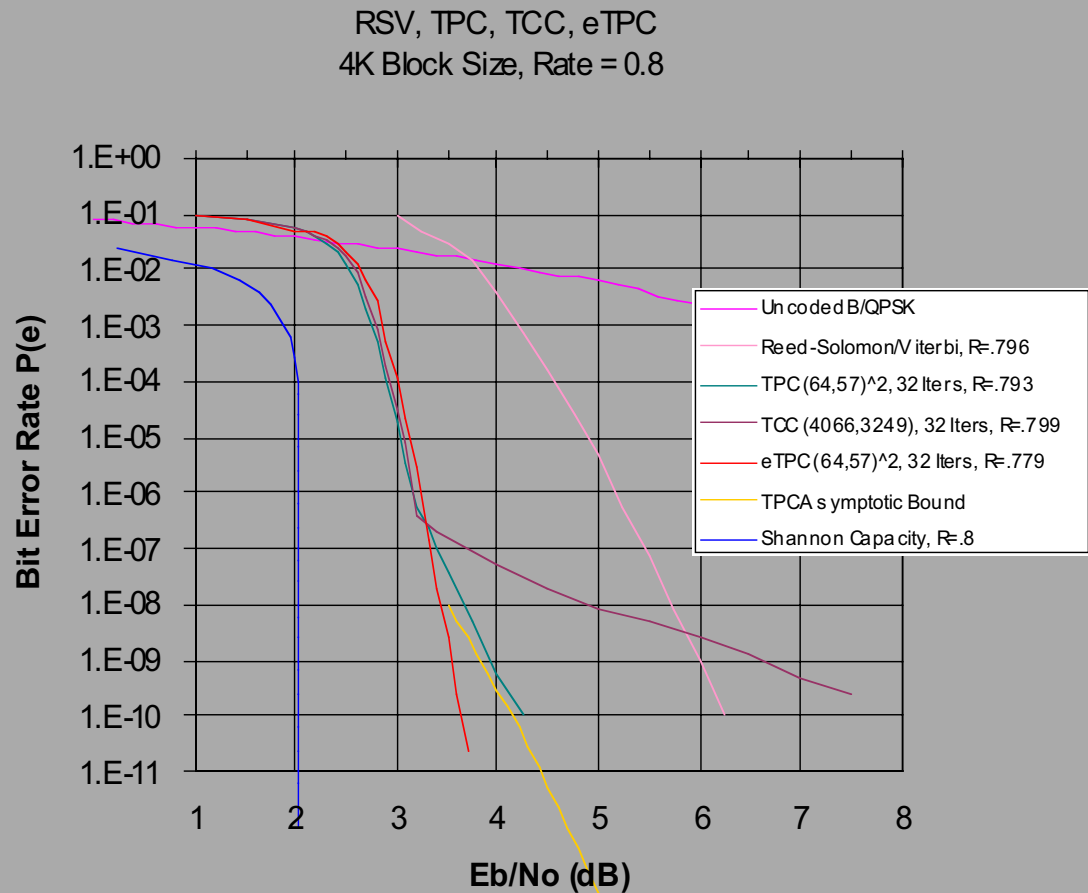


# Error floor vs. Flare

- ◆ Because of high  $D_{min}$  (typ 16 or greater), there is no error floor for TPCs
- ◆ Depending on the TPC code used:
  - No Flare
  - Minor flaring starting @ about  $10^{-7}$ 
    - Flare is  $< 0.3$  dB/decade
  - Minor flaring starting @ about  $10^{-12}$
- ◆ Flare is predictable
  - Block size
  - $D_{min}$

# Turbo Product Codes

## Performance Comparison



# Competitive Comparisons

<i>Attribute</i>	<i>TPC</i>	<i>RS</i>	<i>Viterbi</i>	<i>RS-Viterbi</i>	<i>TCC</i>
Coding Gain					
@ BER of $10^{3,-4}$	4	2	4	1	5
$10^{6,-7}$	5	3	2	3	4
$<10^{10}$	4-5	3	2	4	1
Latency (bits)					
$<256$	4	2	5	1	5
1024 – 204	5	3-4	3	3	5
$>4096$	5	3	2	4	5
Complexity	3	5	4	3	1-2
IP	4-5	5	5	5	2
Code Rate (flex/range)	5	2	4	3	5
Max Data Rate	4	5	3	3	2

# Why Use TPCs?

- ◆ **Excellent performance at high code rates, no puncturing required**
- ◆ **Low complexity relative to coding gain**
  - **Lower cost**
  - **Lower power consumption**
- ◆ **TPCs offer significant improvement over concatenated Reed–Solomon/Viterbi**
- ◆ **TPCs are available as standard products and as licensable cores**

# Flexibility

- ◆ **A single low cost TPC encoder/decoder can support code rates from 1/5 to 19/20**
  - **No puncturing required**
  - **Code change on the fly supports changing channel**
    - **Near zero latency, no “tail biting” required**
- ◆ **Readily adapted to most any constellation**
  - **See IEE paper (dgw,Nov99)**
- ◆ **Can support any packet size**
  - **See RAWCON paper (EH,Aug99)**
- ◆ **Available as standard ASICs or licensed Cores**
  - **VHDL or Verilog targeted to any foundry**
  - **Licenses available to end users and foundries**

# TPC Constituent Codes

- ◆ Constituent Codes can be mixed and matched to achieve desired code characteristics
- ◆ Two or three dimensions as desired

## Extended Hamming Codes

(128, 120)  
(64, 57)  
(32, 26)  
(16, 11)  
(8, 4)  
-

## Parity Codes

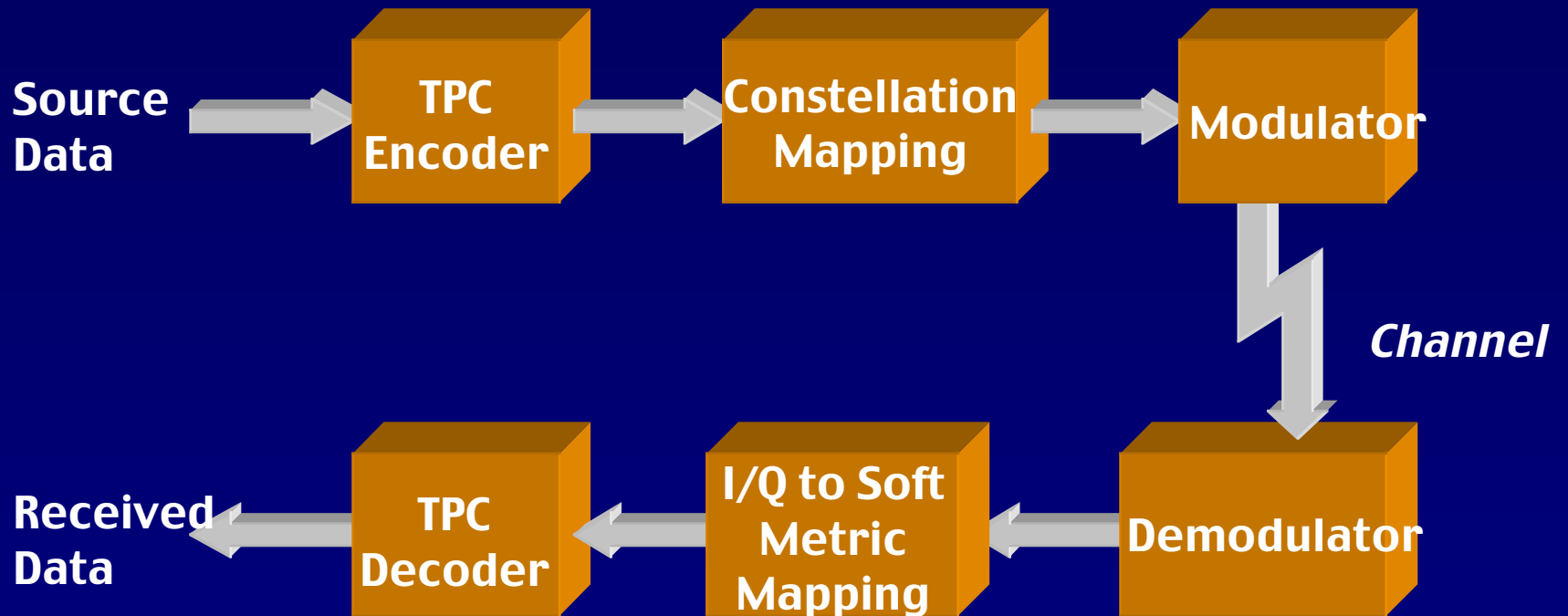
(128, 127)  
(64, 63)  
(32, 31)  
(16, 15)  
(8, 7)  
(4, 3)

# Representative Codes

- ◆ TPCs can provide a wide range of code rates and block sizes
- ◆ Code shortening enhances this flexibility
  - Shorten rows, columns, and/or bits

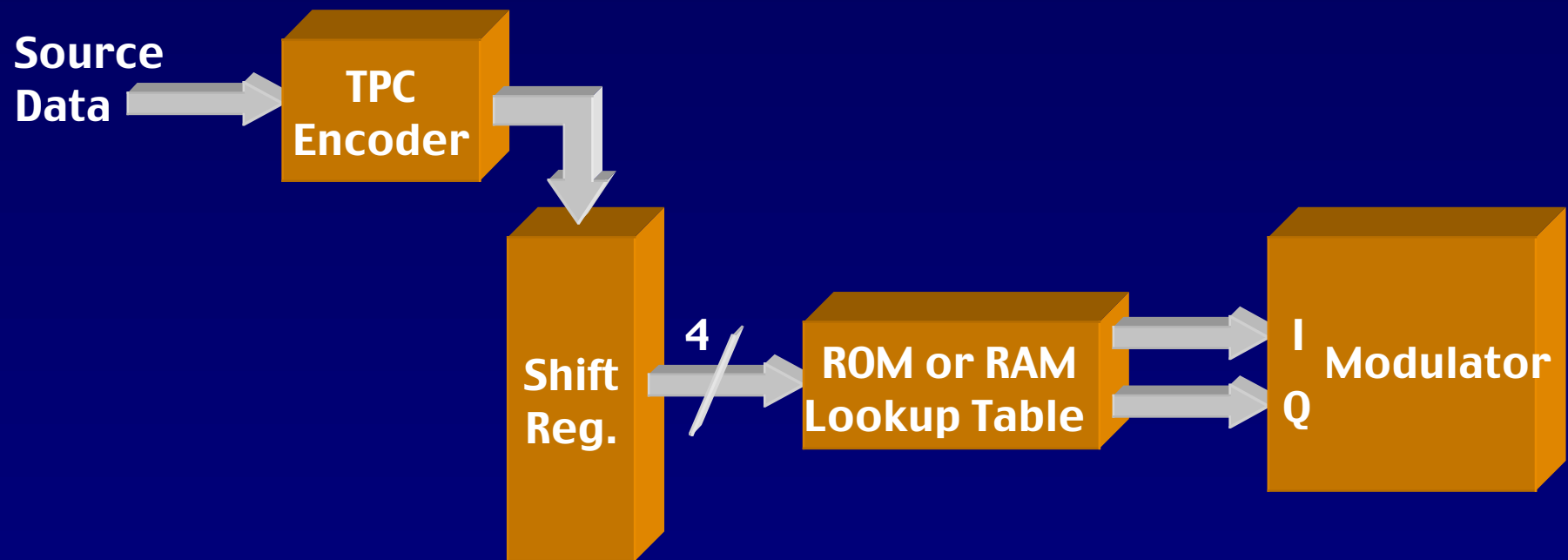
Product Code	Block Size	Code Rate
(128,127) x (128,127)	16,383	0.98
(128,120) x (128,127)	16,383	0.93
(64,57) x (32,26)	2,048	0.72
(32,26) x (16,15) x (8,7)	4,096	0.66
(16,11) x (16,11)	256	0.47
(16,11) x (16,11) x (16,11)	4,096	0.32

# Combining TPC Coding with Modulation

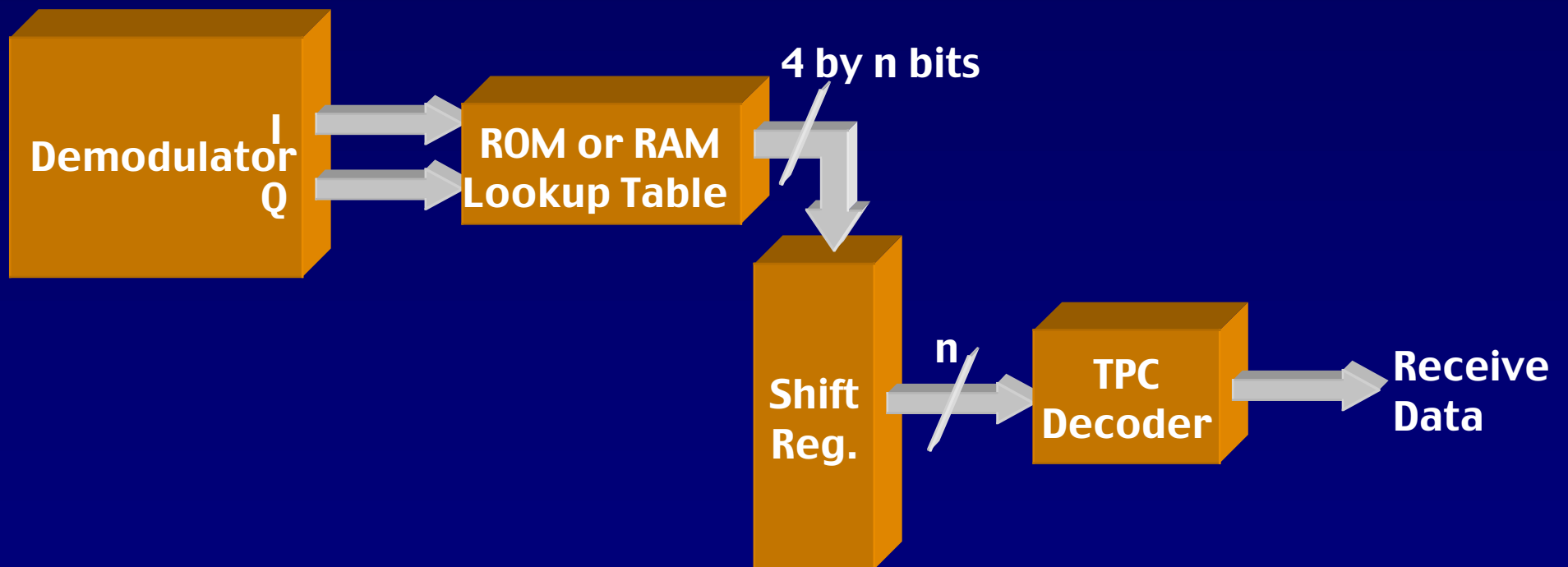




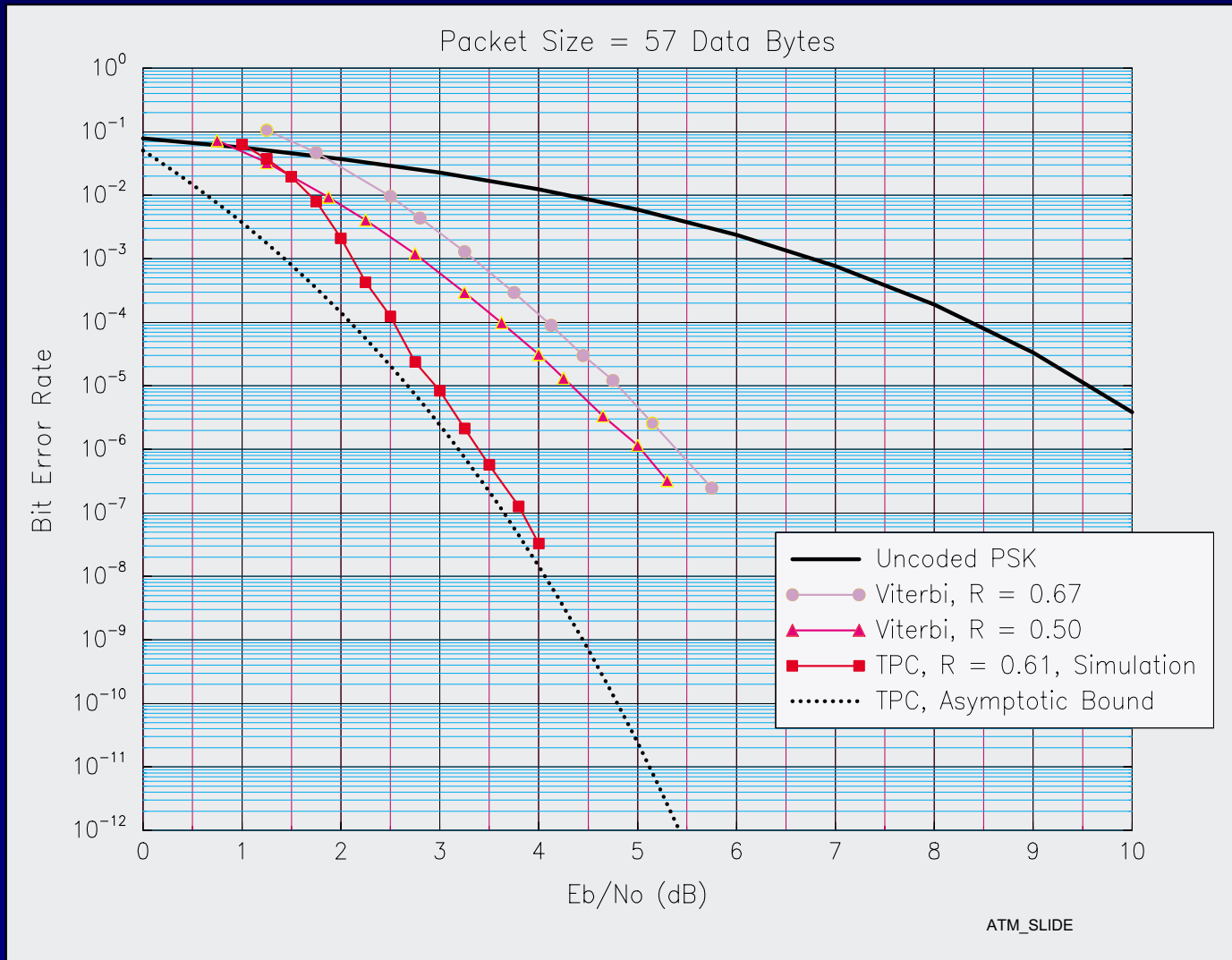
# Constellation Mapping



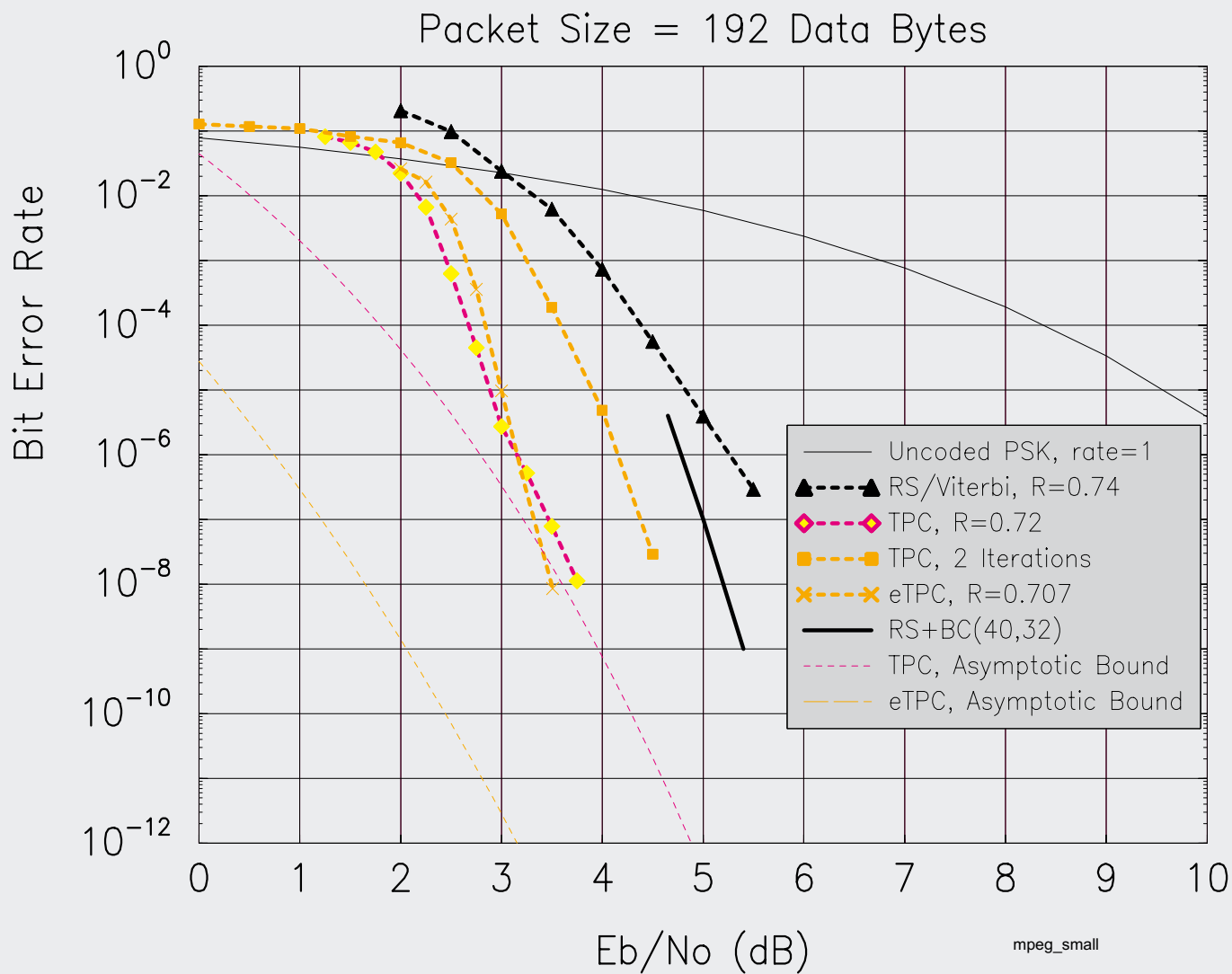
# I/Q Soft Metric Mapping



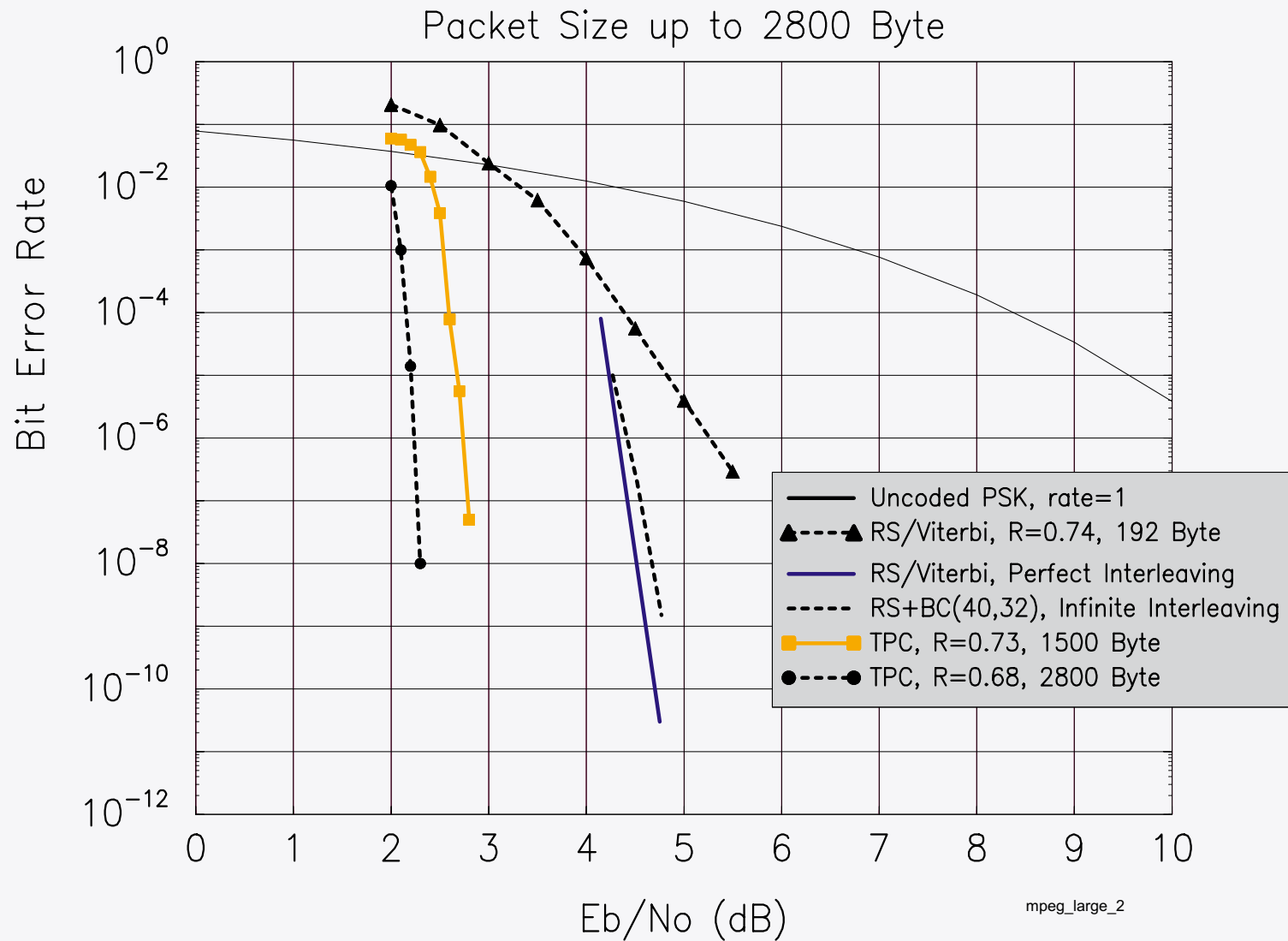
# ATM Performance



# MPEG Performance



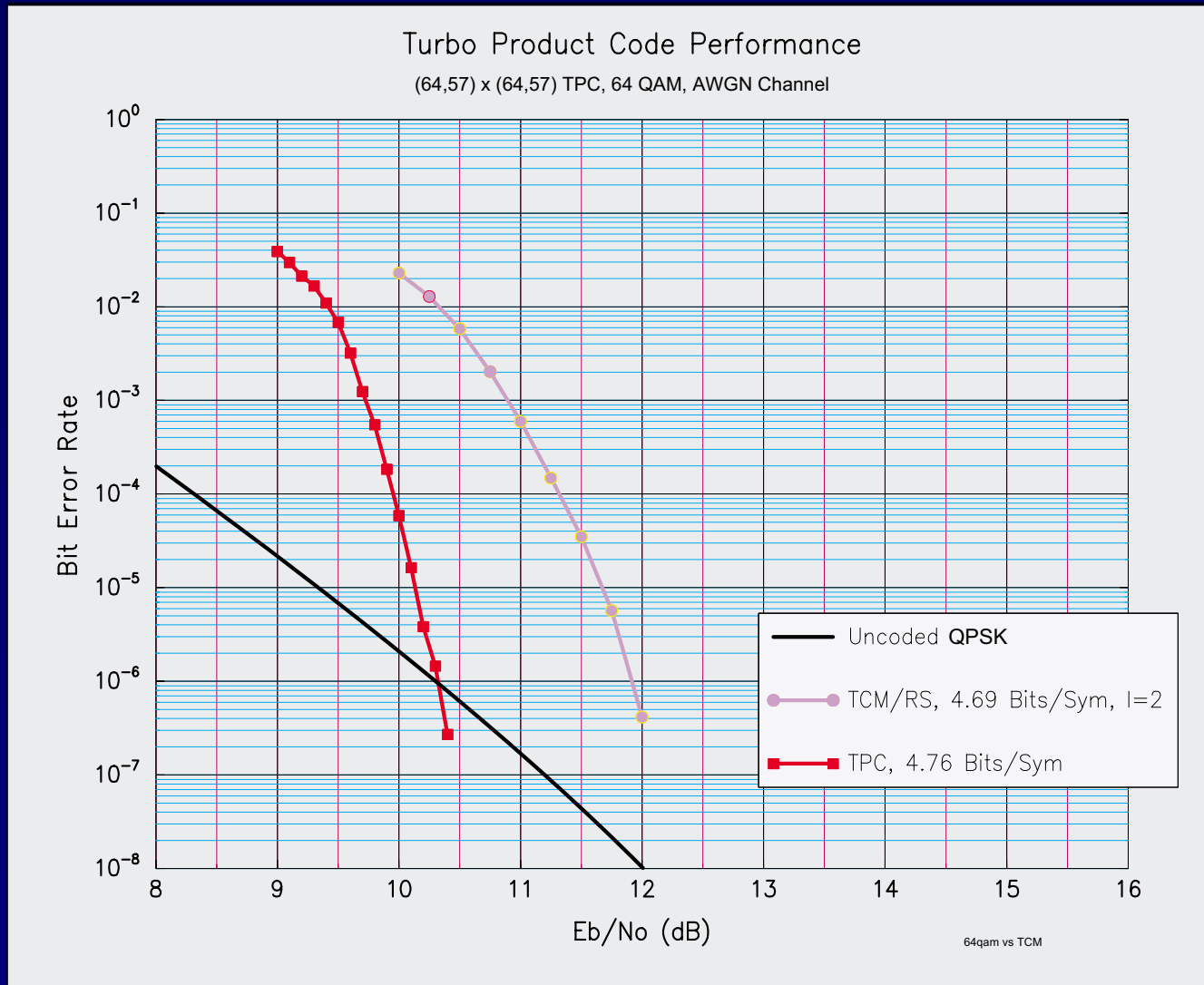
# Large Packet Size Plot



# Higher Order Modulation

- ◆ Constellation is Square 64-QAM
- ◆ TPC
  - $(64,57)^2 = (4096,3249)$
  - 4.76 Bits/Symbol
- ◆ Concatenated Code
  - Ungerboeck 16 State, 2-D TCM
  - (255,239), t=8 Reed-Solomon
  - Interleaver Depth of 2
  - 4.69 Bits/Symbol

# TPC vs TCM with RS



# AHA4501

- ◆ **Single Chip Turbo Product Code Encoder/Decoder**
- ◆ **No external memory required for encoding or decoding**
- ◆ **Up to 2 dB coding gain over concatenated Reed–Solomon/Viterbi**
- ◆ **Up to 3 dB coding gain over standalone RS**



# AHA4501

- ◆ **Channel rates for AHA4501 are up to 36 Mbit/sec**
  - **Multiple AHA4501 devices can be run in parallel with a simple multiplexor to achieve any desired speed**
- ◆ **Block sizes from 256 bits to 4096 bits**
- ◆ **AHA4501 handles all data formatting for 2 or 3 dimensional codes in a single chip**
  - **Half Duplex Encode/Decode**
  - **Helical Encoding On-chip**
  - **Support for external block synchronization**
  - **No external memory required**

# New TPC Products

- ◆ **Astro OC-3**
  - Supports OC-3 data rates (155 Mbits/s)
  - Block size up to 16K bits
  - Code rates from 1/4 to 0.98
  - ATM packet or user defined formats
  - Parallel data input/output, programmable
  - Supports code changing on the fly
  - Protos: June 2000 Production: September 2000
- ◆ **Astro LE**
  - Low Cost
  - Data rates to 25 Mbits/sec
  - Architectural definition stage
  - Protos: 4th Quarter, 2000

# Next Generation TPC Trends

SoC type logic replaces costly FPGA functions resulting in significant cost savings



- **Framing and packet synchronization**
- **Bit and block error rate monitoring**
- **Soft metric computation logic**
- **Encoder and decoder on one chip**
- **Code change on the fly**
- **Wide range of block size and code rates**

# Astro OC-3

## Applications

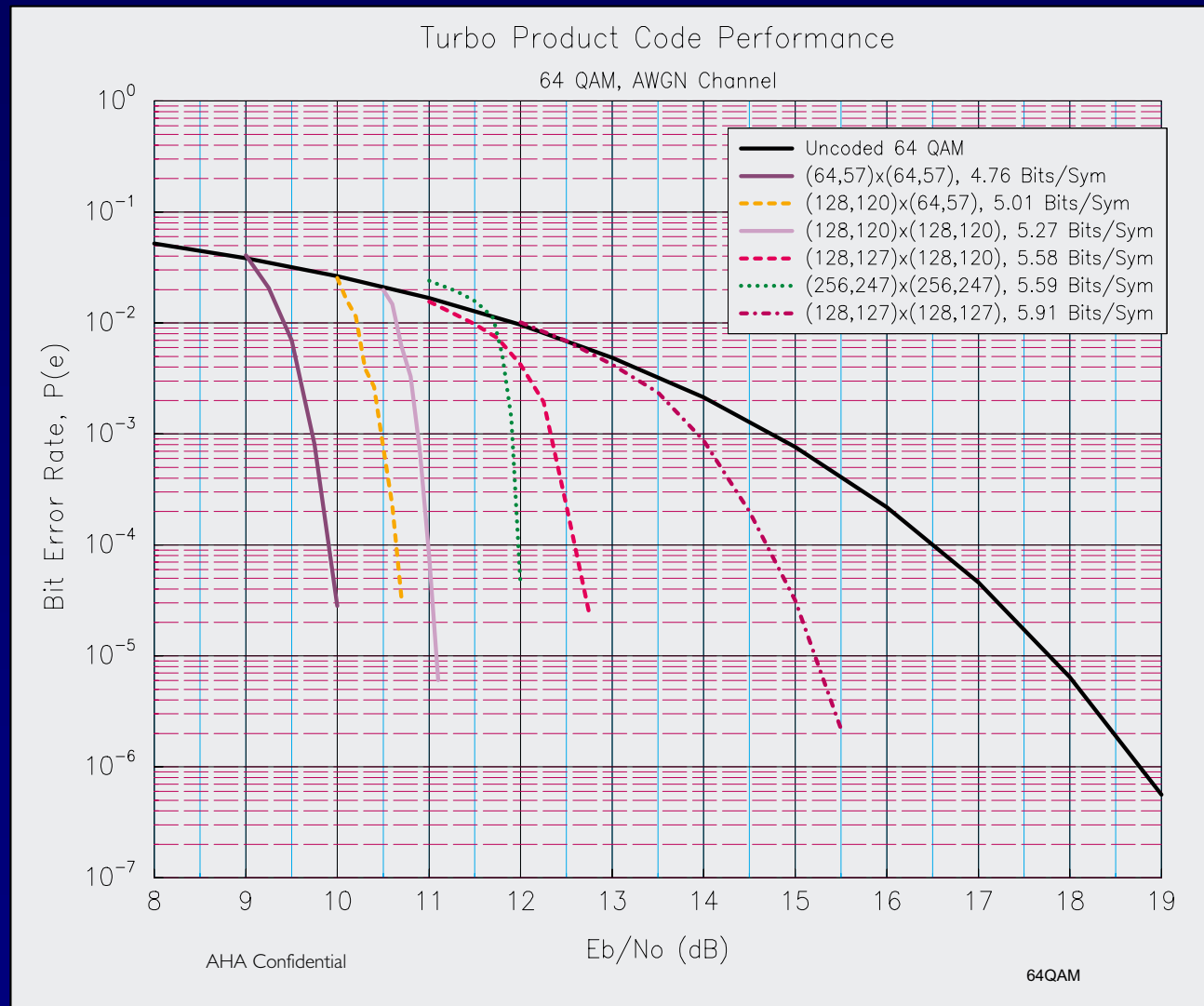
**High data rate, bandwidth efficient, data reliability.**

**Anything over 50 Mbits/sec.**

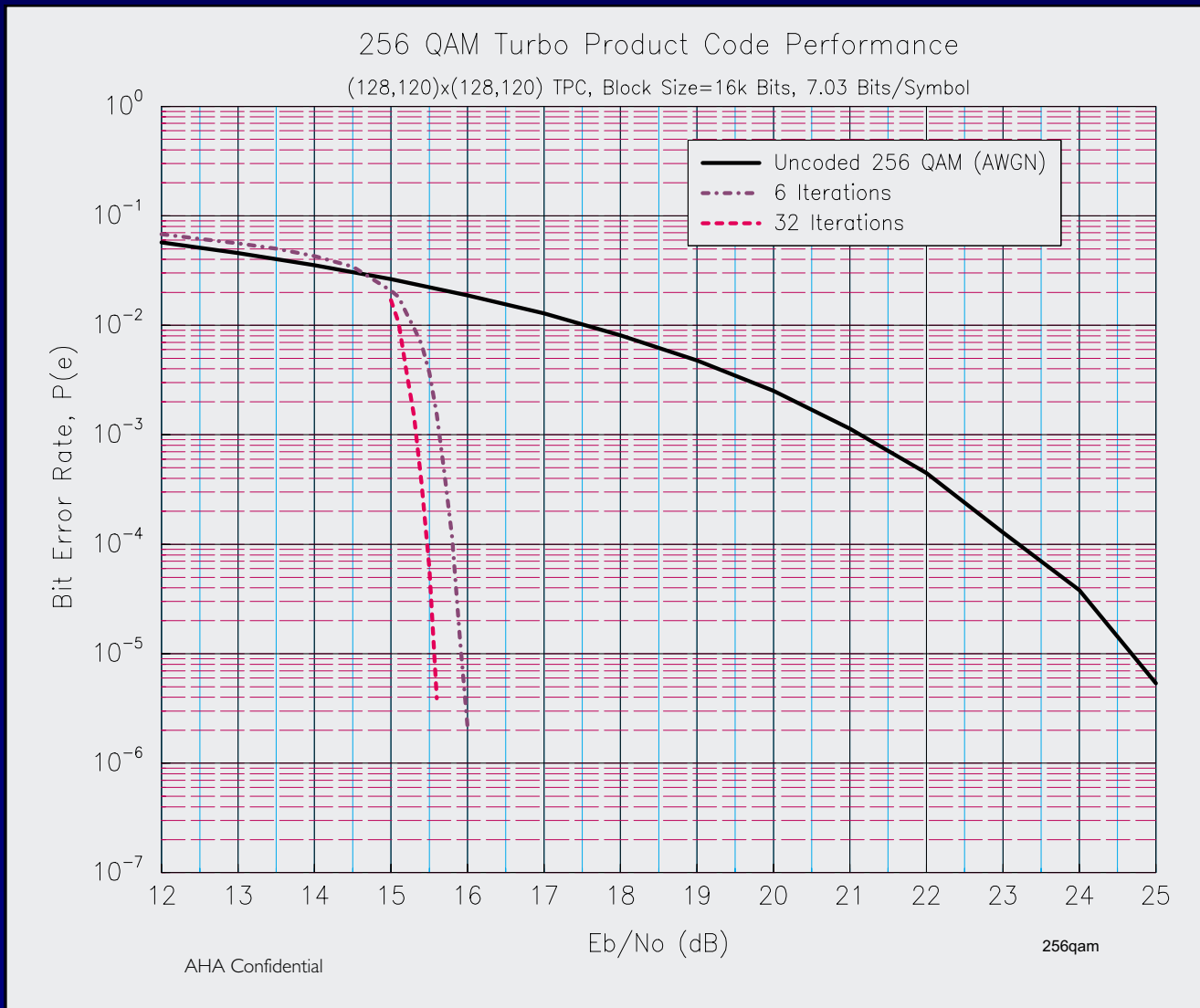
- ◆ **Satellite communications**
  - **Data and video**
  - **HDTV**
  - **Electronic news gathering**
- ◆ **Broadband point-to-point**
- ◆ **Point to multi-point broadband**



# TPC with 64 QAM



# TPC with 256 QAM



# Support Tools

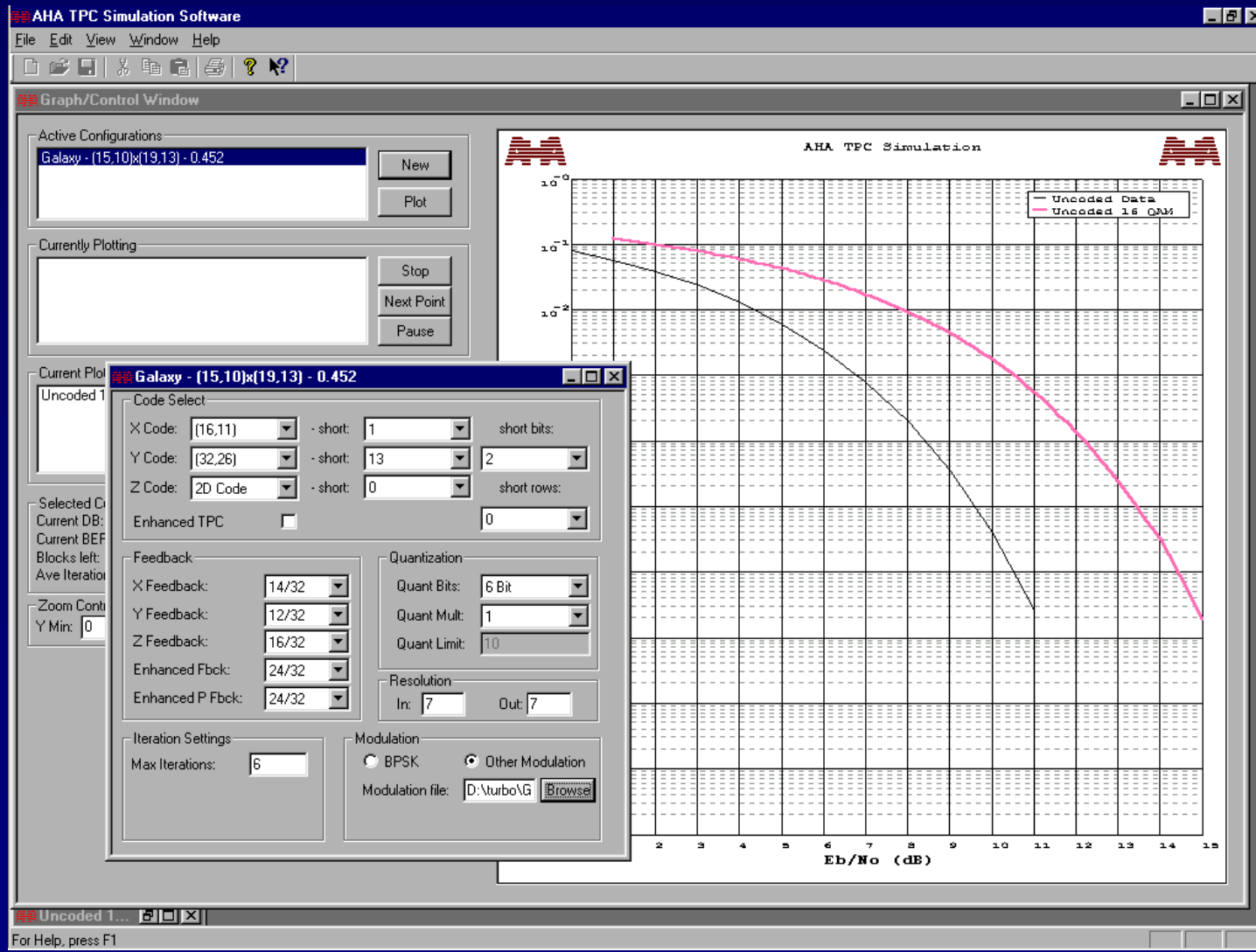
- ◆ **Evaluation SW**
- ◆ **Hardware demonstration board(s)**
  - **Generate BER curves**
  - **Connect to modem for system analysis**
  - **Connect to BER test equipment**
- ◆ **Application Notes**
  - **TPC primer**
  - **Designers guide**
  - **Code shortening**
  - **Burst Error performance**
- ◆ **Galaxy Simulation SW**

# Galaxy Simulation Software

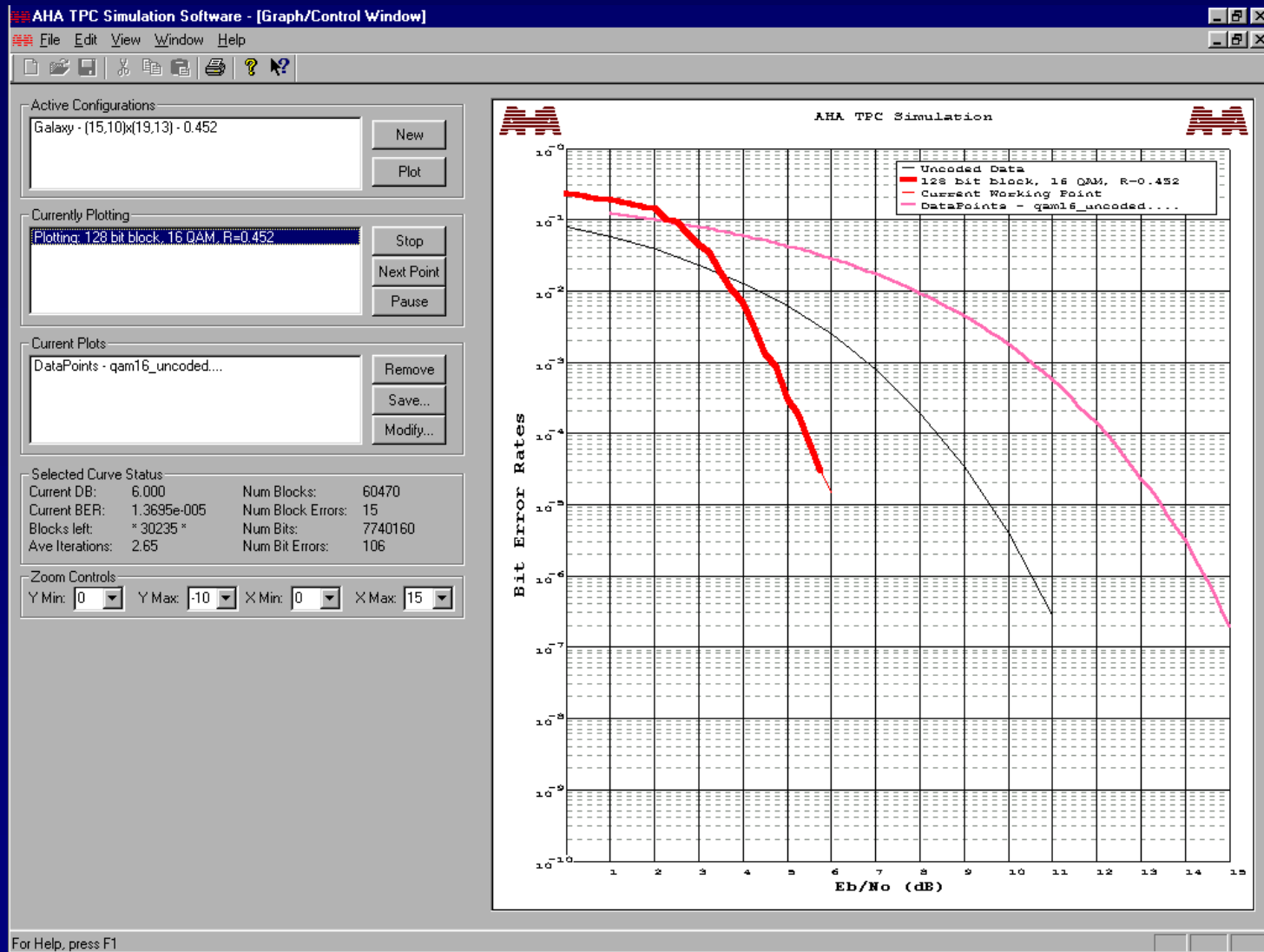
- ◆ **Windows based simulation software**
  - Support for all Galaxy TPC Core codes
  - Supports all AHA TPC ASIC codes
- ◆ **Evaluation version**
  - No cost to qualified customers
- ◆ **Simulation Toolkit**
  - Two versions available
    - C/C++ API
    - Matlab API
  - Enables integration of TPC encoding/decoding with user system and channel modeling



# Galaxy Simulation Software



# Galaxy Simulation Software



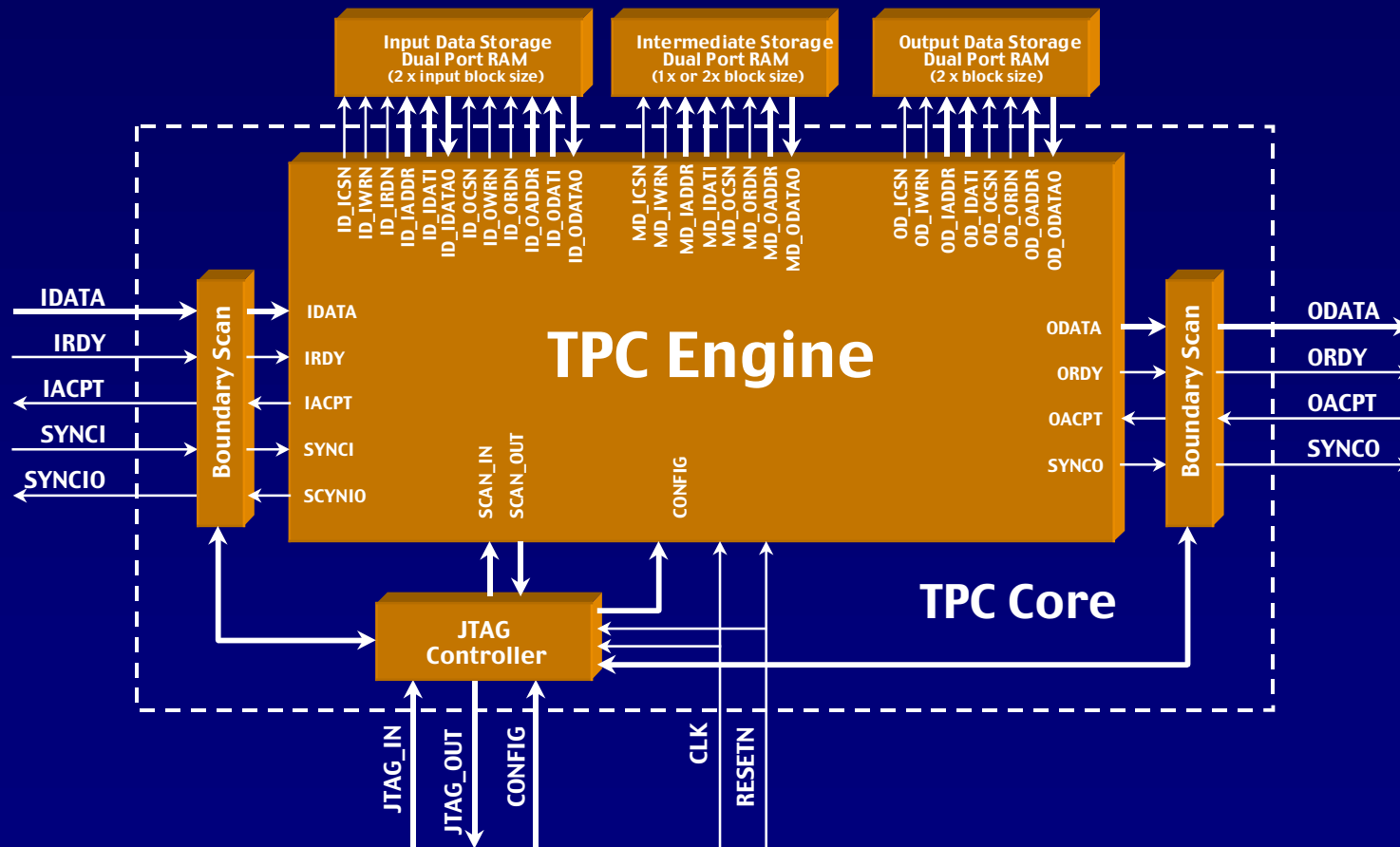
# TPC Galaxy Core Generator

- ◆ **Generates custom TPC cores**
- ◆ **VHDL or Verilog**
- ◆ **Data rates to 1+ Gbit/sec in 0.25 um CMOS**
- ◆ **Code rates from 0.20 to 0.98**
- ◆ **Block sizes from 64 bits to 128 Kbits**
- ◆ **Incorporates next generation eTPC technology**

# Turbo Product Code Cores

- ◆ Deliverables for Galaxy TPC Decoder cores include:
  - C/C++ behavioral model
  - VHDL or Verilog netlist
  - Verification vectors (>95% fault coverage)
  - VHDL or Verilog test bench
  - Bus functional model
- ◆ HW performance that matches SW simulations
- ◆ Supported with Galaxy Simulation Toolkit

# Turbo Product Code Core



# Conclusions

- ◆ **TPCs provide a superior, cost effective FEC solution for a wide range of applications**
- ◆ **Available now**
- ◆ **Standard products and licensable cores**
- ◆ **More information and support is available**
  - **Application Notes**
  - **Product Brief**

# Contact Information

- ◆ Website: [www.aha.com](http://www.aha.com)
- ◆ Phone: 509-334-1000
- ◆ email: [tpc@aha.com](mailto:tpc@aha.com)
- ◆ Register online
  - Enable “cookies” for document access



**Thank You!**

**Questions?**