Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)

Submission Title: [Implementation of a 480Mbps Viterbi Decoder for IEEE 802.15.3a]
Date Submitted: [15 September, 2003]

Source: [Harald Elders-Boll, Herbert Dawid] Company [Synopsys]
Address [Kaiserstrasse 100, 52134 Herzogenrath, Germany]
Voice:[+49 24079558 84], FAX: [+49 24079558 44], E-Mail:[Harald.Elders-Boll@synopsys.com, Herbert.Dawid@synopsys.com]

Re: []

Abstract: [Implementation of high speed Viterbi decoder, which meets the data throughput requirements for all proposals under consideration by 802.15.3a]

Purpose: [To establish the feasibility of implementing the proposals under consideration by 802.15.3a. In particular, discuss several important technical aspects of the high-speed implementation of Viterbi decoders and provide guidelines concerning the architectural trade-offs involved.]

Notice: This document has been prepared to assist the IEEE P802.15. 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.

Implementation of a 480 Mbps Viterbi Decoder for IEEE 802.15.3a

Harald Elders-Boll, Herbert Dawid

Synopsys

Solutions Group

Introduction

- **All PHY proposals currently discussed in 820.15.3a:**
  - Information bit rates up to 480Mbit/s
  - Convolutional codes:
    - Multiband OFDM: r=1/3, K=7, information rate $\leq$ 480Mbit/s
    - Xtreme/Parthus: r=1/2, K=7, information rate $\leq$ 200Mbit/s

$\Rightarrow$ **High-speed Viterbi decoder (VD) required**

- **Purpose of this presentation:**
  - Establish technical feasibility of proposals
  - Choice of high-speed Viterbi decoder architecture:
    - Discuss various technical alternatives
    - Provide guidelines for implementation (Do's & Don'ts)

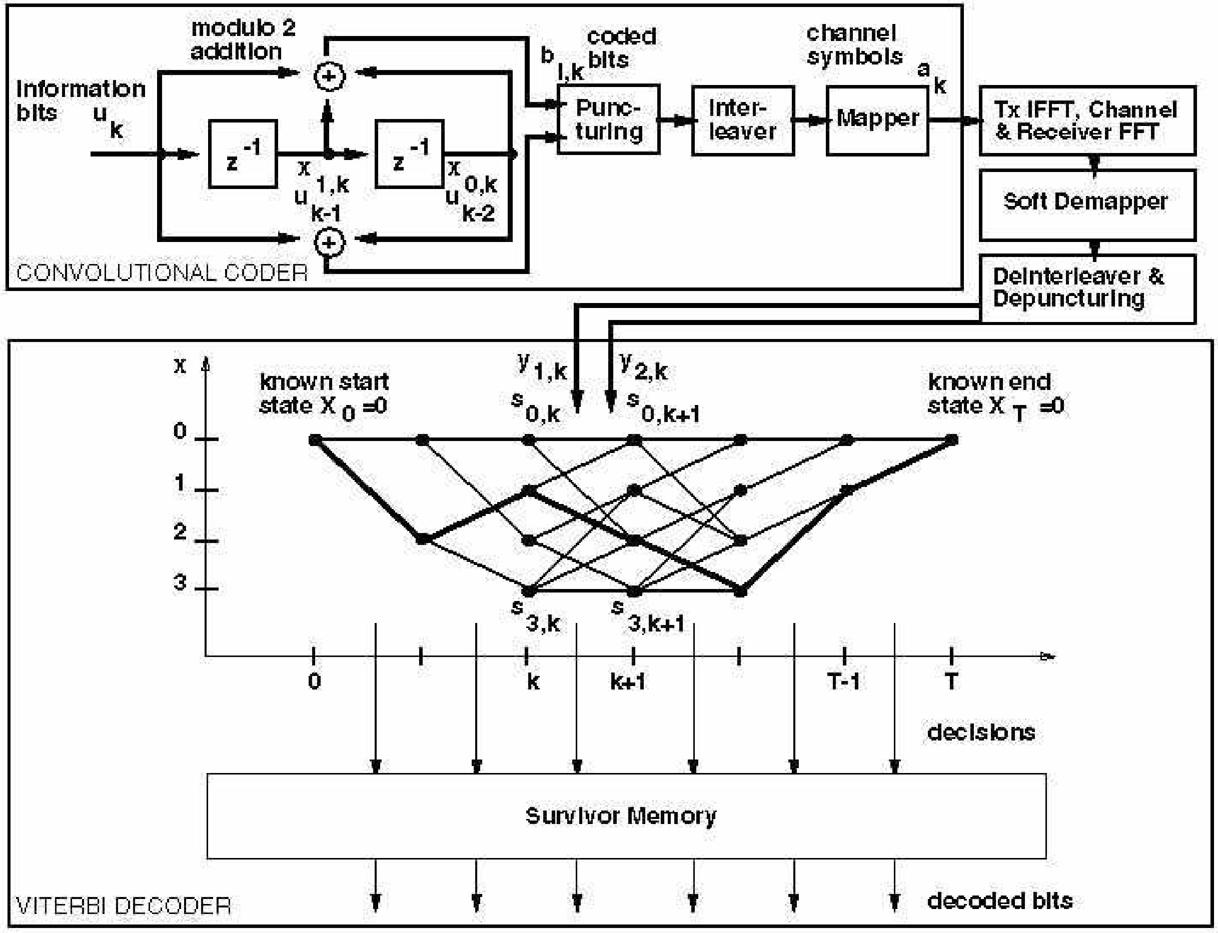
Viterbi Decoder Requirements

- **Assumption: Multiband OFDM proposal gets accepted**
- **Throughput:**
  - Information bit rate = 480Mbit/s = 150 bits per OFDM symbol
- **Latency: Minimum of**
  - Requirement in streaming packet mode w. Dly-ACK/no-ACK:
    Latency = MIFS+PLCPpreamble - OtherProcessingDelays
    = 2us + 4.6875us - OtherProcessingDelays
  - Requirement with Imm-Ack:
    Latency = SIFS - InterleaverDelay - OtherProcessingDelays
    = 10us - 0.9375us - OtherProcessingDelays

⇒ Throughput imposes most critical constraint concerning implementation

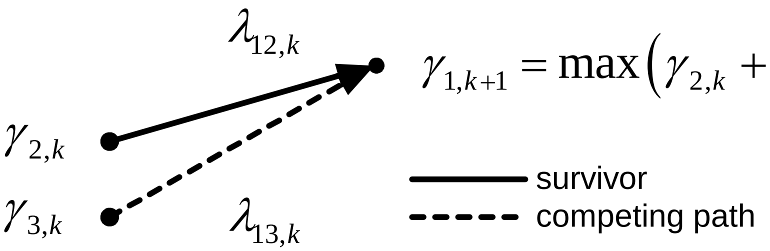
Viterbi Decoder Basics: Trellis Diagram

- **Example:**
  - r=1/2
  - K=3



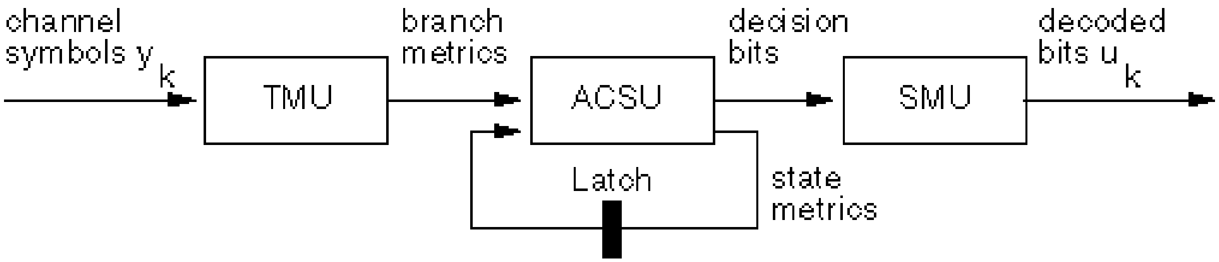
Viterbi Decoder Basics: ACS Unit

- **One step in the trellis diagram:**
  - Update 2^{K-1} state metrics γ_i = 2^{K-1} **A**dd-**C**ompare-**S**elect operations



$$\gamma_{1,k+1} = \max\left(\gamma_{2,k} + \lambda_{12,k}, \gamma_{3,k} + \lambda_{13,k}\right)$$

- **ACSU is the only recursive part in the VD:**



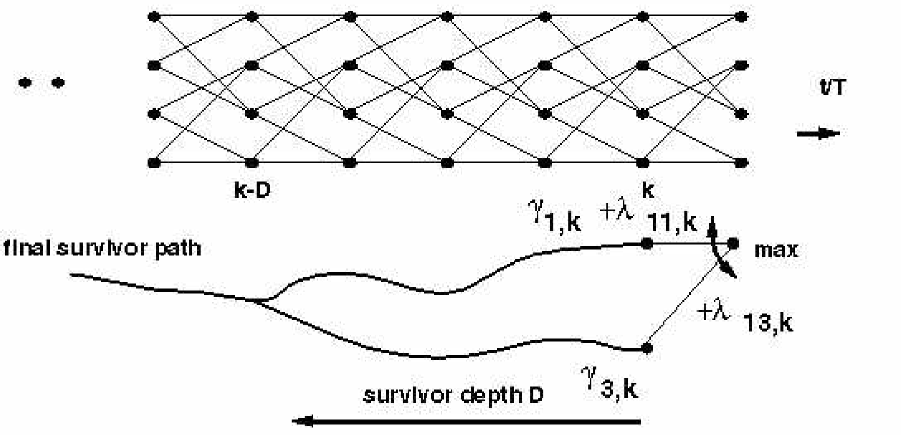
⇒ Clock rate of ACSU determines overall VD throughput

Obtaining the Information Bits (1)

- **Decoding of information bits at the end of code block:**
  - Large latency
  - Huge memory for SMU required: $2^{K-1} \cdot T$ bits (T=code block length)

  ⇒ Not useful for an efficient high-speed HW implementation

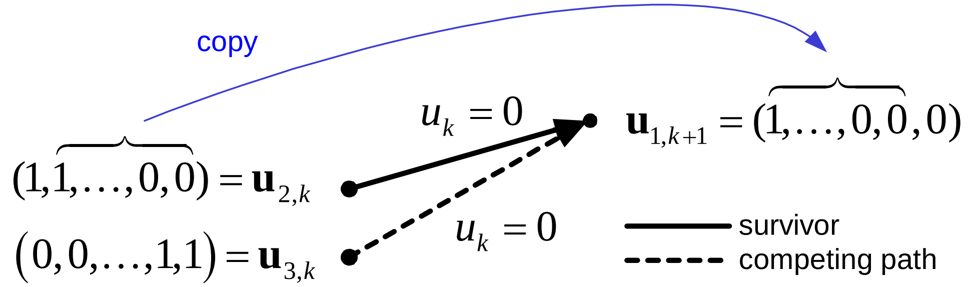
- **Truncation property of the Viterbi algorithm:**
  - Survivors of all states at time instant k merge at time instant k-D
  - D: Survivor depth
  - Rule of thumb:
    - D = 5K
  - **Puncturing: D has to be determined by simulations!**



Obtaining the Information Bits (2)

- **Register Exchange Algorithm (REA):**
  - Store: Sequence of information symbols of each state's survivor
  - Update per trellis step:
    - Copy sequence from state the chosen branch originated from

    ⇒Simultaneous access of **whole sequence** of info bits required

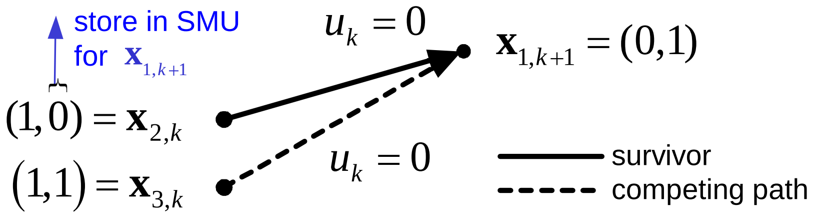

copy
$u_k = 0$
$\mathbf{u}_{1,k+1} = (\overbrace{1,\ldots,0,0},0)$
$(1,\overbrace{1,\ldots,0,0}) = \mathbf{u}_{2,k}$
$(0,0,\ldots,1,1) = \mathbf{u}_{3,k}$
$u_k = 0$
survivor
competing path

  - High-speed HW implementation:
    - High access bandwidth requires implementation by registers

⇒ High power consumption ⇒ Don't use REA!

Obtaining the Information Bits (3)

- **Traceback Algorithm (TBA):**
  - Update per trellis step:
    - Store the current decisions of the ACS units only


store in SMU
for $\mathbf{x}_{1,k+1}$
$u_k = 0$
$\mathbf{x}_{1,k+1} = (0,1)$
$(1,0) = \mathbf{x}_{2,k}$
$(1,1) = \mathbf{x}_{3,k}$
$u_k = 0$
survivor
competing path

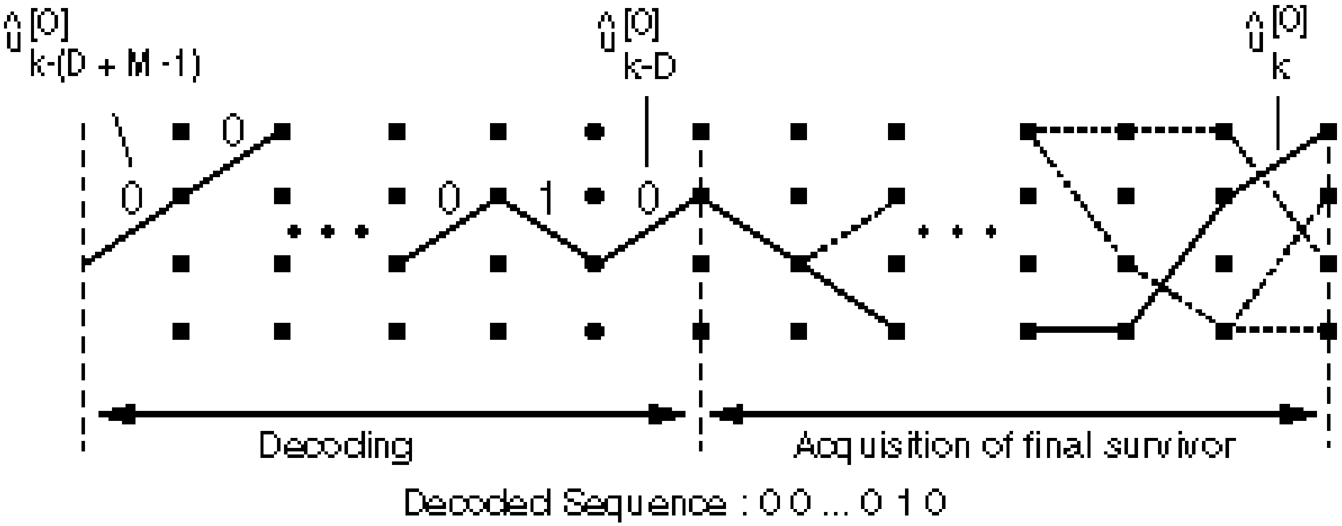
  - Output of information bits:
    - Trace back through trellis and extract information symbols
  - High-speed HW implementation:
    - Much lower access bandwidth than REA facilitates implementation by RAM

$\Rightarrow$ Low power consumption $\Rightarrow$ Do traceback!

Obtaining the Information Bits (4)

- **Block Traceback:**
  - Acquire final survivor in D steps
  - Decode M information symbols

⇒ Storage requirements for SMU: $2^{K-1} \cdot (D+M)$ symbols


$\hat{u}^{[0]}_{k-(D+M-1)}$
$\hat{u}^{[0]}_{k-D}$
$\hat{u}^{[0]}_{k}$
0
0
0
1
0
0
Decoding
Acquisition of final survivor
Decoded Sequence : 0 0 ... 0 1 0

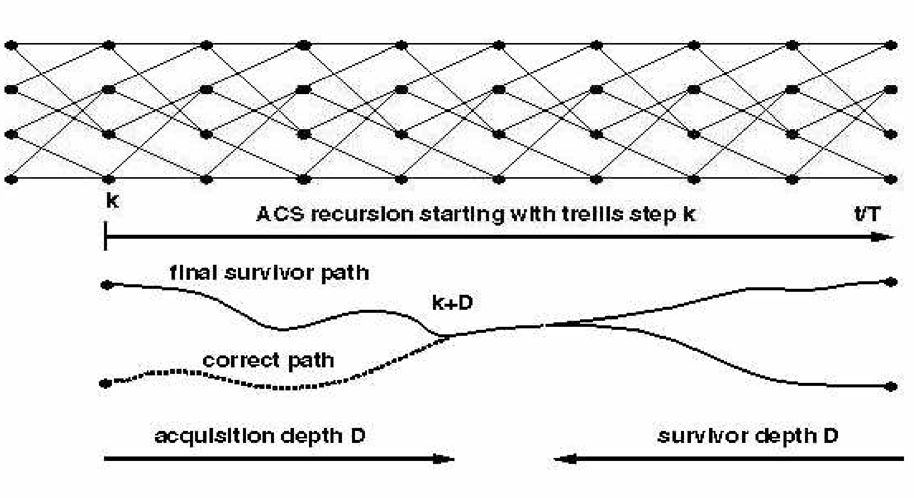
M symbols D symbols

High Speed Viterbi Decoder Architectures

- **Maximum clock rate of ACSU HW implementation:**
  - Depends on:
    - Complexity of ACSU (quantization of branch & state metrics!)
    - Target technology (semiconductor process)
  - Limits:
    - Throughput
- **VD architectures with throughput > ACSU clock rate:**
  - Parallel ACS units with acquisition (and truncation)
  - M-step (Radix-2^M) ACSU
  - Parallel forward & backward ACS units (Minimized method)

Parallel ACS Units with Acquisition

- **Acquisition property of the Viterbi algorithm:**
  - Starting at any state at time instant k, the final survivor will merge with the correct path at time instant k+D
  - D: Acquisition depth
    = survivor depth
  - Rule of thumb:
    - D = 5K
  - Puncturing: D has to be determined by simulations!



- **Acquisition: Decisions cannot be used for traceback**

⇒ **Throughput increases less than linear with #ACSU**

Semiring Notation of ACS Operation

- **Define maximum selection as algebraic addition $\oplus$:**

$$a \oplus b := \max(a,b), \text{ with neutral element: } \underline{0} := -\infty, \;\; a \oplus \underline{0} = a$$

- **Define addition as algebraic multiplication $\otimes$:**

$$a \otimes b := a + b, \text{ with neutral element: } \underline{1} := 0, \;\; a \otimes \underline{1} = a$$

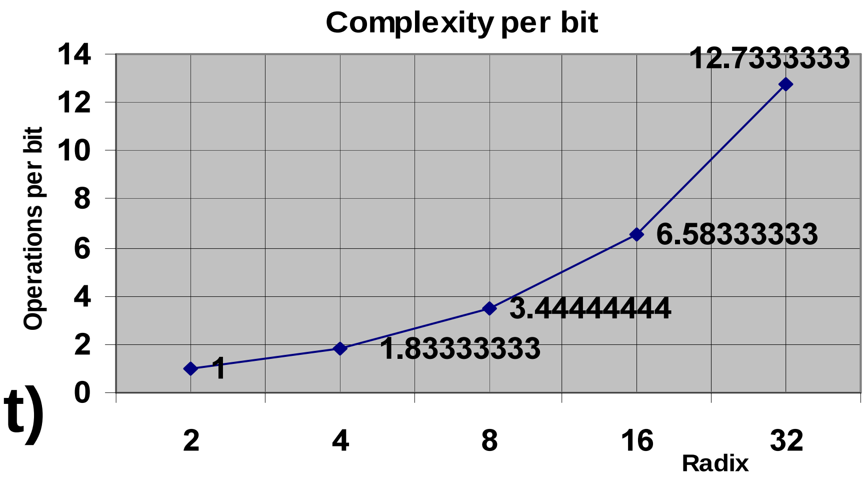
- **ACS update can be written as matrix-vector product:**

$$\Gamma_{k+1} = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix}_{k+1} = \begin{bmatrix} \max\left(\lambda_{00} + \gamma_0, \lambda_{01} + \gamma_1\right) \\ \max\left(\lambda_{12} + \gamma_2, \lambda_{13} + \gamma_3\right) \\ \max\left(\lambda_{20} + \gamma_0, \lambda_{21} + \gamma_1\right) \\ \max\left(\lambda_{32} + \gamma_2, \lambda_{33} + \gamma_3\right) \end{bmatrix}_k = \begin{bmatrix} \lambda_{00} & \lambda_{01} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \lambda_{12} & \lambda_{13} \\ \lambda_{20} & \lambda_{21} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \lambda_{32} & \lambda_{33} \end{bmatrix}_k \otimes \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix}_k$$

$$\Gamma_{k+1} = \Lambda_k \otimes \Gamma_k$$

M-Step (Radix-2^M) ACSU

- **M steps of 1-step ACSU:** $\Gamma_{k+M} = \Lambda_{k+M-1} \otimes (\ldots \otimes (\Lambda_k \otimes \Gamma_k))$
  - M matrix-vector products
- **M-step ACSU:** $\Gamma_{k+M} = \left(\Lambda_{k+M-1} \otimes \ldots \otimes \Lambda_k\right) \otimes \Gamma_k$
  - M-1 matrix-matrix products
    + 1 matrix-vector product
- **Complexity per bit increases exponentially! (Gate count per bit, power consumption per bit)**



⇒ Don’t use M-step ACSU with M>2 (Radix>4)!

Parallel Forward & Backward ACS

- **Calculation of $\Gamma_{\mathbf{k}}$ by an ACS acquisition recursion:**
  - Initialize $\Gamma_{\text{k-D}}$ by all-zero vector: $\Gamma_{k-D} = (0,\ldots,0)^T = (\underline{1},\ldots,\underline{1})^T$
  - D steps of forward ACS recursion:

$$\Gamma_k = \Lambda_{k-1} \otimes \ldots \otimes \left(\Lambda_{k-D} \otimes (\underline{1},\ldots,\underline{1})^T\right)$$

- **Best-state decoding & traceback (truncation):**
  - Chose maximum of $\Gamma_{\text{k+D}}$: $\max_i \gamma_{i,k+D} = (\underline{1},\ldots,\underline{1}) \otimes \Gamma_{k+D}$
  - Traceback over D steps:

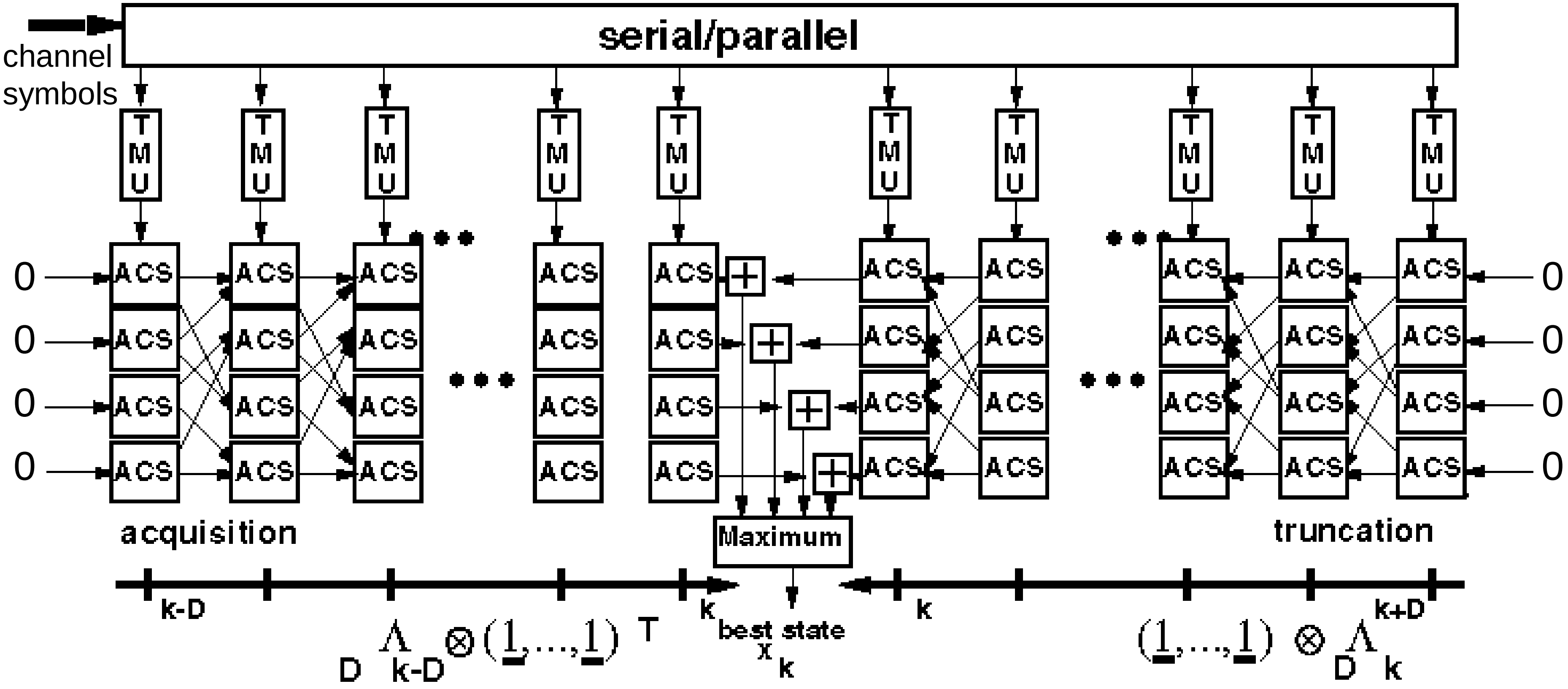
$$\max_i \gamma_{i,k+D} = (\underline{1},\ldots,\underline{1}) \otimes \Gamma_{k+D} = (\underline{1},\ldots,\underline{1}) \otimes \left(\Lambda_{k+D-1} \otimes \ldots \otimes \Lambda_k\right) \otimes \Gamma_k$$

$$= \left(\Lambda_k^T \otimes \ldots \otimes \left(\Lambda_{k+D-1}^T \otimes (\underline{1},\ldots,\underline{1})^T\right)\right)^T \otimes \Gamma_k$$

$\Rightarrow$ **Traceback can be implemented as backward ACS!**

Parallel Forward & Backward ACS

- **Resulting structure:**


channel symbols
serial/parallel
TMU
ACS
0
0
0
0
acquisition
Maximum
truncation
k-D
k
best state
x_k
k
k+D
${}_D\Lambda_{k-D} \otimes (\underline{1},\ldots,\underline{1})^T$
$(\underline{1},\ldots,\underline{1}) \otimes {}_D\Lambda_k$

Minimized Method

Remaining problem to be solved:

- How to partition the Rx data into blocks
- Schedule for processing of thereof

Minimized Method:

- Process non-overlapping blocks of size 2D

⇒Overhead minimized

⇒ Do minimized method ACS!

High-speed VD Design Issues

- **Lessons learned:**

	Do’s	Don’ts
ACSU	Minimized Method Parallel ACSUs	M-Step ACSU Radix-2^M ACSU
SMU	Traceback	Register Exchange

- **Further critical design issues/parameters:**
  - Quantization of ACS unit (input & internal wordlengths):
    - OFDM ⇒ performance simulations with **fading** required
  - Acquisition/truncation depth D with puncturing:
    - Has to be determined by system simulations
  - Overall decoder architecture and scheduling

Synopsys UWB VD Features

- Synthesizable RTL code
- Code parameters according to MBOA proposal
- Trial synthesis results:
  - TSMC 0.13 um technology (slow, TSMC13_conservative)
  - Clock speed up to 300 MHz
- Throughput = 2 times clock speed ≥ 480 Mbit/s
- Latency < 4 OFDM symbols
- Very low complexity: 90K gates + 20Kbit RAM
- Ultra low power consumption
- Implementation loss ≤ 0.1dB

Conclusion

- Designing a Viterbi decoder that meets the requirements imposed by the MBOA proposal is a challenge!
- This challenge can be handled even for 480Mbit/s by careful system & architectural design
- Development of such a Viterbi decoder and a compatible FFT/IFFT is in an advanced state
- Planned availability end of 2003!