Tx Equalizer Coefficient issues

Andre Szczepanek
Texas Instruments
a-szczepanek@ti.com

Comment #64

- **The off-axis requirements in table 72-11 do not match the governing equations of the transmit equalizer. All 3 measurement points are dependant on all 3 coefficients.**
  - If Vpk is kept constant, a step on any coefficient will affect at least two of the 3 measurement points.
  - If Vpk is not kept constant, a step on any coefficient will affect all 3 measurement points.
- **Recalculate the off-axis entries based on the governing equations of the transmit**

Comment #65

- **Draft 2.0 required that C_0 shall be adjusted to maintain Vpk/A over all transmitter states (k). This requirement has been removed in Draft 2.1, and the transmitter output waveform requirements have been changed to render constant Vpk implementations non-compliant.**
- **Implementing Tx equalization on SERDES using assignable CML output fingers is an area-efficient alternative to DAC style structures. Forty fingers of 2.5% meet the performance requirements adopted in May Motion #10, whilst automatically providing constant Vpk. However the coefficient step trading (to/from C_0) required to maintain constant Vpk mean that the measured step changes in Table 72-11 are doubled.**
- **We are concerned that the changes in Draft 2.1, preclude the use of natively constant-Vpk transmit structures for no demonstrable benefit.**
  - Of course it is possible to make a fingered approach work with non-constant Vpk, by doubling native resolution, or by turning fingers off, but this increases transmitter complexity and area for the dubious benefit of reduced output swing.
- **Re-instate the constant Vpk requirement, and reflect this requirement in Table 72-11 values.**
  - Or allow constant Vpk by providing an additional or modified Table 72-11.

Observations

- **There are 2 schools of thought on TX FIR implementations**
  1) Constant V_{pk}
     - Does not require Cursor control
     - Maximizes signal strength
     - Usually augmented by an overall gain control
       - Allows signal strength to be adjusted independent of equalization
       - Allows link power reduction
     - 3 tap FIR only has 2 degrees of freedom (128 states)
       - Leads to simpler, more efficient, implementations
  2) Independent coefficients (Variable V_{pk})
     - Analogous to Rx DFE coefficients
       - Training algorithms resemble Rx algorithms
     - 3 tap FIR has 3 degrees of freedom (>1000 states ?)
     - Difficult to adjust signal strength just via coefficients
       - Still need gain control
- **All contributed Tx FIR analysis has been for constant V_{pk}**

Changes

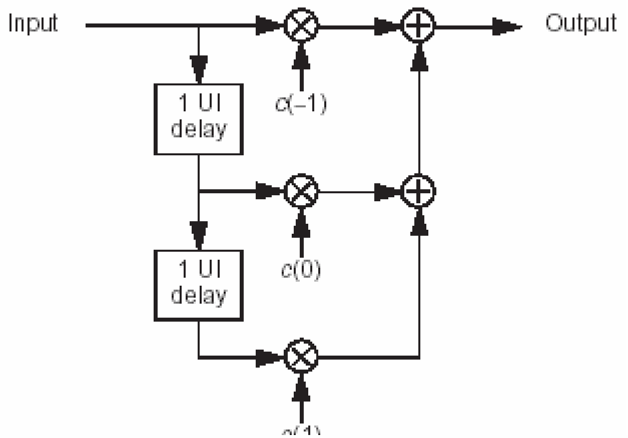
- **The changes recommended in healey_01_0905 to clean up the transmitter waveform tests removed the constant V_{pk} requirement.**
    - Rather than just removing a requirement on non-constant Vpk transmitters it made constant Vpk transmitters non-compliant
  - The bulk of this presentation was on improving the test methodology
  - In this context, I don't believe the practical impact of removing V_{pk} was realized.
    - It wasn't discussed
  - I have voted NO on the ballot based on this change to ensure that we discuss it now
- **There may be practical reasons to limit ourselves to one style of equalizer**
  - Is there an equalizer training algorithm that will work with both ?
    - Maintaining constant Vpk in the training commands will work

Conclusion

- **I am concerned that in order to reach consensus we have ended up with a lowest common denominator approach.**
  - Efficient constant Vpk implementations are now excluded
  - We still have issues with (comments on) the Tx test methodology
- **I'd like to see a return to a constant Vpk based test methodology.**
  - Maintain Vpk by changing C_0 in concert with pre or post cursor in the training packets.
    - This is compatible with either implementation
- **Are we wise to not define an algorithm ?**

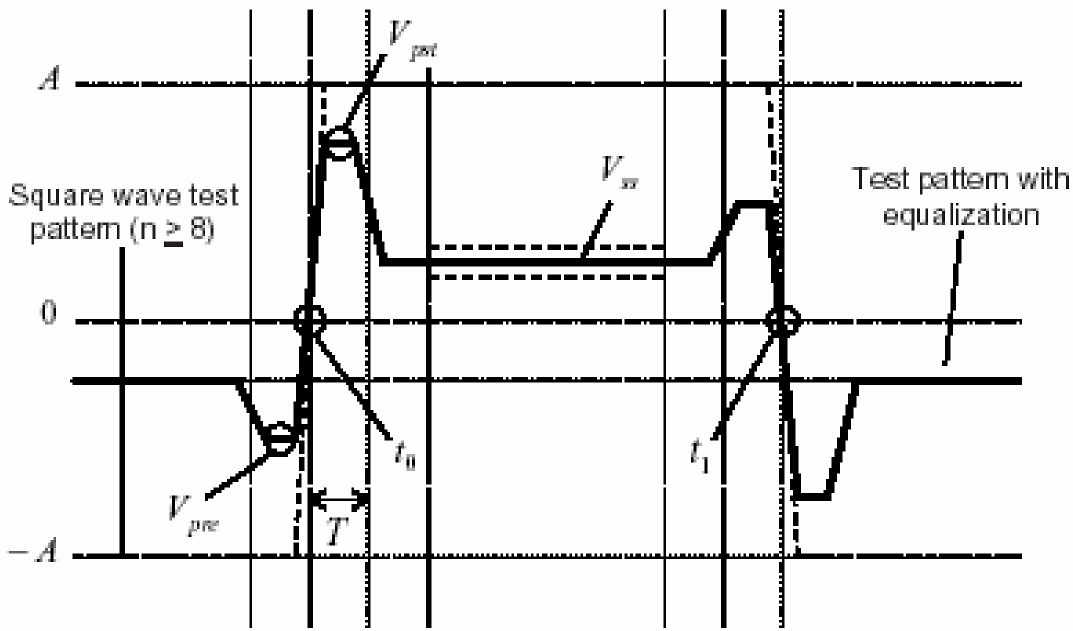
Backup slides

Measurements are not independent



- $V_{pre} = +c(-1) - c(0) - c(+1)$
- $V_{pst} = +c(-1) + c(0) - c(+1)$
- $V_{ss} = +c(-1) + c(0) + c(+1)$
- $A = V_{pst} - V_{pre} - V_{ss}$

  $= -c(-1) + c(0) - c(+1)$



- Note c(-1) & c(+1) are always negative !
- So A = |c(-1)|+ |c(0)|+ |c(+1)|

TEXAS INSTRUMENTS

Interdepenency (A not constant)

- **V_{pre}, V_{pst}, V_{ss} & A are all sums of the 3 coefficients**
  - They differ only in the sign of the coefficients
- **If A is not kept constant, a step in any coefficient will cause an equal changes in V_{pre} , V_{pst} & V_{ss}**
  - The effect will differ only in sign
    - c(-1)++ causes V_{pre}++ , V_{pst}++ & V_{ss}++
    - c(0)++ causes V_{pre}-- , V_{pst}++ & V_{ss}++
    - c(+1)++ causes V_{pre}-- , V_{pst}-- & V_{ss}++

Interdepenency (with constant Max amplitude)

- **To keep A constant (-c(-1) +c(0) -c(+1)) must be kept constant.**
  - A change in one coefficient must be offset by an equivalent total change in the other 2 coefficients.
- **Practical implementations cannot arbitrarily scale output.**
  - The high speed DSP needed to scale the output to keep A constant is not practical at 10Gbps
  - Must make simple changes to the other coefficients Eg
    - Change other two coefficients by ½ step each
    - Or change one of the other coefficients by one step

Effect of ½ step trade

- **Consider a change in one coefficient offset by ½ step changes in the other 2 coefficients.**
  - If c(-1)++, C(0)+= ½, C(+1)-= ½
    - V_{pre} = +c(-1) –c(0) -c(+1) = V_{pre} +=1
    - V_{pst} = +c(-1) +c(0) -c(+1) = V_{pst} +=2
    - V_{ss} = +c(-1) +c(0) +c(+1) = V_{ss}

- **One measurement point changes by one step, another by two steps, the other stays the same.**

Effect of 1 step trade

- **Consider offsetting a change in c(-1) or c(+1) by a 1 step change in c(0).**
  - If c(-1)++, C(0)++, no change in C(+1) for constant A
    - V_{pre} = +c(-1) –c(0) -c(+1) = V_{pre}
    - V_{pst} = +c(-1) +c(0) -c(+1) = V_{pst} +=2
    - V_{ss} = +c(-1) +c(0) +c(+1) = V_{ss} +=2
  - If c(+1)++, C(0)++, no change in C(-1) for constant A
    - V_{pre} = +c(-1) –c(0) -c(+1) = V_{pre} -=2
    - V_{pst} = +c(-1) +c(0) -c(+1) = V_{pst}
    - V_{ss} = +c(-1) +c(0) +c(+1) = V_{ss} +=2
- **Changes in c(0) must be offset against c(-1) or c(+1)**
  - How to decide which ?