Hardware implementation of Sequence Estimators with Block Termination in the presence of ISI

Presentation to IEEE 802.3bj

Arash Farhood – Cortina Systems

Joel Goergen – Cisco

Elizabeth Kochuparambil - Cisco

Supporters

- Dariush Dabiri, APM
- Matt Brown, APM
- Adee Ran, Intel Corp
- Richard Mellitz, Intel Corp
- Lup Ng, Cortina Systems

Agenda

- Objective of this presentation
- ISI cancellation is required for 100G Base KR4/KP4 systems
- High-speed implementation of sequence estimators with feed-back
- Look-ahead implementation of DFE and VA
- Block based implementation of DFE and VA
- Block termination
- Simulation results
- Impact of 45/46 block termination on 100G KP4 systems
- Summary

Objective

- 100G Base KP4 transmitter periodically sends PAM2 termination symbols to terminate every block of PAM4 symbols. This is called partial termination.
- Partial termination can be very effectively used to implement complex sequence estimators such as Viterbi Algorithm (VA) on the received side
- As we will show in this presentation, with partial termination, the VA complexity is comparable to a Decision-Feedback-Equalizer (DFE) with or without partial termination
- This is a response to the unresolved comment by Charles Moore, Avago Technologies, comment #80 against Cl94 SC 94.2.2.4 of Draft 1.0. The comments asks for the removal of partial termination scheme

ISI cancellation is required for proper operation of 100G Base KR4/KP4

- Both 100G KR4 and KP4 channels are band limited and introduce substantial ISI to the received signal
- A combination of filtering at the transmitter and filtering/sequence-estimation at the receiver can be used to reliably recover the signal.
- Common known techniques are the use of Decision-Feedback-Equalizer (DFE) or Maximum-Likelihood-Sequence-Estimation implement through a Viterbi Algorithm (VA) to recover the signal
- One can consider the DFE and VA the two sides of spectrum. It is possible to combine these algorithms and come up with many different architectures trading of complexity and estimation gain.

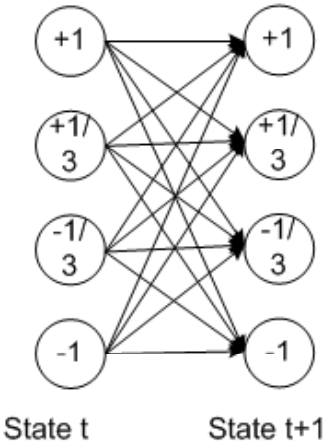
In this presentation we are going to study the impact of block termination to hardware implementation of feedback sequence estimators in general.

High-speed implementation of sequence estimators with feed-back

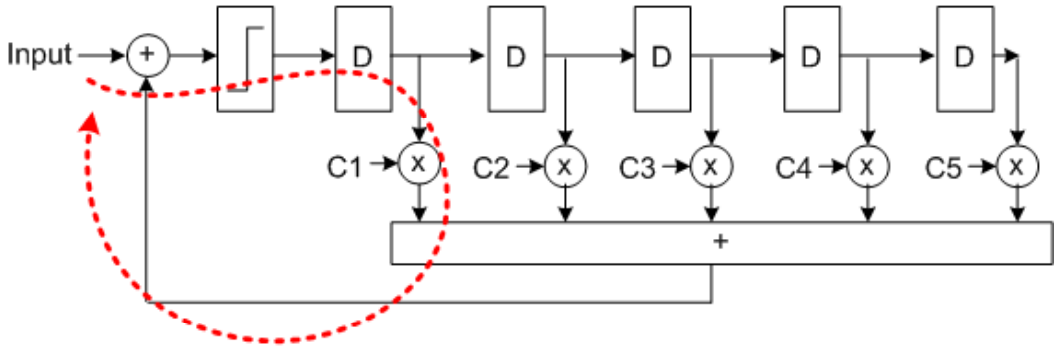
- Both VA and DFE are feedback systems. The current state is computed as function of previous states. In the case of DFE, the state of current symbol is recovered simply by looking at the “n” previously recovered symbols.

- For 100G Base KP4, the symbol rate is about 13Gb/s per lane. This requires a 13GHz feedback implementation (or some way of estimating it). This can be a challenge.

With VA, the state values at t+1 are function of state values at t



With DFE, the red line shows the high-speed feedback through the first tap



High-speed implementation of sequence estimators with feed-back

In digital or Analog implementation of VA or DFE (or any hybrid architecture), there are two ways to deal with the high speed feedback:

- **Look ahead**: These techniques pre-assume a set *S* of potential starting states, evaluate all outcomes for *S* and then select the right member state from *S*.

- **Block based**: These techniques partition the input data into blocks. The state values for each block is estimated or fixed and is independent of previous blocks.

- It is also possible to the above two methods and produce some higher radix block based algorithm

Look ahead implementation of DFE or VA

- For look ahead implementation of DFE or VA, a set *S* of starting states is pre-assumed. This set is most often taken as the set of all possible states. For example, a look ahead implementation of PAM2 DFE for 100G Base KR4 will assume the state of the first flip-flop as both 0 and 1, then compute the DFE value for both possibilities, and then select the right one. This is a well known implementation of DFE called half-rate architecture (since the computation is done at half of the baud rate)

- Note that with Look ahead technique, we are trying to simply transform the algorithm into a more hardware friendly implementation. The nature of the function is not changed.

Look ahead implementation of DFE and VA

- The complexity of a look ahead architecture is proportional to the amount of look ahead performed.
- The amount of look ahead needed is a function of
  - Number of times the design is unrolled and the size of the state
  - Size of the symbol set
- For a digital implementation, lets assume the size of symbol set is *M* (ie 2 for PAM2 and 4 for PAM4), amount of unrolling is *u* times and number of state registers are *v* bits. Then just due to the unrolling the hardware area is expanded by

Equation 1: area is proportional to M^u

Look ahead implementation of DFE and VA

- Equation 1 is not a correct equation if the amount of unrolling exceeds the number of state bits in the system (ie $u>v$).

- This can generally happen in a digital implementation of DFE or VA. For example a digital implementation of a 5-tap DFE for 100G Base KR4 at around 400MHz would require unrolling of factor 32x (the Baud rate is around 13Gs/s. 13GHz/32 is approximately 400MHz). So u=32 and v=5

- When u>v, all the possibilities are exhausted after the first v unrolls. So the hardware area is only expanded by

Equation 2: Area is proportional to $M^{min(u,v)}$

Look ahead implementation of DFE and VA

- The complexity can further be reduced by exploring properties of the states and deduct the value of some of the states from the other states. This is a linear reduction of size *Kp*

Equation 3: Area is proportional to $M^{min(u,v)}/Kp$

- The complexity can again be reduced by only exploring a sub-set of the entire state set *S*. The idea is the unexplored states are unlikely states and can be ignored. As far as we understand the current literature, this introduces another linear reduction of the order *Ks*. Note that since all states are not explored, this is technically not equivalent to the original high-speed algorithm.

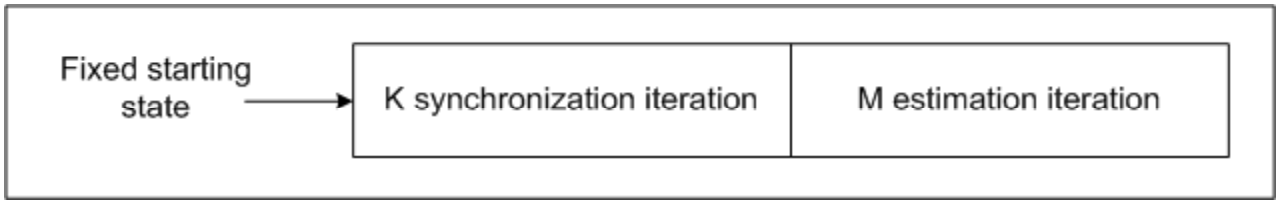
Equation 4: Complexity is proportional to $M^{min(u,v)}/KpKs$

Block based implementation of DFE and VA

- With block based implementation, the idea is, due to **finite memory** of the sequence estimator, starting from a state and after some number of iterations, the final state is the same regardless of the starting state.

- **The block based algorithms most often result in substantial area reduction since hardware area has now linear complexity rather than exponential complexity in the look ahead case**

- For a DFE, the starting state can be any state. For VA, the starting state can be a fair state (like 0 weight for all potential starting states)

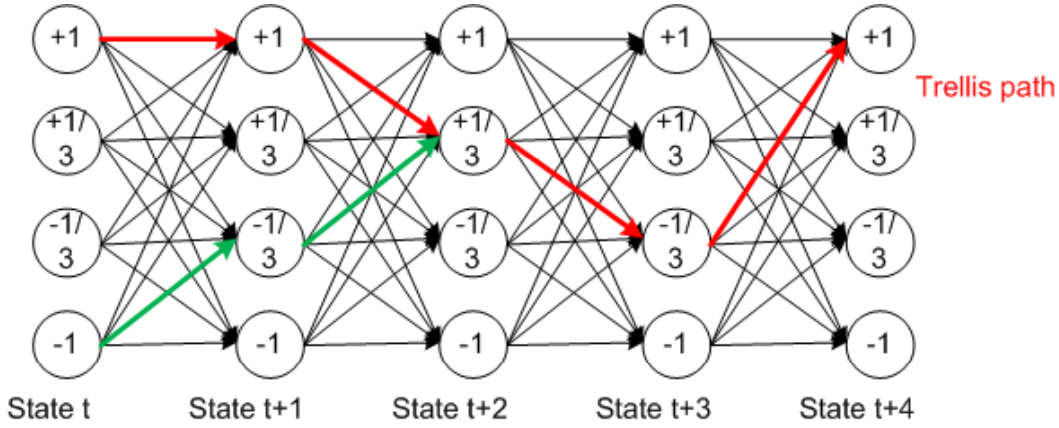


- The extra iterations required to put the estimator in a good state are called **synchronization iterations**. Let's assume *K* is the synchronization length.

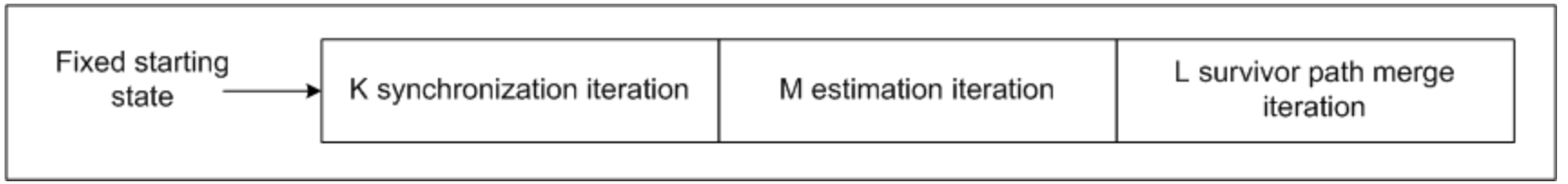
Block based implementation of DFE and VA

- For the VA, the objective is to find the most likely sequence as the estimation of the transmitted sequence. The search is done by finding the shortest path leading to a state. This path is called *survivor path.*

- If the survivor paths from all possible states at time *n* are traced back, then with high probability all the path merge to a single path at time *n-L. L* is the survivor path length



Block based implementation of DFE and VA



- In a simple block based implementation, to estimate *M* symbols, one need extra *K* synchronization iteration at the beginning and *L* synchronization iteration at the end.
- The utilization of this hardware is

Equation 4: Utilization=M/(K+M+L)

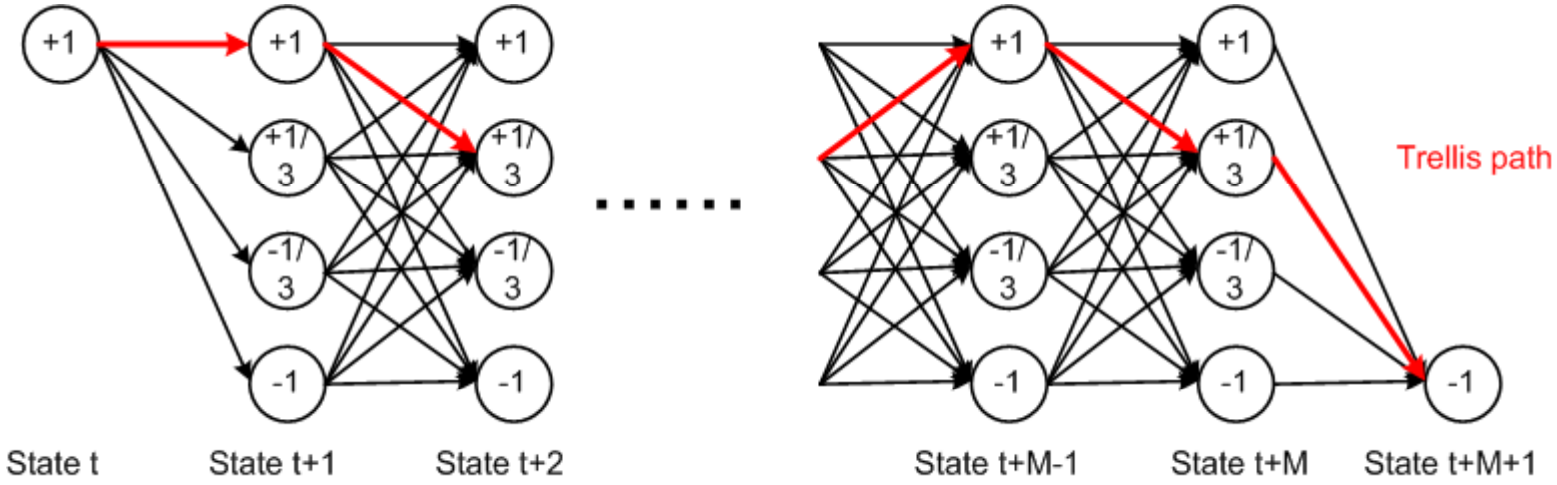
Equation 5: Area is a linear function of M

- Increasing *M* can improve utilization, however the hardware area is a linear function of *M* and latency is also some function of *M*.
- For a reasonable hardware latency and complexity, for 100G Base KP4, utilization of close to **50%** is achievable.

Block Termination

- One easy way to improve the hardware utilization of the VA is to terminate the transmit blocks with known symbols.

- On the receiver, the block based VA is aligned to these termination symbols and since the state of termination is known, there is no reason to assume multiple starting and ending states anymore. This means
  - The synchronization problem is eliminated. Ie K=1
  - The survivor path should merge on the last known state, Ie L=1



Block Termination

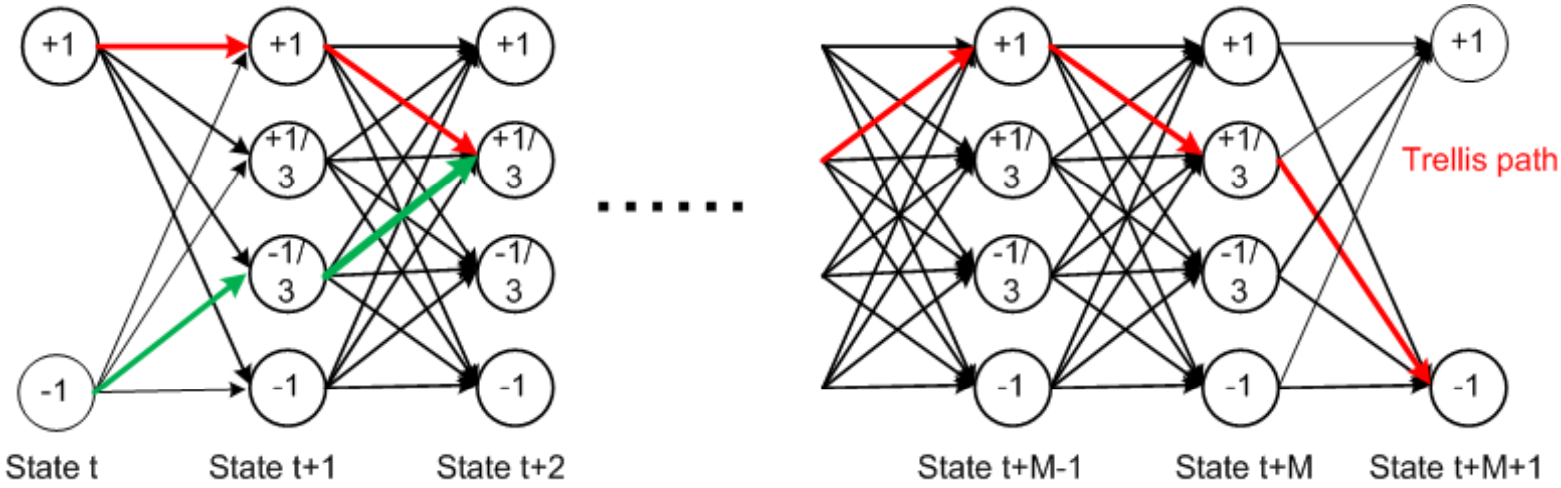
- By adding fix termination symbols to the transmitted stream, the utilization of the block based VA is now

Equation 6: Utilization=M/(M+2)

- For a typical 100G Base KP4 digital implementation of VA, M=32. (This will result in digital clock of about 400MHz). With M=32, the utilization is about 94% per Eq.6. This is almost 2X of utilization increase (or 2X area reduction) compared to the non-terminated case.

Block Termination

- Dariush Dabiri from Applied Micro showed in dabiri_01b_0112 that the block termination can further be optimized by transmitting one information bit in the termination symbol.
- The termination symbol is now a PAM2 symbol and the scheme is called "*Partial termination*".
- dabiri_01b_0112 also showed that there is no practical performance difference between partial termination and full termination



Block Termination

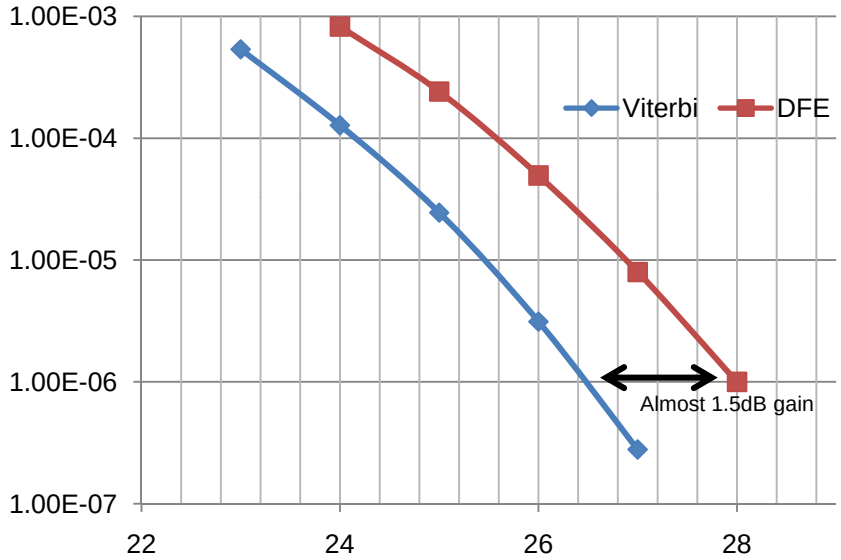
- Full-Termination and Partial-Termination shown in previous pages are not the only way that block based VA can be simplified.

- Among many other termination options, a more standard way of terminating a block is to make the termination symbol some sort of parity (like even or odd parity)

- Generally there is some sequence estimation gain due to adding termination which exceeds the loss due to increased baud-rate. This gain varies as a function of channel and the termination scheme

Simulation results

- Reference Channel model was chosen from “Measured ATCA Backplane channels” uploaded Wolfgang Meier Sep-20-2011
- IBM ran PAM4 simulation on Emerson Channel (from the above) called Thru_S06-P20-10-AB_S14-P23-04-CD_NNN.s4p. The result are in meghelli_01_0112.pdf



- A simplified time domain sim environment (compared to IBM sims) was used to compare Viterbi and DFE solutions
- The simulation is at baud rate with a 3 tap TX FFE. No jitter is included and package is not modeled.
- The channel has some reflections. Very large FFE or a floating tap FFE on the receiver could have improved both Viterbi and DFE results. This is was not added.
- The DFE simulation is running at a **lower baud** rate with block termination removed.

Impact of 45/46 block termination on 100G KP4 systems

- 100Gbps copper systems are operating to their limits and not much margin is left on the table
- Implementation tradeoffs and tools will be key for Phy developers
- 45/46 partial block termination helps the RX ability to correct a little more errors
- “A little” some times can be all the difference needed to make the system work.
- Penalty of baud-rate increase and added hardware complexity due to termination is minor compared to the benefit of a working system

Summary

- Block based VA implementation results in substantial gate count reduction compared to the look-ahead implementation
- Block termination will almost result in 50% area reduction of the VA.
- With Block termination, the VA implementation becomes an area competitive option for 100G Base KP4.
- With block terminated VA implementation, the sequence estimation is improved due to
  - The optimality of the VA algorithm (compared to DFE for example)
  - And some gain resulting from the termination symbols
- Penalty of baud-rate increase and added hardware complexity due to termination is minor compared to the benefit of a working system

802.3bj team did a good job adding partial block termination to 100G Base KP4