P1394a
Ft Lauderdale
December 4/5 1997

Continuous SClk Proposal

Colin Whitby-Strevens

Motivation

- Significant class of 1394 applications potentially can operate within the single time domain provided by the PHY's SClk
  - significant cost savings by eliminating multiple clock sources and synchronisers
  - simplifies design and accelerates time to market
- These applications only run the Link and Application layers if the PHY is running
- Nearly possible in current proposals, but
  - SClk is not provided until LPS is asserted
  - SClk is stopped during PHY/Link interface reset as part of a handshake
  - SClk duty cycle is inefficient

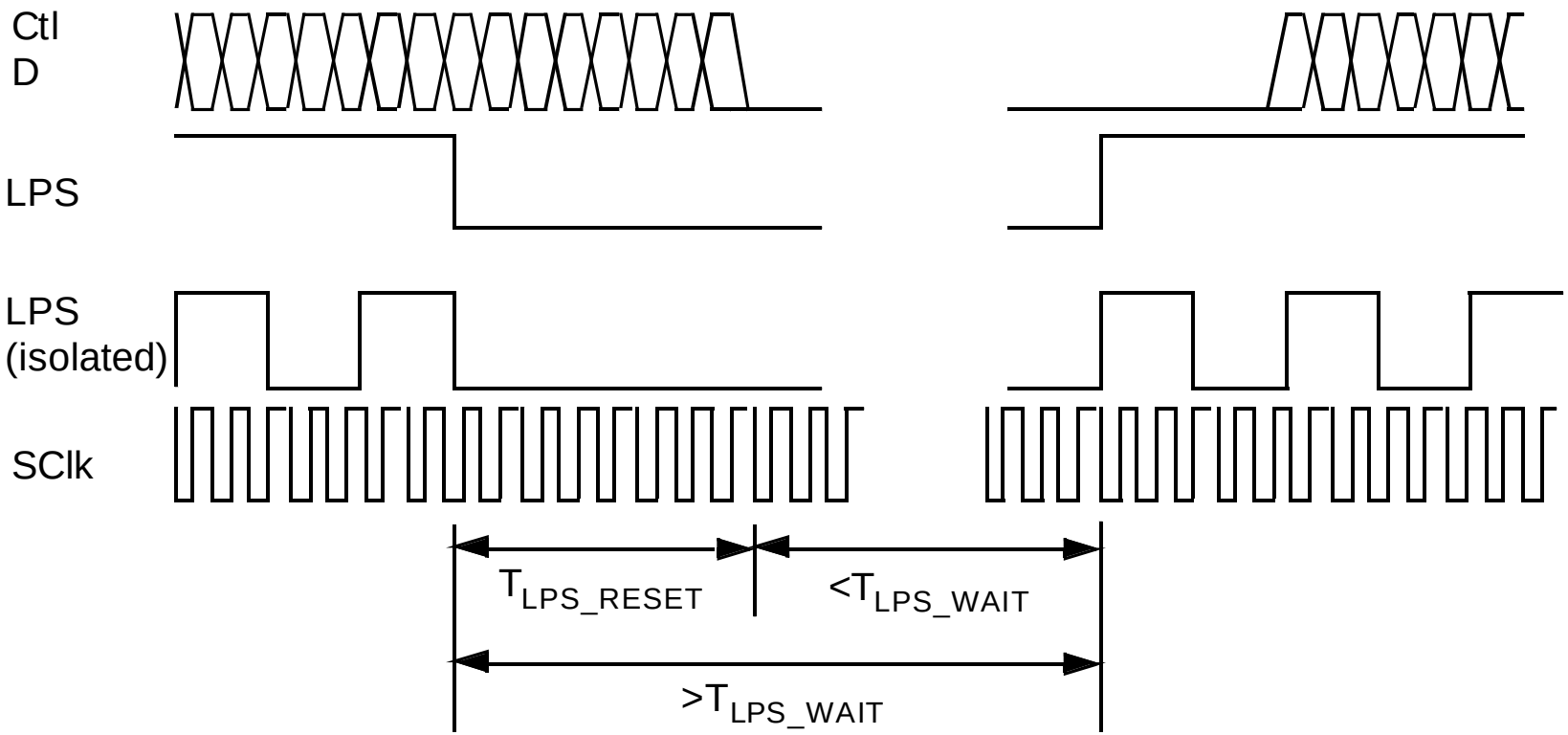
Proposal

- Tighten up duty cycle for SClk
- Change PHY/Link interface specification to distinguish between:
  - **reset** (LPS is deasserted for approx 10 us)
    - ✓ enough for isolation barrier recovery time but no more
  - **disable** (LPS is deasserted for more than 30 us)
- PHY does not take SClk away during reset
- PHY takes SClk away on disable
  - ✓ Link goes into Suspend
  - ✓ PHY core may also Suspend
  - ✓ Resume is initiated only over the bus
  - ✓ PHY asserts SClk before or with LinkOn, so that Link can wake up and generate LPS as a oscillating signal derived from SClk
- Need to define initialization of the interface carefully too (current spec is unclear on this)
  - SClk provided by the PHY after suitable wait to allow isolation barrier to settle, and waits for a limited time to see LPS

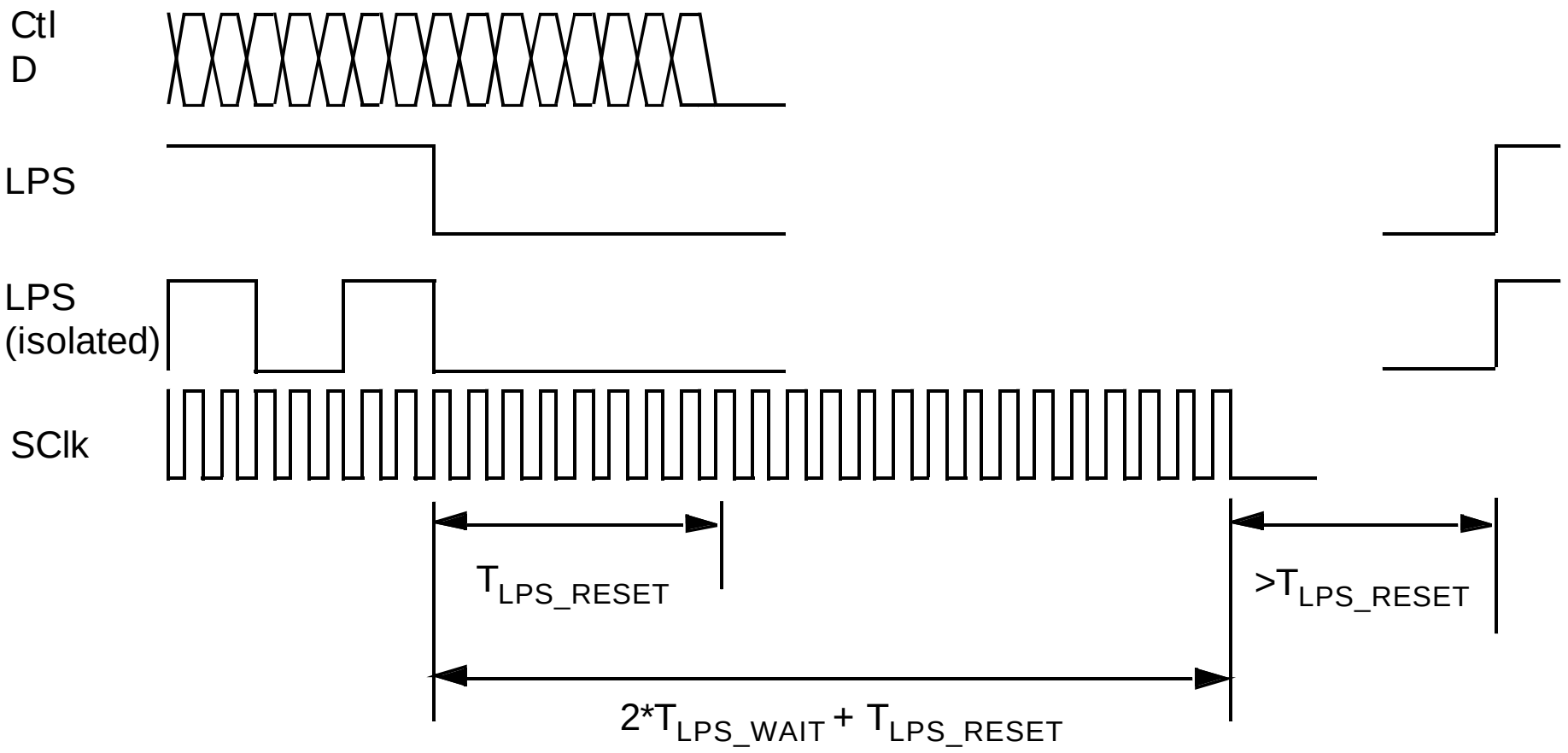
Details

- SClk specification (Table 5-18)
  - The duty cycle is modified to 47.5%/52.5% (say)
  - This increases the effective clock period for the Link/Application from 8ns to 9.5ns.
  - Good thing to do anyway - and believed to be easy, maybe already true, for most/all existing PHY designs
- Initialization and reset (Clause 5.1)
  - Modify the specification of TLPS_WAIT to “LPS deassertion time”.
  - Modify the minimum value for TLPS_WAIT to 12.5 us.
  - Modify text to specify interface reset and interface disable separately
  - Modify reset timing diagram and add new disable timing diagram
  - Define power reset
- LinkOn
  - define to provide SClk

Reset timing diagram



Disable timing diagram



Initialization

- Following a power reset,
  - the PHY shall provide SClk for a period of 2 * TLPS_WAIT
  - then check for the presence of the LPS signal.
  - If the LPS signal is not asserted then the PHY shall reset and possibly disable the interface as if the LPS signal was deasserted at that moment
    - ✓ follow the reset specification
- Once SClk is provided the PHY and link shall condition their Ctl[0:1] and D[0:n] outputs in accordance with table 5-3.
  - In this table, SClk cycles are counted from the later of (i) the time at which SClk is provided, (ii) the time the PHY detects the presence of LPS.
- Need a power reset spec anyway
  - (not difficult, just needs doing)

LinkOn, etc

- Modify to read
  - When either LPS is absent or the PHY register Link-active bit is zero, a PH_EVENT.indication of LINK_ON shall cause the assertion of LinkOn, and shall cause the PHY to provide SClk as defined in 5.1.1. The LinkOn signal shall persist so long as the logical AND of the LPS signal and Link_active is zero.

- Note that “single clock domain” devices are provided with SClk from the PHY before having to assert LPS, and so are able to derive LPS as an oscillating signal from SClk.

Issues

- PHY will generate SClk when Link may not be powered
  - suitable protection is normally provided
  - problem exists anyway with LinkOn and, inthe other direction, with LPS.
- PHY will generate SClk when Link is not expecting it
  - But the PHY will not try to send anything to the Link if there is no LPS, and whenever the Link asserts LPS, it must expect the PHY to send SClk and try to talk to it
- Rules must allow the Link to determine unambiguously when the PHY/Link interface reset has completed (counting the 6 cycles)
- Could make the feature implementation dependent
  - control the Continuous SClk facility by a strap-option and register bit
  - needs thinking through to ensure that start-up is OK