







10GBase-T Active / Low-Power Idle Toggling

Gavin Parnaby
George Zimmerman

Supporters

-  Robert Hays (Intel)
-  Brad Booth (AMCC)
-  Brian Murray (LSI Logic)
-  Adam Healey (LSI Logic)
-  Dan Dove (ProCurve Networking by HP)
-  George Zimmerman (Solarflare)
-  Gavin Parnaby (Solarflare)

Overview

-  Introduction
-  10GBASE-T Low-Power Idle concept
-  Details
-  Signaling
-  Power & Recovery time estimates
-  Conclusions

10GBase-T low power idle concept

- ✈ Power down transmitter and receiver circuits when there is no data to be transferred, saving power
- ✈ PMA and PCS maintain synchronization to enable rapid return to full 10G rate
- ✈ Refresh coefficients using periodic LDPC frame(s)
 - ✈ Multiple off/on rates supported to trade off power vs response time
- ✈ Receiver powers up in time to receive the frame, which contains control codewords to maintain sleep / wake up PHY

Details

- ✚ Add a counting state machine for low power LP_IDLE mode to wake up periodically
 - ✚ Turn off receivers, transmitters for N frames [Solarflare lab work shows static coefficients can maintain link for up to 3 minutes; 1 min equiv to $N=200e6$]; N selected from 10,100,400,1000
 - ✚ Turn on receiver (or transmitter) on schedule for M frames (M is selected from 1, 4)
 - ✚ Refresh timing / coefficients in PAM-16 (no infofield exchange)
 - ✚ M frames (resolved by PHYs during autoneg)
 - ✚ Check for “wake-up” codeword (N_IDLE) / or stay in low power idle mode (LP_IDLE)
 - ✚ Transition back to active mode or go back to “counting sleep” depending on control received
 - ✚ Maintains PMA & PCS structures
 - ✚ Vendor-dependent hardware scheduling

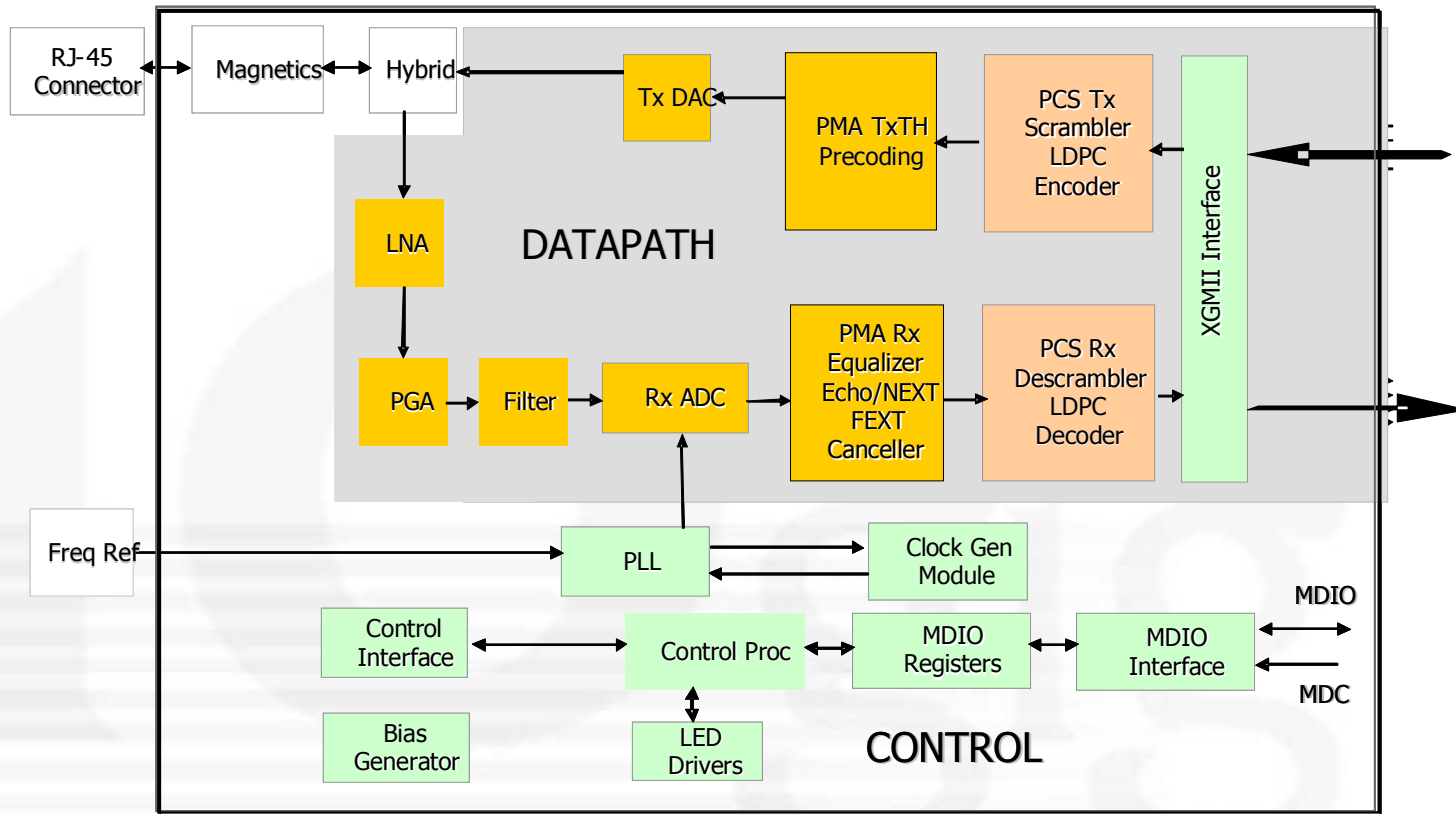
Signaling (I)

- ✈ Autoneg advertises EEE capability
 - ✈ Exchange next pages (XNP)
 - ✈ Advertise max LP_IDLE time and minimum N_IDLE time; implementation dependent
 - ✈ Resolve N and M
 - ✈ M frames of LDPC (ON) (suggest $M=1/4$)
 - ✈ N frames of quiet (OFF) (suggest $N=10/100/400/1000$)
 - ✈ Advertise maximum recovery time – other end needs to know when to start sending data after wake up call; preferably the next frame

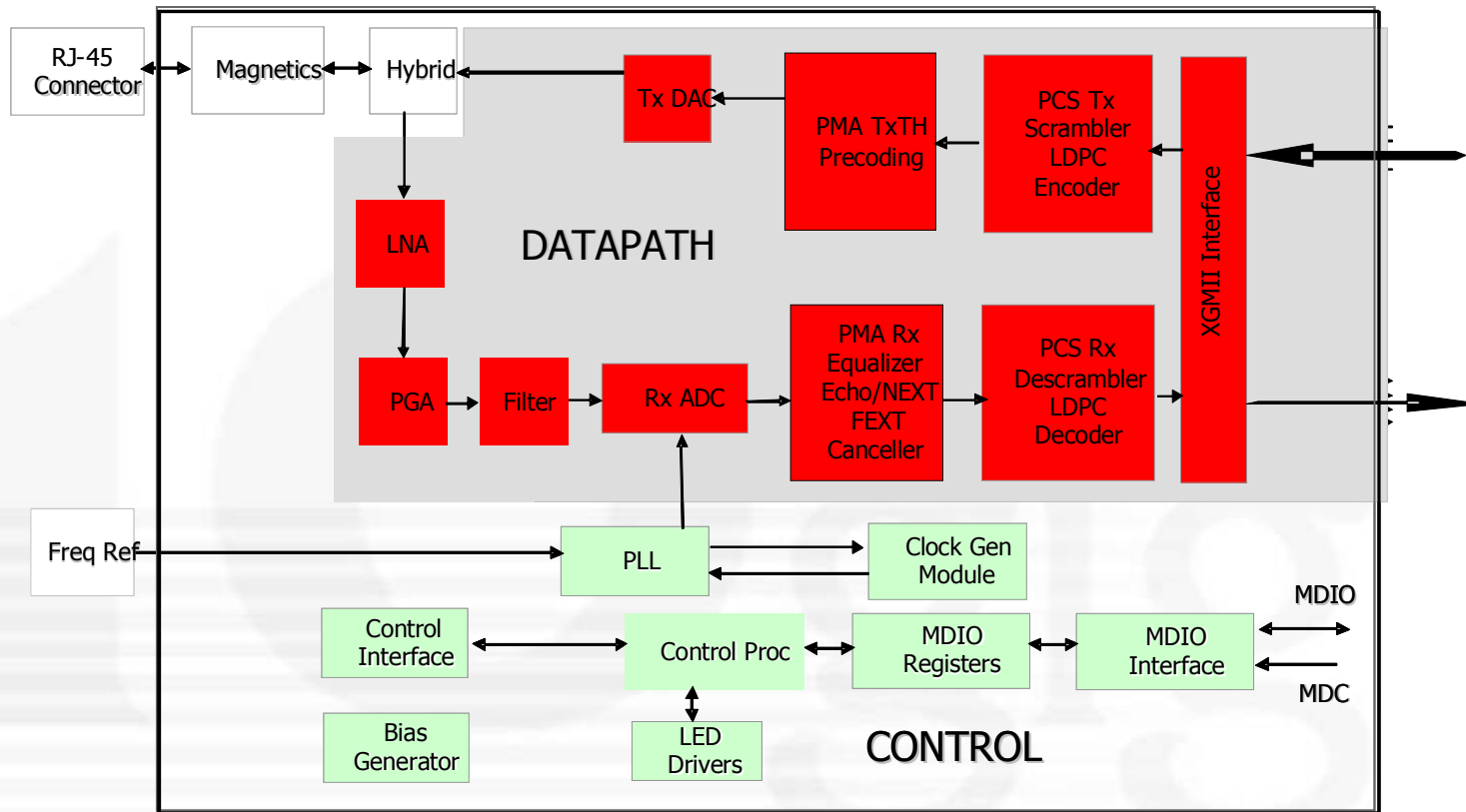
Signaling (II)

- 📎 Use XGMII interface to encode Normal idle vs EEE Low power idle
- 📎 See Dan Dove presentation
- 📎 PCS detects LP_IDLE control code on XGMII / from rx data and transitions PHY to EEE mode (or maintains EEE mode if already enabled)
- 📎 PCS detects N_IDLE control code and transitions back to full 10GBASE-T mode
 - 📎 Recommend symmetric operation
- 📎 Lower latency for on/off with this approach vs MAC controlling PHY via MDIO or fast start-up
- 📎 Scramblers are left on with zero input to maintain state

10GBASE-T Transceiver - On state



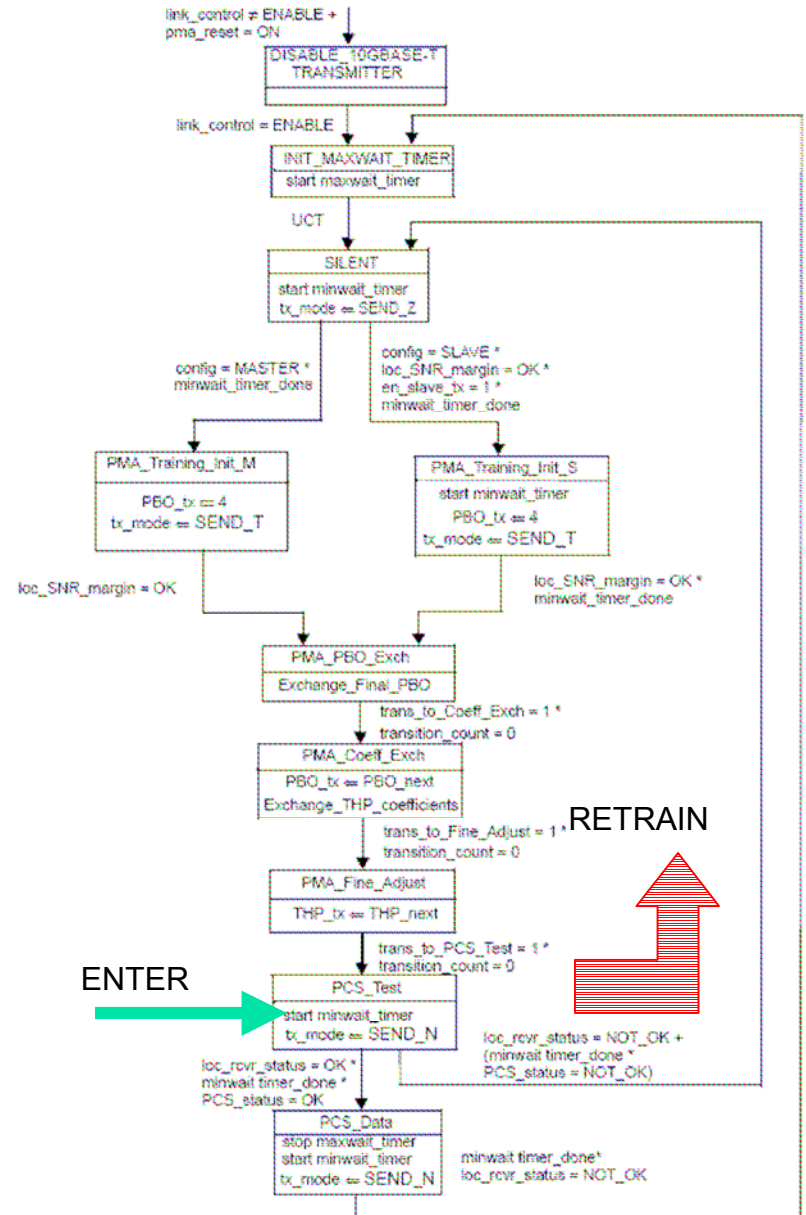
10GBASE-T Transceiver – Low Power Idle state



RED BLOCKS ARE POWERED DOWN OR HAVE CLOCKS GATED BETWEEN PULSED FRAMES

Restart Sequence

- ✎ 10GBASE-T PHY Control State Machine (Fig. 55.4.6.1)
- ✎ Entrance point for EEE state-restore:
 - ✎ PCS_Test (1msec fixed)
 - ✎ Required to maintain link quality
 - ✎ Test time limited by desire to see enough LDPC frames
 - ✎ Full retrain triggered if PCS_Test fails
 - ✎ Forces min time in normal mode > 1ms



Power Estimates

- 🚗 Off_frame power: $P_{\text{off}} = P_{\text{clkgen}} + P_{\text{ctl}} + P_{\text{ovr}}$
 - 🚗 P_{ovr} : vendor-specific overhead for leakage & housekeeping (e.g., MDIO) needed in ANY low-power state
 - 🚗 Current designs conservative estimate: 15% of total power
 - 🚗 Leakage as normal and overhead circuitry is kept at full activity
 - 🚗 Can be reduced
 - 🚗 Higher than assumptions in diab_2_0907

- 🚗 Deep_Sleep_Idle_frame power: $P_{\text{on}} = P_{\text{nominal}} - P_{\text{LDPC}} - P_{\text{ENX}}$
 - 🚗 Current designs estimate: 65% nominal power
 - 🚗 Consistent with overhead+50% analog assumption of diab_2_0907

- 🚗 1:N frame decimation gives:
 - 🚗 $P_{\text{deep_sleep_idle}}: [(N-1) * P_{\text{off}} + (P_{\text{on}})] / N$

- 🚗 1:10 frame decimation, (equiv 1G traffic load)
 - 🚗 $P = [9 * .15 + .65] / 10 = 20\%$ of nominal 10G PHY power level

Recovery Time

- ✈ Low Power Idle transition time will be limited by startup/"sync" time
 - ✈ Maintain PCS and PMA synchronization
 - ✈ Enable blind return to high rate
 - ✈ Limited only by block sizes, latency and prop delay
- ✈ Bring-up time = $T_{\text{interface sync}} + T_{\text{latency}} + T_{\text{next_frame}}$
 - ✈ $T_{\text{latency}} (10\text{GBASE-T}) = 2.5\text{usec}$
 - ✈ $T_{\text{interface sync}} = \text{negligible}$
 - ✈ Time to next frame: $T_{\text{next_frame}}$ can be negotiated
 - ✈ Longer times allow deeper power down (leakage & overhead savings)
 - ✈ Shorter times allow faster transition, less savings
 - ✈ Examples:
 - ✈ 1:10 - $T_{\text{next_active_frame}} = 3.2\text{usec}$, 20% nom pwr
 - ✈ 1:1000 - $T_{\text{next_active_frame}} = 320\text{usec}$, 8% nom pwr
 - ✈ Longer times enable greater overhead savings; full power down
 - ✈ Allow at least x10 steps
- ✈ **Enables < 10 usec transitions, with significant power savings**

Open issues

- ✈ Resolving negotiation of M “on frames” per N “off frames”
 - ✈ Advertise support within small set
 - ✈ Pick N,M using resolution matrix
 - ✈ Allows PHYs to trade off wake up latency vs block size
- ✈ Support for asymmetric operation (recommend symmetric only)
 - ✈ Timing synchronization with tx/rx and adaptive filters adds complexity

Conclusions

- ✈ Low Power Idles can be structured by periodic transmission of LDPC frames
 - ✈ Use inherent framing in PCS of 10GBASE-T
 - ✈ Uses existing PCS and PMA with minimal “flywheel” logic
- ✈ Rate control through MAC control words
- ✈ 10 usec-scale recovery times are achievable
- ✈ Achieves adjustable efficiency better than 10X improvement for the PHY
 - ✈ Much better for the entire system