# Arbitration Enhancements

Jerry Hauck/Ganesh Murthy
Intel Corporation

- Is ***enab_accel*** adequate to handle both ack accelerated and fly-by arbitration enhancements?
- What should be the power up state?
- What is the usage model?
- Is there a better way to enable arbitration enhancements?

- ***enab_accel*** is adequate to handle all arbitration enhancements.
  - Both ack accelerated and fly-by ack accelerations occur only during the asynchronous phase. No need for separate bits.
  - Cycle Starts disable both ack accelerated and fly-by ack acceleration.
- ***enab_accel*** should power up off or disabled
  - Leaving it turned on may cause cycle starvation since legacy link can't disable it
  - Not optimal but acceptable for P1394a Links
  - Power up Defaults of all PHY registers needed in draft 0.09
- ***enab_accel*** should be turned on if and only if:
  - P1394a PHY is a root
  - Attached link can generate CycleSync LREQ
  - P1394a Link or P1394a aware higher layer will turn ***enab_accel*** bit ON

- There is a better way!

  - Let the P1394a PHY “learn” when to turn on ***enab_accel***...
  - ***enab_accel*** bit powers up disabled
  - P1394a PHY detects it is root and automatically turns on ***enab_accel***
  - P1394a PHY gets a CycleSync LREQ, “knows” it is attached to a P1394a LINK and so it turns on ***enab_accel (***one 125 uS Period of no enhancements***)***
  - Lets P1394A PHY work with a 1394-1995 Link if it is root
  - Higher level layers are not allowed to write to ***enab_accel*** bit.