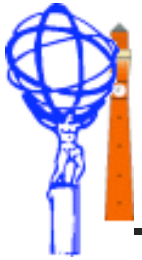
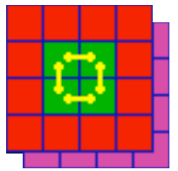


CPM timing calibration issues

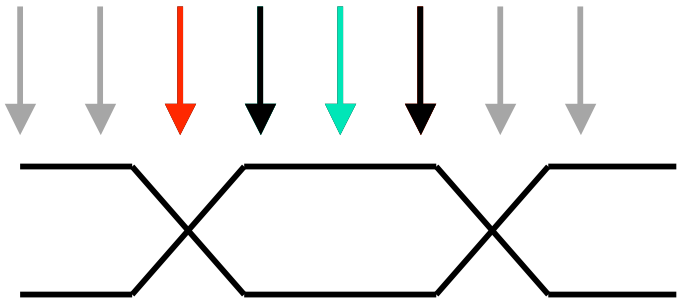


- Scope of talk:
  - Mostly about CP Chip 160 Mbit/s input
    - Similar arguments apply to Serialiser input
  - Future calibration strategy
    - Firmware/improved firmware/software
  - NOT about the current 4-phase problems

- Current firmware calibration algorithm
- Observations from hardware experience
- Improved algorithm?
- Firmware vs Software

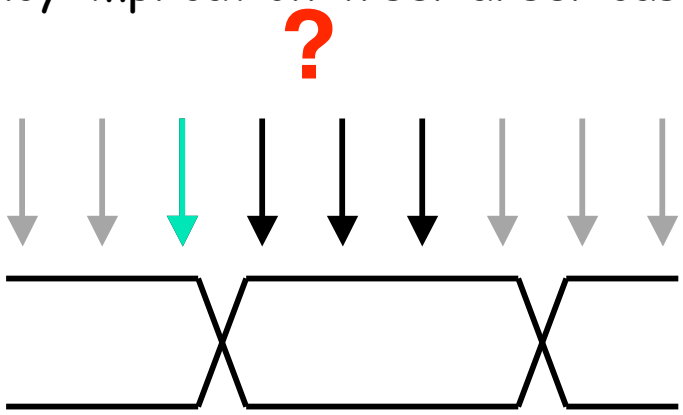
Current calibration algorithm

- Count errors on four phases
- Look for worst
- Choose opposite phase



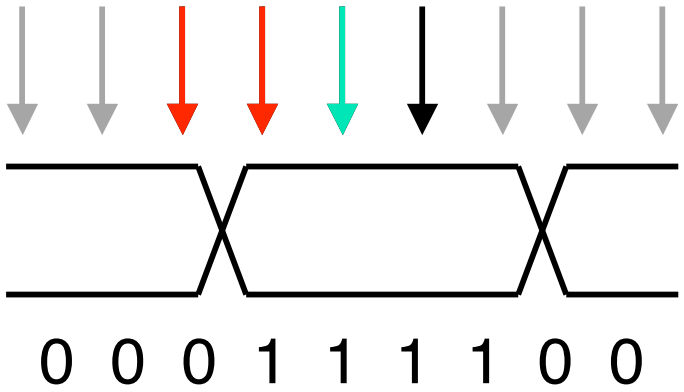
Hardware observations

- Eye patterns (on-board) look very clean and wide
- Timing scans give <1ns transition period
- Quite possible to get 4 'good' phases
- Algorithm uses arbitrary choice – could be close to an edge
  - Also latency implication in serialiser case

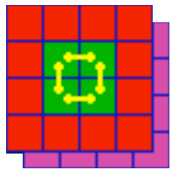


Possible Improved Algorithm

- Use value change as extra information
  - Necessary when all four phases are good
  - Backup/confirmation for three good phase cases
- Set phase at safe distance from transition
  - Can control latency better this way too

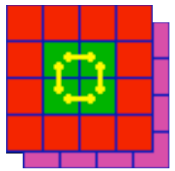


Software vs Firmware



- New algorithm is more complex
  - At least two ways to choose phase
  - Could think of even more sophisticated additions
- All these algorithms could be duplicated in software
  - Software has more time and is more easily debuggable
  - More flexibility for pathological cases
- Calculated phases will be stored in database
  - Reloaded at the start of each run
  - No need to do firmware calibration every time

Conclusions



- If we go the four-phase route, calibration needs improvement
  - Applies to both serialiser and CP chip
- Improvement could just mean software intervention
  - Need to moderate firmware calibration using a software correction – needed anyway to get slice alignment correct
  - In that case, firmware logic can stay as it is
  - But then again, do we need a firmware calibration at all?