Pipelining Tomlinson-Harashima Precoders

Keshab K. Parhi and Yongru Gu

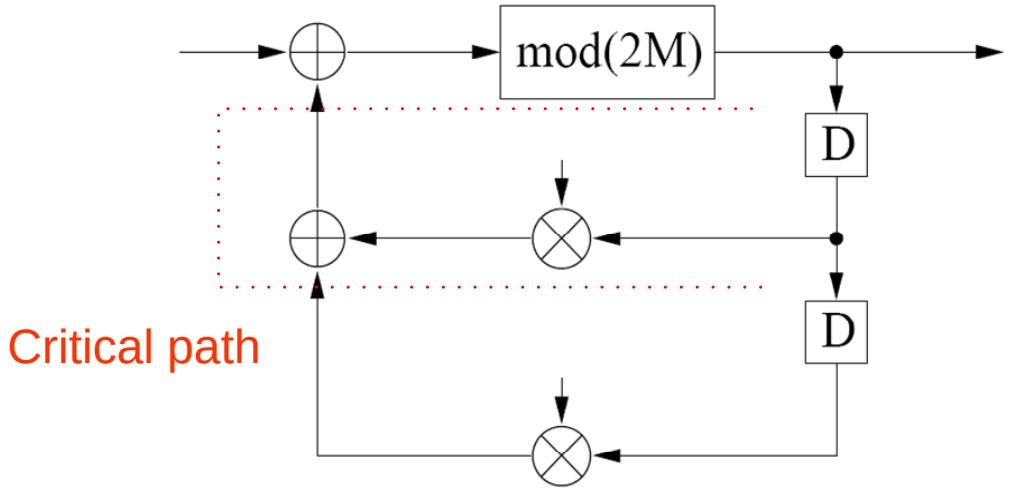
Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis, MN 55455
Email: {parhi,yrg}@ece.umn.edu

IEEE 802.3an Task Force Meeting
September 2004
Ottawa, Canada

10GBASE-T Modulation Scheme

- Tomlinson-Harashima precoding + LDPC + PAM-M modulation
- Clock Speed: 800MHz – 1GHz

A 2nd Order FIR TH Precoder and its Critical Path

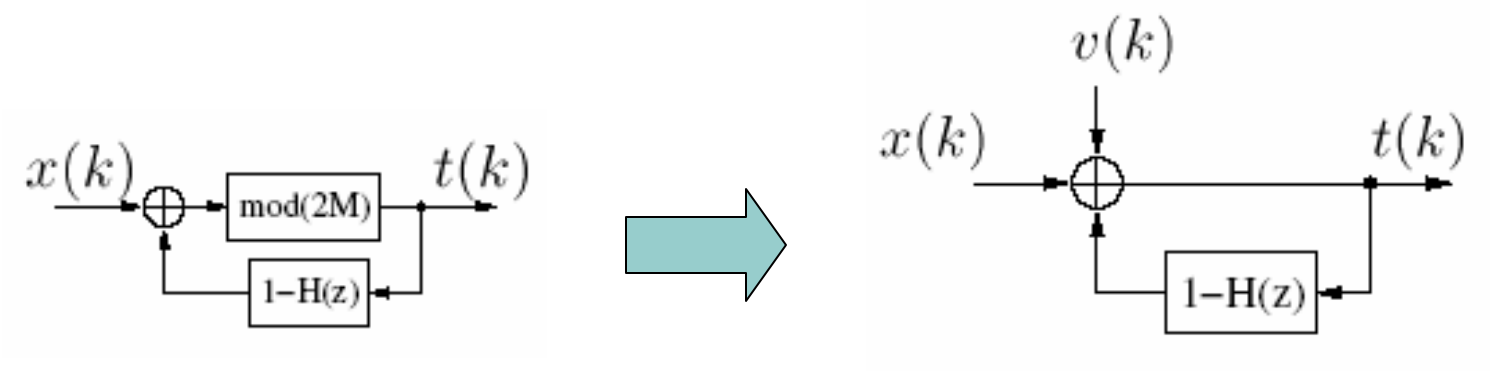


- The speed of the precoder is limited by $T_{critical}$=2Ta+Tm+Tmod
- $T_{critical}$ cannot be reduced by using retiming as the iteration bound of the precoder, T_{∞}, is also equal to 2Ta+Tm+Tmod [1].
- High-Speed Precoder Implementation is an open problem for 10GBASE-T (powell_1_0304.pdf, zimmerman_1_0504.pdf)

Difficulty in Pipelining TH Precoders

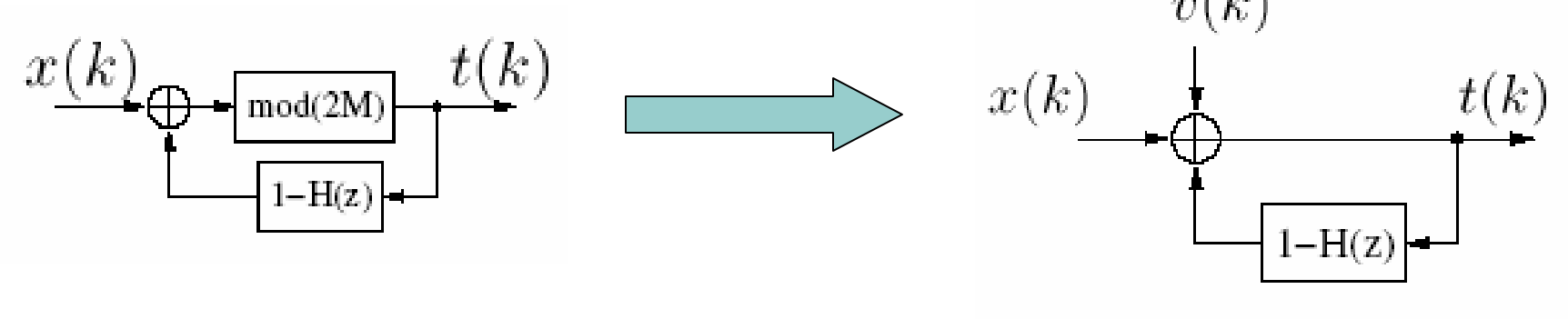
- The architecture of a TH precoder is similar to that of a DFE. However, for fixed-point implementation, the output level is finite but very large (=2^N where N is the wordlength). So classical pre-computation techniques, which were successfully used to pipeline DFEs [2], cannot be applied to pipeline TH precoders.
- Classical look-ahead techniques, used to pipeline IIR filters [1], cannot be applied to pipeline TH precoders since TH precoders contain nonlinear elements in their feedback loops.

Pipelining A TH Precoder based on its Equivalent Form

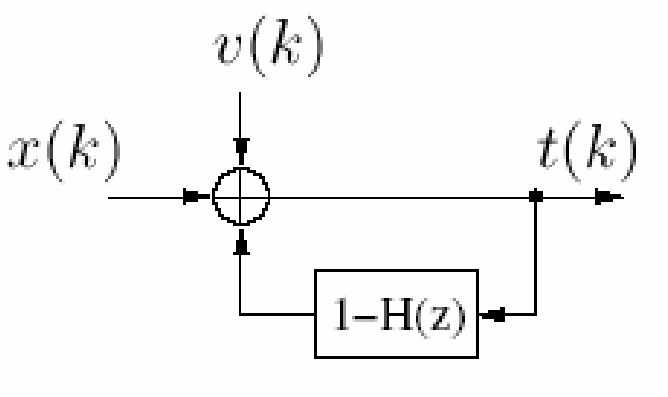


- A TH precoder can be viewed as an IIR filter with an input equal to the sum of the original input to the TH precoder and a compensation signal *v*(*k*);
- *v*(*k*) is a multiple of 2M, and is added to the input to the IIR filter 1/H(z) such that the output of the IIR filter is in the range of [-M, M);
- |v(k)| <=(1+Σ|hi|)M so v(k) only has finite levels.
- Classical techniques, such as the clustered and the scattered look-ahead techniques, can be used to pipeline the IIR filter [1].

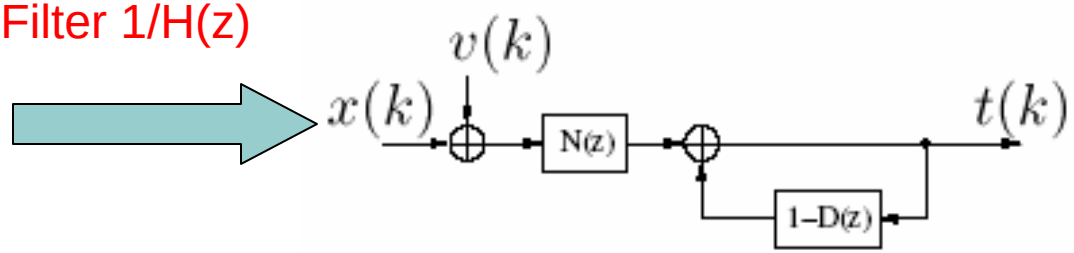
Pipelining A TH Precoder: Step 1



Pipelining A TH Precoder: Step 2

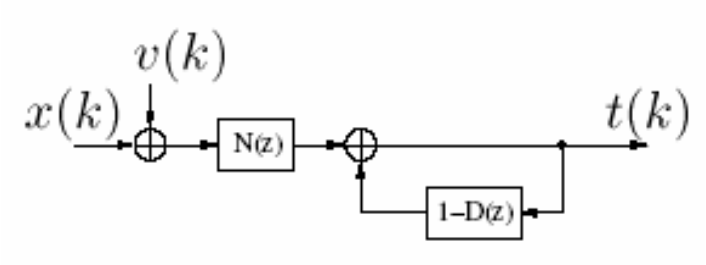


Pipelining the IIR Filter 1/H(z)

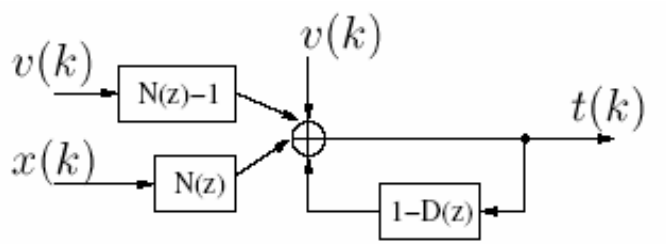


- Where N(z) is a pipelining polynomial of the form of $1+\Sigma n_i z^{-i}$ (See reference [1]).
- 1/D(z) is a pipelined IIR filter.

Pipelining A TH Precoder: Step 3

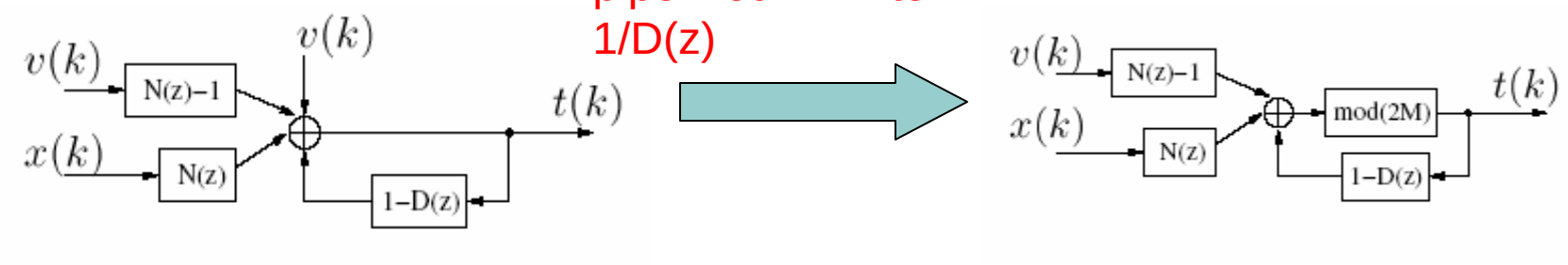


Reformulation

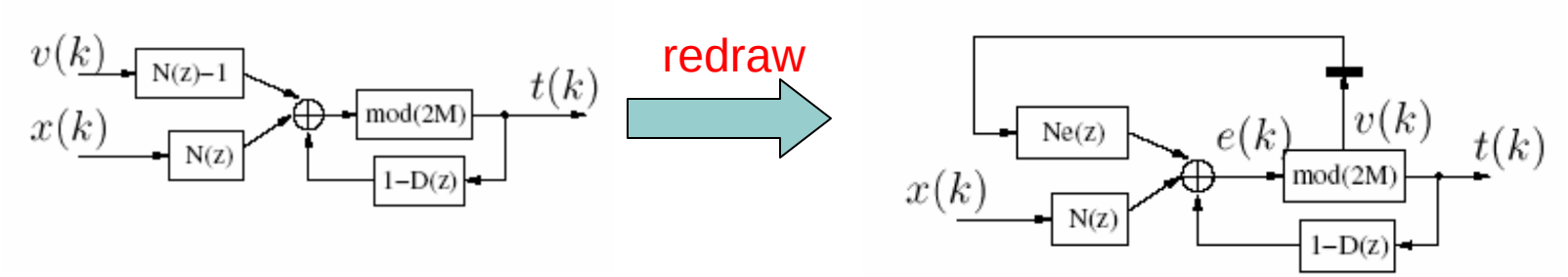


Pipelining A TH Precoder: Step 4

Eliminating v(k) as
an explicit input to the
pipelined IIR filter
1/D(z)



Pipelining A TH Precoder: Step 5

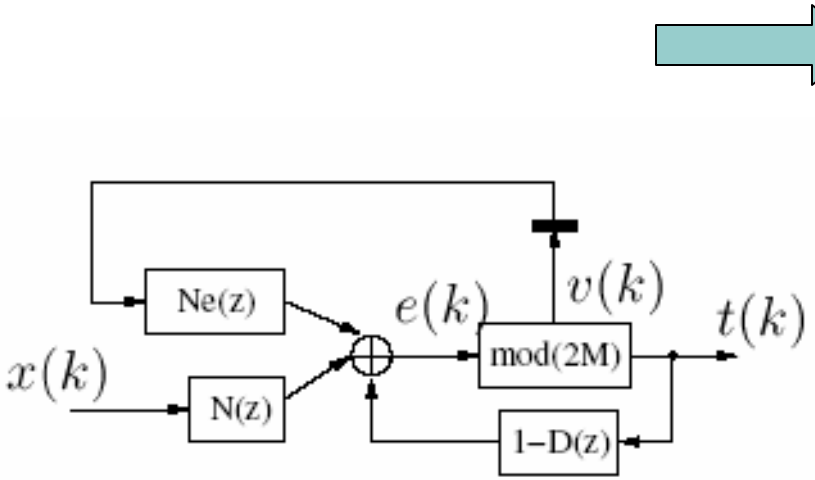


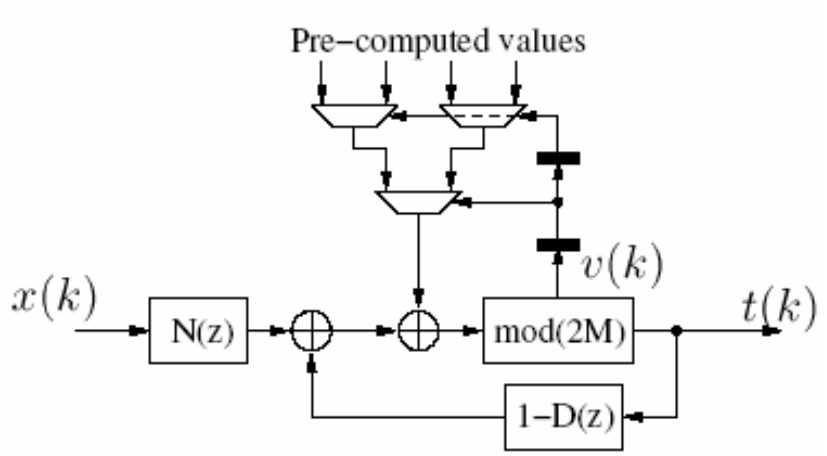
- $N(z) = 1+\Sigma n_i z^{-i}$

- $N_e(z) = \Sigma n_i z^{-i+1}$
- Two loops: One pipelined loop and one non-pipelined but with finite levels. We can use the pre-computation technique to the non-pipelined one.

Pipelining A TH Precoder: Step 6

precomputation





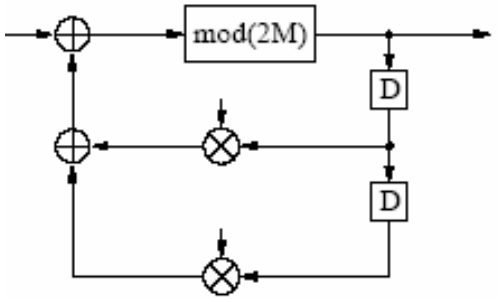
- With enough pipelining level:

$$T_\infty = T_a + T_{\text{mod}} + T_{mux}$$

- If M is a power of 2:

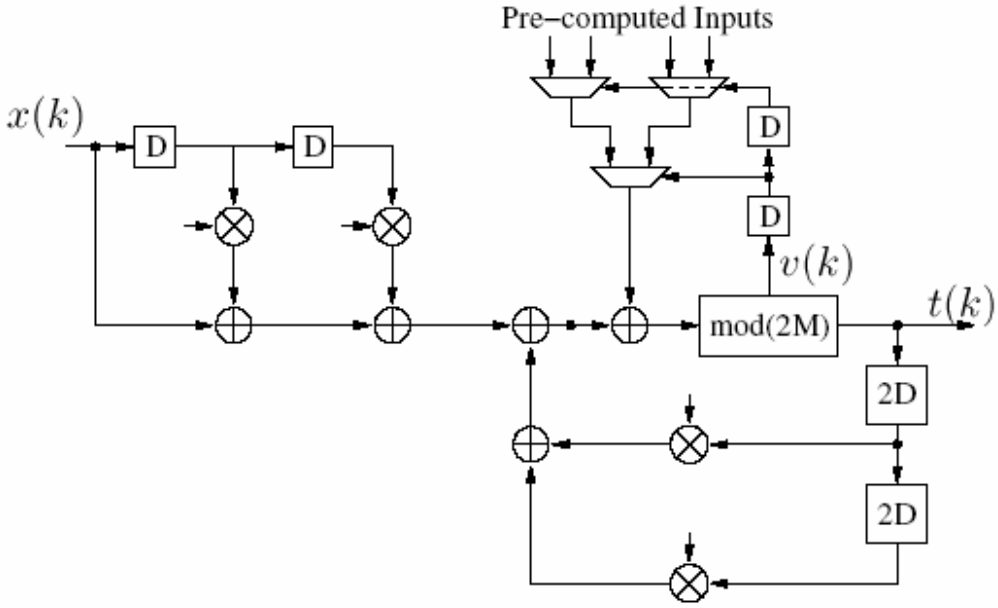
$$T_\infty = T_a + T_{mux}$$

Example: Pipelining A 2nd Order FIR TH Precoder



Original precoder

$$T_\infty = 2T_a + T_m + T_{\text{mod}}$$



Pipelined Design

$$T_\infty = \frac{3T_a + T_m + T_{\text{mod}}}{2}$$

Pros and Cons

- Pros:
  - Low Latency
  - It can be easily generalized to pipeline IIR TH precoders.
- Cons:
  - High complexity, especially for high-level pipelining (But for 10GBASE-T, we expect that 2 to 3-level pipelining is enough. In addition, it may be enough if we just apply precomputation to first several taps of the FIR filter Ne(z)).

References

[1] K. K. Parhi, *VLSI Digital Signal Processing System Design and Implementation*, John Wiley & Son, Inc., New York, 1999.

[2] K. K. Parhi, “Pipelining in algorithms with quantizer loops,” *IEEE Trans. on Circuits and Systems*, vol. 37, no. 7, pp. 745-754, July 1991.