Proposal for training patterns for the start-up protocol

Adee Ran, Cisco

Kent Lusted, Intel

Supporters

- Omer Sella, Imperial College London
- Whay Lee, Marvell
- Mike Dudek, Marvell
- Margaret Johnston, Cadence
- Mau-Lin Wu, Mediatek
- Pirooz Tooyserkani, Cisco
- Arthur Marris, Cadence
- Zvi Rechtman, NVIDIA
- Ali Ghiasi, Ghiasi Quantum/Marvell
- Mike Wingrove, Ciena
- Luz Osorio, Ciena
- Paul Brooks, Viavi
- Matt Brown, Alphawave
- David Cassan, Alphwave
- Leon Bruckman, Huawei
- Dave Estes, Spirent

Introduction

- The issues with existing training patterns were initially presented in the ad hoc presentation ran_3dj_elec_01_240208.
    - Brief summary of the problem statement: the existing patterns are created by zero-padding a PRBS13Q sequence to a length that is a multiple of 32 UI. When these patterns are processed by a time-interleaved (polyphase) ADC, the pattern is repetitive with a short period, and is unsuitable for training.
- A proposal was made to add new training patterns based on free-running PRBS13 or PRBS31 generator, with no other change of the training frame format.
    - These new patterns are very simple to implement and test and free-running PRBS generators are already included in most SerDes.
- It was demonstrated that the proposed training pattern has more uniform DC content across 64 phases than the existing ones.
    - Other properties are also improved by this choice.

Proposal

- Keep the existing training patterns and their encoding in the **modulation and precoding request** field (PAM2, PAM4, PAM4 with precoding – based on PRBS13)
- Add another training pattern created from a **free-running PRBS13 generator** (with same per-lane polynomial as in the existing pattern), without the zero pad symbols
  - The Marker/Control/Status DME portion periodically overrides the PRBS13 generator output (288 UI every 16672 UI – same as in clause 136)
  - The PRBS generator is not stopped or reset
- Add a similar training pattern using **free-running PRBS31** instead of PRBS13, without changing the frame structure
  - In this case, the same polynomial (Eq 49-2 from IEEE Std. 802.3-2022) is used in all lanes. The offset between PRBS31 generators on different lanes should be made sufficiently large to effectively decorrelate their patterns.
  - Two options when used with PAM4 encoding: with/without precoding.
- The new patterns will be selectable in training using a “pattern request” variable (replacing the “modulation and precoding” variable) in the control and status fields.
- Training starts with PAM2 as in clause 136/162. A receiver may ask to switch to another pattern (possibly more than once) based on its preference.
  - The final pattern cannot be PAM2 (same as Cl 136/162).
  - Precoding is used after training if the final training pattern included precoding.
- **All the above are to be added in Annex 176A**.

Suggested changes to the control field

Current control field structure (clause 162)

9:8	Modulation and precoding request	9 8 1 1 = PAM4 with precoding 1 0 = PAM4 0 1 = Reserved 0 0 = PAM2
7:5	Reserved	Transmit as 0, ignore on receipt
4:2	Coefficient select	4 3 2 1 0 0 = Reserved 1 0 1 = c(–3) 1 1 0 = c(–2) 1 1 1 = c(–1) 0 0 0 = c(0) 0 0 1 = c(1) 0 1 x = Reserved

Proposed change

Bit 7 can be interpreted as “PRBS31 enable”, and PRBS31 is always free-running.

9:7	Pattern request	9 8 7 1 1 1 = PAM4 free-running PRBS31 with precoding 1 0 1 = Reserved 0 1 1 = PAM4 free-running PRBS31 0 0 1 = PAM2 free-running PRBS31 1 1 0 = PAM4 PRBS13 with precoding 1 0 0 = PAM4 PRBS13 0 1 0 = PAM4 free-running PRBS13 0 0 0 = PAM2 PRBS13
6:5	Reserved	Transmit as 0, ignore on receipt
4:2	Coefficient select	4 3 2 1 0 0 = Reserved 1 0 1 = c(–3) 1 1 0 = c(–2) 1 1 1 = c(–1) 0 0 0 = c(0) 0 0 1 = c(1) 0 1 0 = Reserved 0 1 1 = Swing control

Bits 9:8 retain their current meaning, except for the currently reserved combination “0 1” that becomes “PAM4 free-running” (with PRBS13).

Subject of a separate proposal

Suggested changes to the status field

Current status field structure (Clause 162)

14:12	Reserved	Transmit as 0, ignore on receipt
11:10	Modulation and precoding Status	11 10 1 1 = PAM4 with precoding 1 0 = PAM4 0 1 = Reserved 0 0 = PAM2

Proposed change

14	One	Transmit as 1
13	Reserved	Transmit as 0, ignore on receipt
12:10	Pattern status	12 11 10 1 1 1 = PAM4 free-running PRBS31 with precoding 1 0 1 = Reserved 0 1 1 = PAM4 free-running PRBS31 0 0 1 = PAM2 free-running PRBS31 1 1 0 = PAM4 PRBS13 with precoding 1 0 0 = PAM4 PRBS13 0 1 0 = PAM4 free-running PRBS13 0 0 0 = PAM2 PRBS13

Same values as in “Pattern request” in the control field.

Thanks!

Problem statement

- Currently the training patterns are created from two full cycles of PRBS13Q followed two 0 symbols (zero pad)
- This makes the training pattern consistent across frames
- However, it can create issues in prevalent time-interleaved (polyphase) ADC implementations…
  - For example, with a 64-phase ADC, the pattern seen on each phase of the ADC repeats itself every two frames (it has a period of 16672/64*2=521 samples)
  - These patterns are not PRBS of any kind, and are unfriendly for calibration/adaptation algorithms
  - Notably, these patterns are very unbalanced (see next slide)
- The training pattern spectrum and statistics are not representative of "mission" data
  - Once data mode is entered, the statistics change considerably
- Moving to 200G/lane, accurate calibration and training will be more important…
  - **We should get this fixed!**

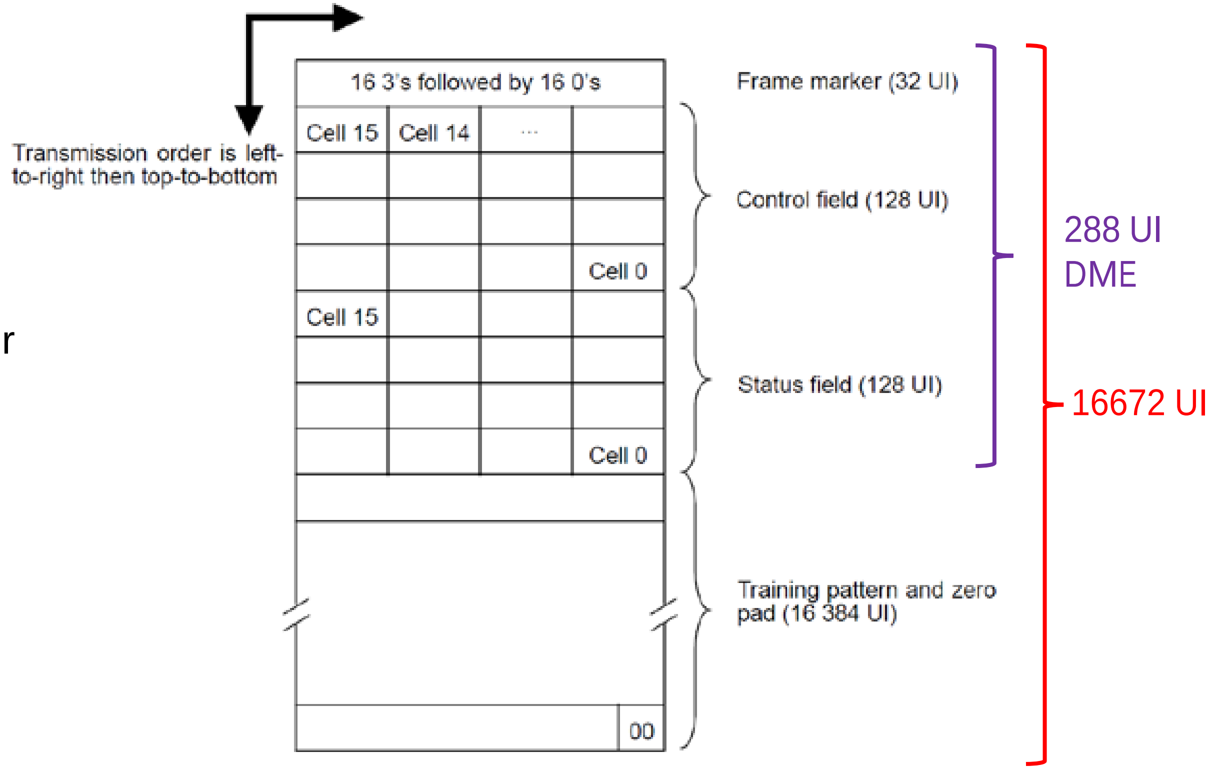
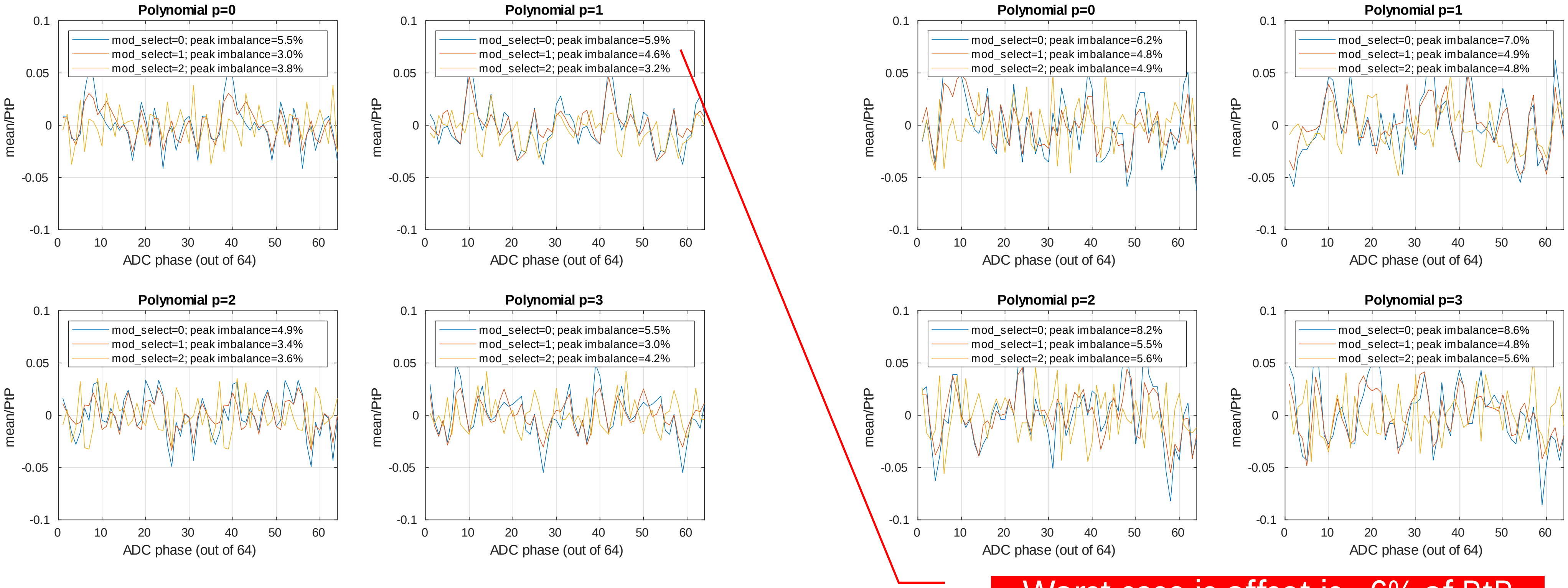

Transmission order is left-to-right then top-to-bottom
16 3's followed by 16 0's
Cell 15
Cell 14
…
Cell 0
Cell 15
Cell 0
00
Frame marker (32 UI)
Control field (128 UI)
Status field (128 UI)
Training pattern and zero pad (16 384 UI)
288 UI DME
16672 UI

Figure 136–3—Training frame structure

Means of the clause 136 training patterns on each phase of a 64-UI subsampled pattern

Full training frame

Training pattern only

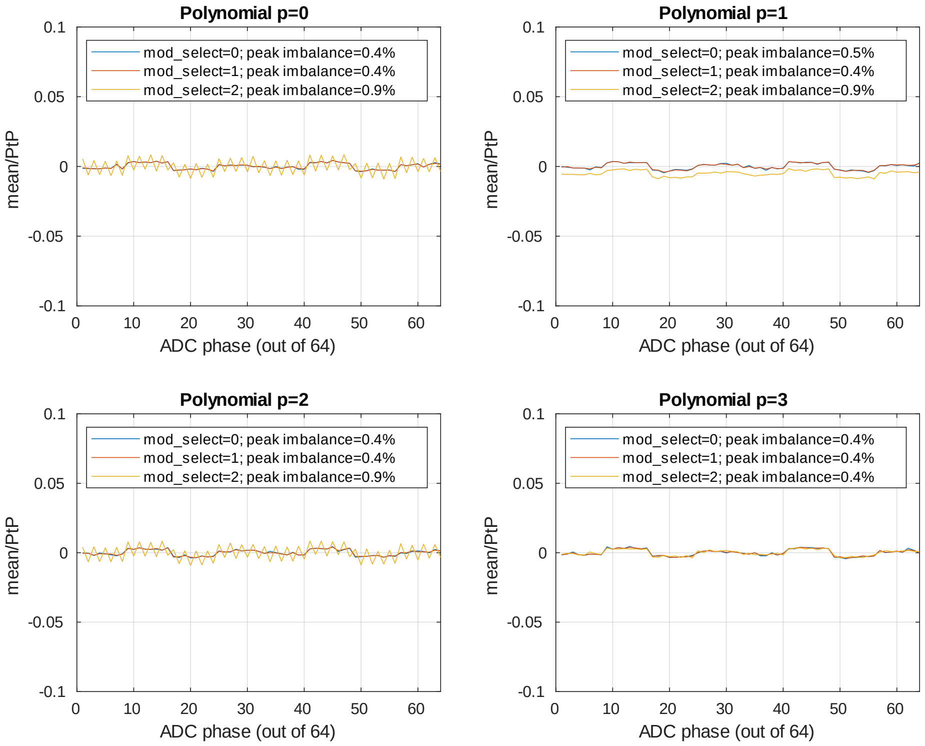


Worst case is offset is ~6% of PtP

mod_select: 0=PAM2, 1=PAM4, 2=PAM4 with precoding

Means of the modified training patterns (free-running PRBS13), same subsampling

Full training frame

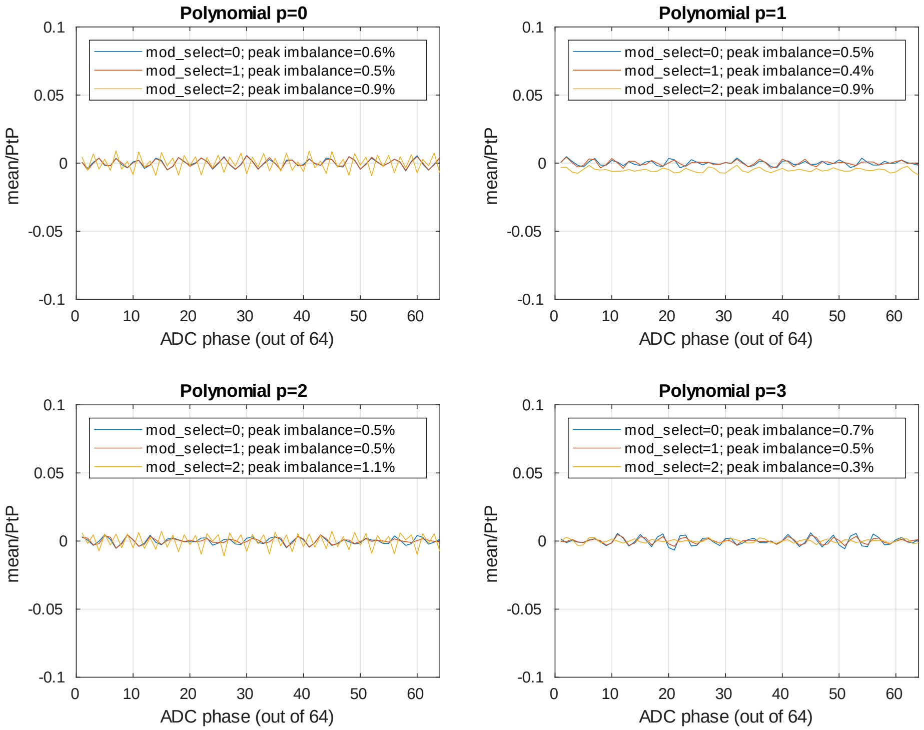

Polynomial p=0

Polynomial p=1

Polynomial p=2

Polynomial p=3

Training pattern only


Polynomial p=0

Polynomial p=1

Polynomial p=2

Polynomial p=3

Worst case is offset is ~1% of PtP

mod_select: 0=PAM2, 1=PAM4, 2=PAM4 with precoding