Supporting material related to comments against Clause 40

Adam Healey, Niall Fitzgerald,
Jacobo Riesco, Brian Murray
LSI Corporation

IEEE P802.3az Task Force Meeting
Dallas, TX
November 2008

Topics

- **1000BASE-T wake time negotiation**
- 1000BASE-T signal_detect definition
- 1000BASE-T PHY Control state diagram

Wake time negotiation

- In Draft 1.0, each PHY advertises the fastest supported wake time and the larger of the two values is used
- Allows robust operation since wake time can be set as a result of field trials and an acceptable rate of failure to wake
  - There will always be some (low) probability of either not waking or getting a symbol error just after wake
  - This probability diminishes with longer wake times
- Also allows wake time to be improved in successive PHY generations
- However, striving for faster wake must result in a trade-off with energy savings
  - The primary aim of Energy Efficient Ethernet is energy savings

Problem statement

- The disadvantage of the current mechanism is that the slowest wake time will always be chosen
  - Applications that requires lower latency are at the mercy of the wake time advertised by the link partner regardless of the fastest wake time that the link partner actually supports
- Also, there is no means to support a wake time that is fast enough but with the greatest energy savings
  - For example, if the system intends to negotiate a 300 microsecond wake time, there is no need to constrain the PHY to a 16 microsecond wake
- Some applications will require greatest energy savings while others require the lowest possible latency

Wake time negotiation proposal

- Define two energy modes: lowest energy and faster wake
- Add a bit to advertise the preferred energy mode
  - 0 = Prefer lowest energy
  - 1 = Prefer faster wake
- If both PHYs prefer lowest energy, resolve to lowest energy and use the lowest energy mode wake time (propose 24 microseconds)
- If either PHY prefers faster wake, resolve to faster wake and use the negotiated wake up time per the current draft

Lowest energy	Faster wake
• Greatest energy savings • Slowest PHY wake time, 24 μs • Used when the system wake is large, e.g. 300 μs • Used by applications requiring the greatest energy savings	• Reduced energy savings • Faster PHY wake time, $\leq$ 16 μs • Used when the system wake is smaller, e.g. 16 μs • Used by applications requiring lower latency

Topics

- 1000BASE-T wake time negotiation
- **1000BASE-T signal_detect definition**
- 1000BASE-T PHY Control state diagram

signal_detect – 1

- signal_detect = FALSE must be detected while the local transmitter is transmitting and hence is intended to be a DSP function
  - This distinction could be made by redefining the condition as zero_detect = TRUE
  - The zero_detect = TRUE assertion time is the receive path latency (T_{rp}) plus the processing time to detect and incoming stream of zeros (T_{zd})
  - Propose that maximum assertion time, T_{sd}, is 0.5 microseconds
  - Care should be taken in the WAIT_QUIET states to ensure that rem_lpi_req is not incorrectly decoded to FALSE prior to zero_detect being set to TRUE (also applies to the proposed POST_UPDATE state)

signal_detect – 2

- signal_detect = TRUE is intended to be an analog function and is only used in the QUIET state
  - The maximum signal_detect = TRUE assertion time, T_{sa}, is suggested to be 0.5 microseconds

- There is one exception…
  - When entering QUIET from the proposed POST_UPDATE state, it may be necessary to wait some time before enabling the analog detector to prevent reflections from triggering it
  - During that time, the DSP function should be used instead (zero_detect = FALSE)

- A new timer, lpi_mask_timer, has been introduced to explicitly enforce this behavior in the QUIET state.
  - Shall also be used to keep transmission enabled while not done (as in WAIT_QUIET) to ensure that if loc_lpi_req is set to FALSE, it can be transmitted in the WAKE state
  - The lpi_mask_timer is suggested to be 2 microseconds (TBC)

Wake-up signal

- The wake-up signal is transmitted during the WAKE state to initiate a refresh or a transition out of low power mode
- This signal may be transmitted while the PHY analog front-end is still powering up, and is not guaranteed (or intended) to be a compliant IDLE signal
- Its only purpose is to cause the link partner to assert signal_detect = TRUE
- It is proposed that this signal should be between 50 to 75% (TBD) of the IDLE levels with the same (+2, 0, −2) symbols ratio as an IDLE signal (with 10% margin)
- After entry into the WAKE state, these requirements must be satisfied within the minimum lpi_waketx_timer less the maximum signal_detect assertion time

Topics

- 1000BASE-T wake time negotiation
- 1000BASE-T signal_detect definition
- **1000BASE-T PHY Control state diagram**

Problem statement

- There is some interest in enforcing that a minimum time is spent in the WAKE_SILENT state to ensure predictable transitions during wake
- A proposal is presented to address this concern while guaranteeing a minimum period of uninterrupted transmission following entry into the UPDATE state to allow adaptive filter coefficient update
  - This is guaranteed in the current PHY Control state diagram

Summary of the proposal

- Force the PHY to remain in the WAKE_SILENT state for a duration of at least lpi_wakemz_timer

- Introduce new POST_UPDATE state, succeeding UPDATE state, to control transitioning into WAIT_QUIET or SEND IDLE OR DATA

- Introduce new variable loc_upd_done
  - Indicates that update of local adaptive filter coefficients has completed
  - Assigned a value of FALSE in the UPDATE state and a value of TRUE in the POST_UPDATE state
  - Communicated to the link partner and received as rem_upd_done
  - Various encodings of loc_upd_done are possible

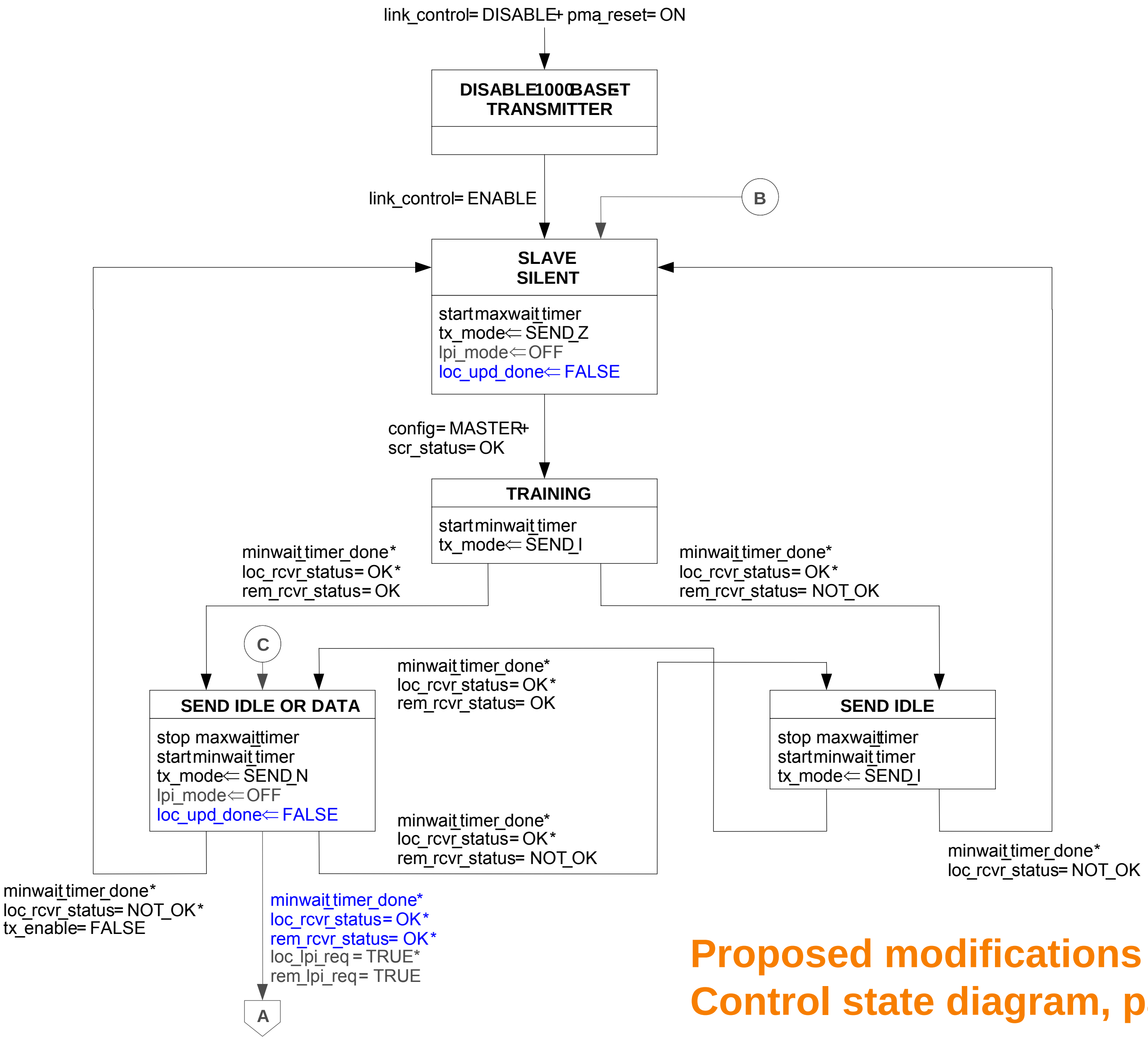
- The transition from WAKE_TRAINING to WAKE_SILENT is no longer required and has been removed

- The lpi_waitwt_timer is no longer required (it was added to combat the fall-through case) and has been removed.

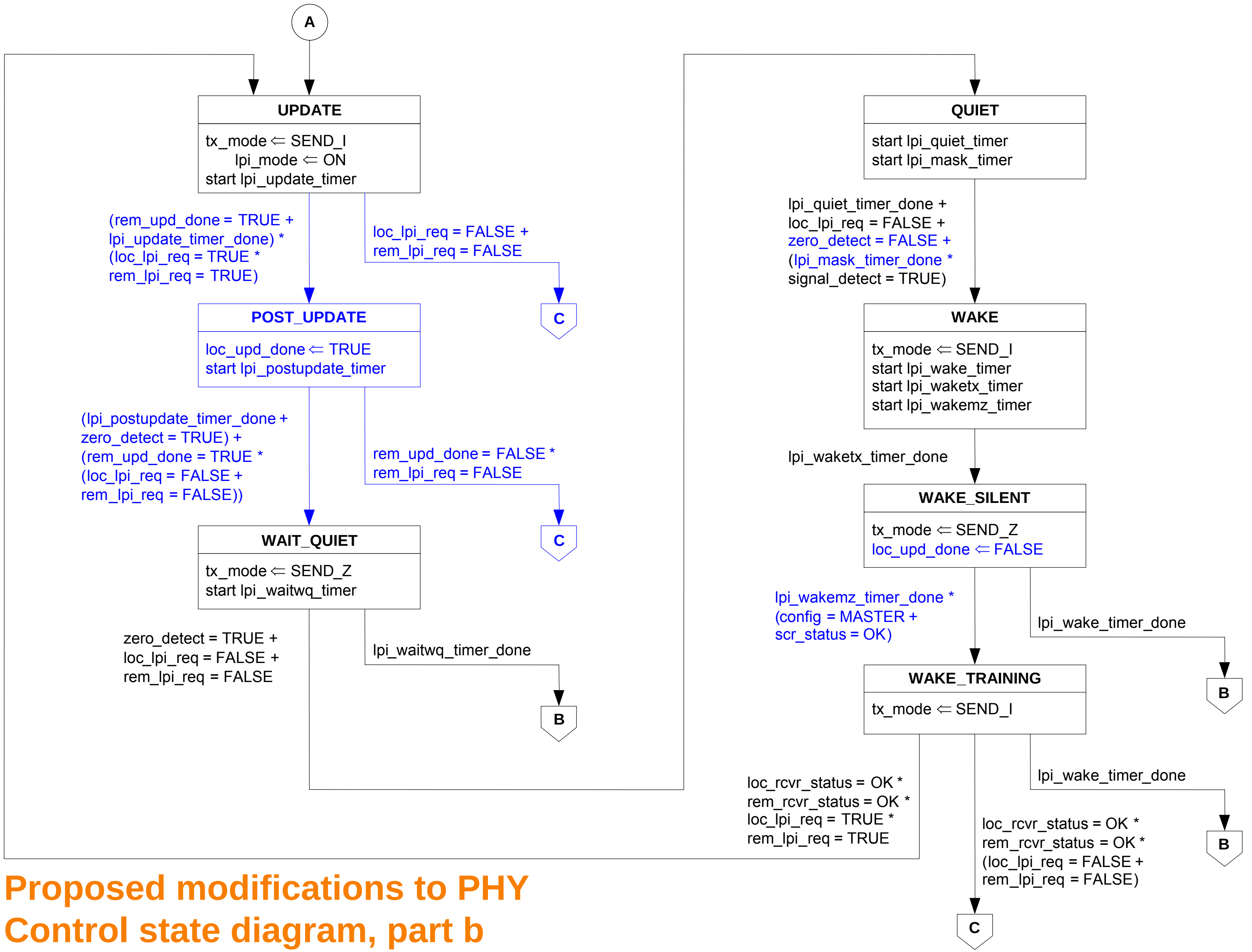
PCS encoding of loc_upd_done

- **40.3.1.3.4 Generation of bits Sd_n[8:0]**
  The bit $Sd_n[1]$ is used to scramble the GMII data bit $TXD_n[1]$ during data mode and to encode loc_upd_done or $cext\_err_n$ otherwise. The proposed definition is as follows:

$$Sd_n[1] = \begin{cases} Sc_n[1] \wedge TXD_n[1], & if\ (tx\_enable_{n-2} = 1) \\ Sc_n[1] \wedge 1, & else\ if\ ((loc\_upd\_done = TRUE)\ and\ (tx\_mode \neq SEND\_Z)) \\ Sc_n[1] \wedge cext\_err_n, & else \end{cases}$$



Proposed modifications to PHY Control state diagram, part a

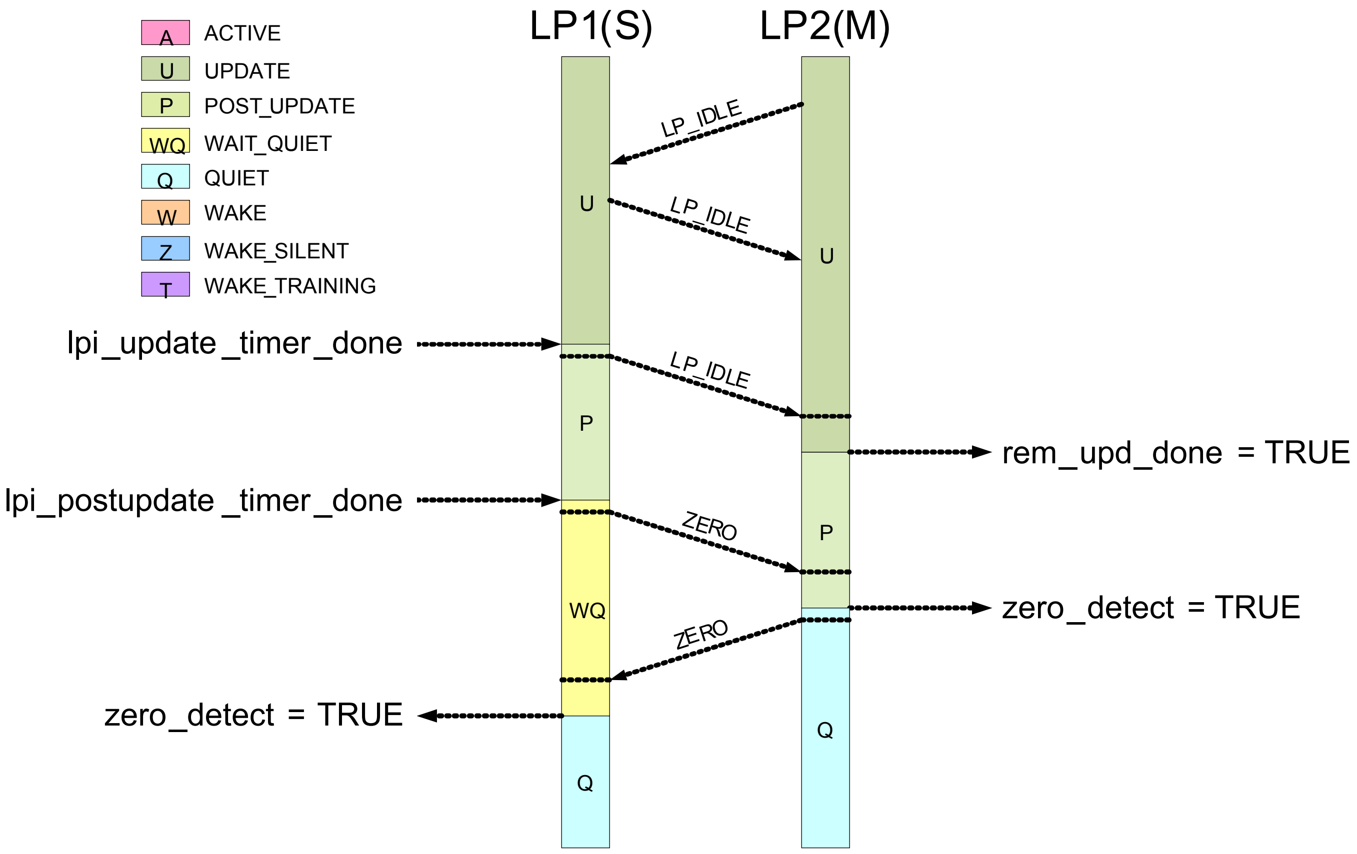


Proposed modifications to PHY Control state diagram, part b

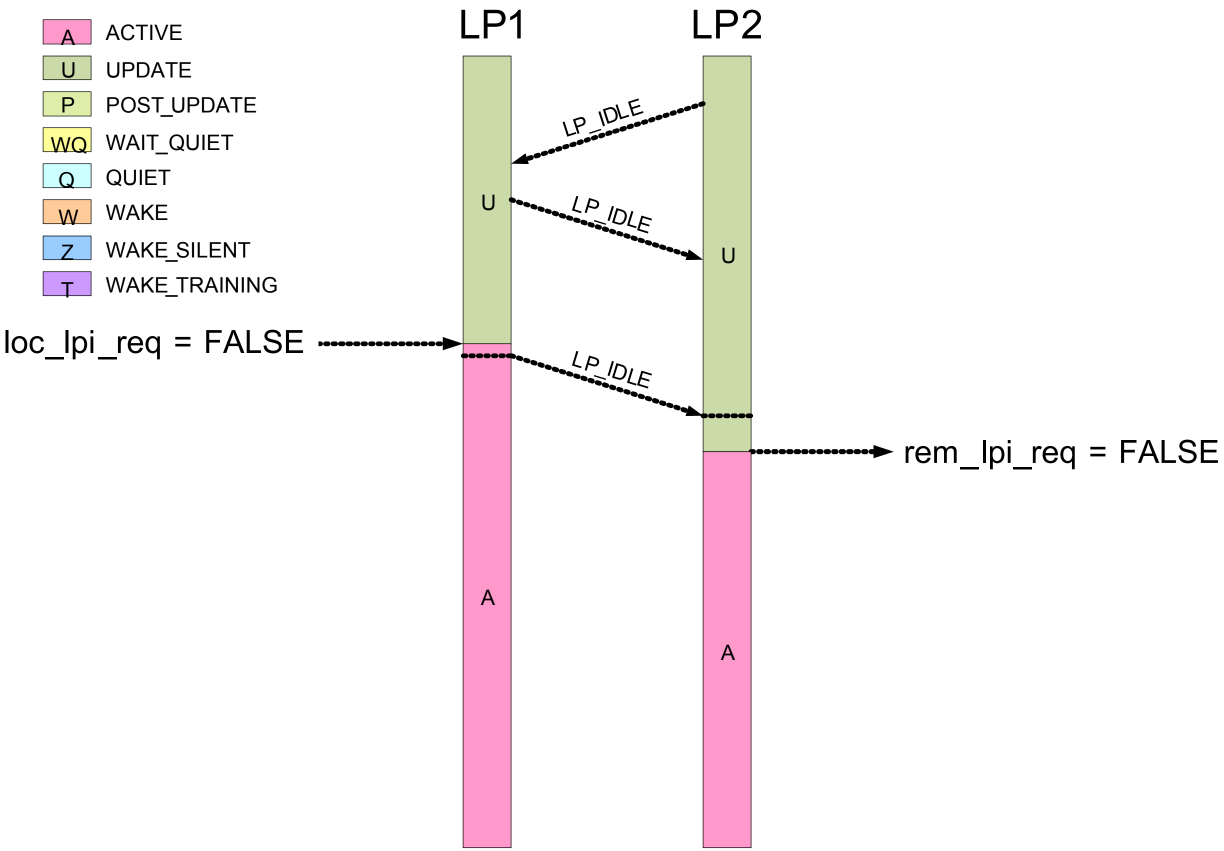
Highlights

- A direct transition is provided from UPDATE (or POST_UPDATE) to SEND IDLE OR DATA if the link partner has not yet completed filter coefficient updates (e.g. rem_upd_done = FALSE)
  - Update of adaptive filter coefficients may continue uninterrupted
- When the remote PHY has signaled completion of update then the transition through to the wake sequence is possible
- Duration of lpi_postupdate_timer is required to be greater than one round-trip delay
  - Propose a range of 2.0 and 2.2 microseconds
- If loc_lpi_req = FALSE during POST_UPDATE, then the local device must wait for rem_upd_done = TRUE before proceeding to WAKE
  - This will not add time to the overall wake time budget
- New signal_detect terminology and lpi_mask_timer included per the preceding definition

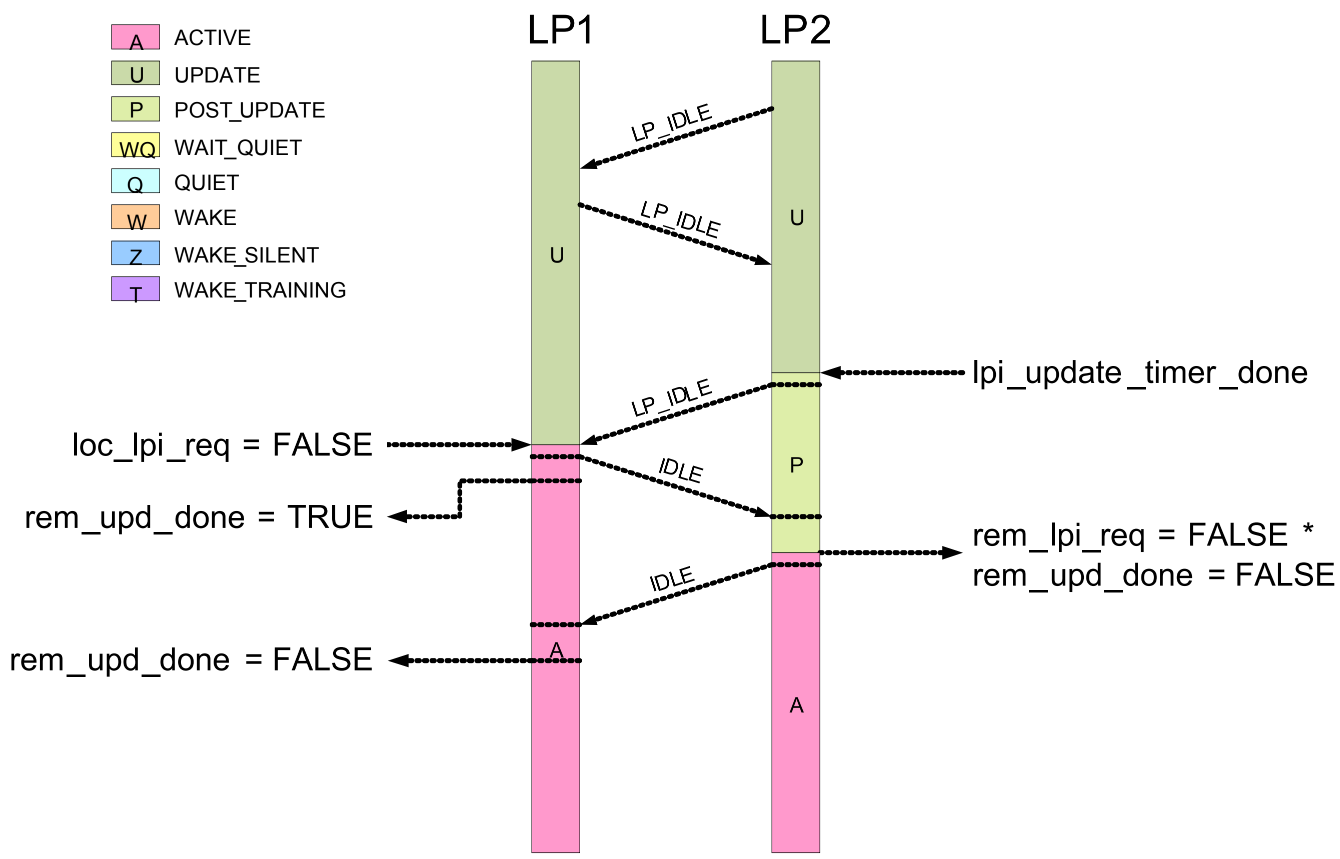
Timing diagram: Enter QUIET



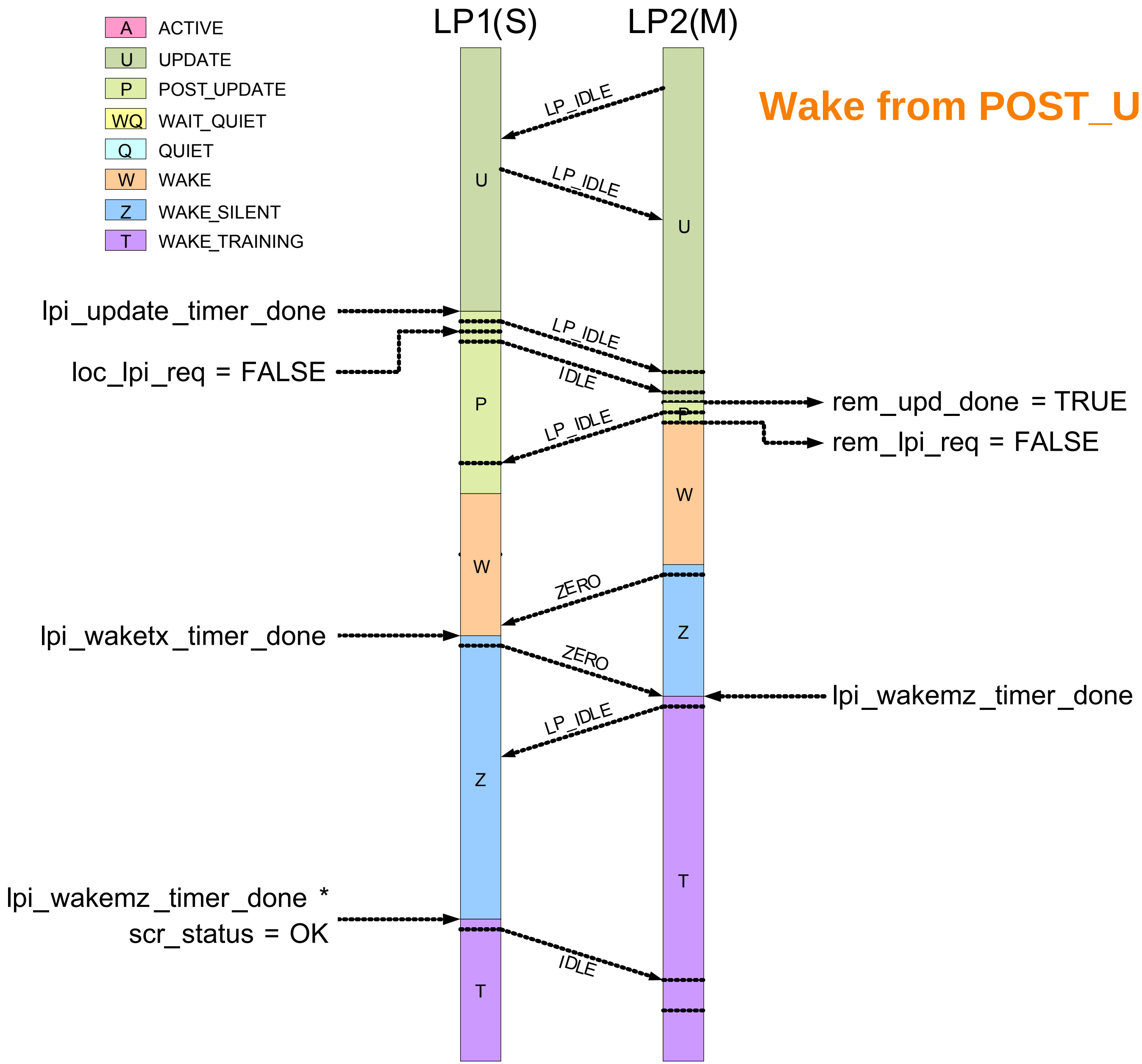
Timing diagram: Wake from UPDATE



Timing diagram: Wake from POST_UDPATE – 1



Wake from POST_UPDATE – 2



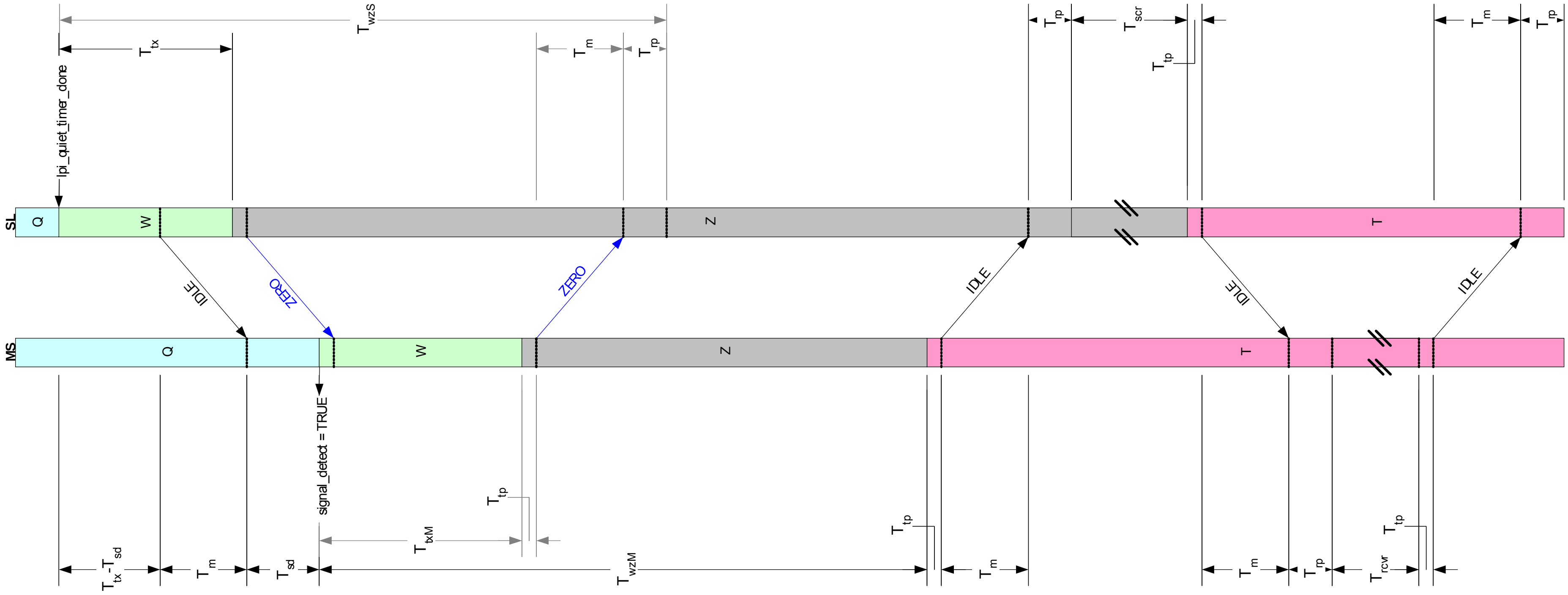
Change to lpi_wakemz_timer

- The value of lpi_wakemz_timer needs to be increased to avoid new corner cases introduced the enforcement of the WAKE_SILENT state during the wake process

Timing analysis: Slave wakes up first

Legend

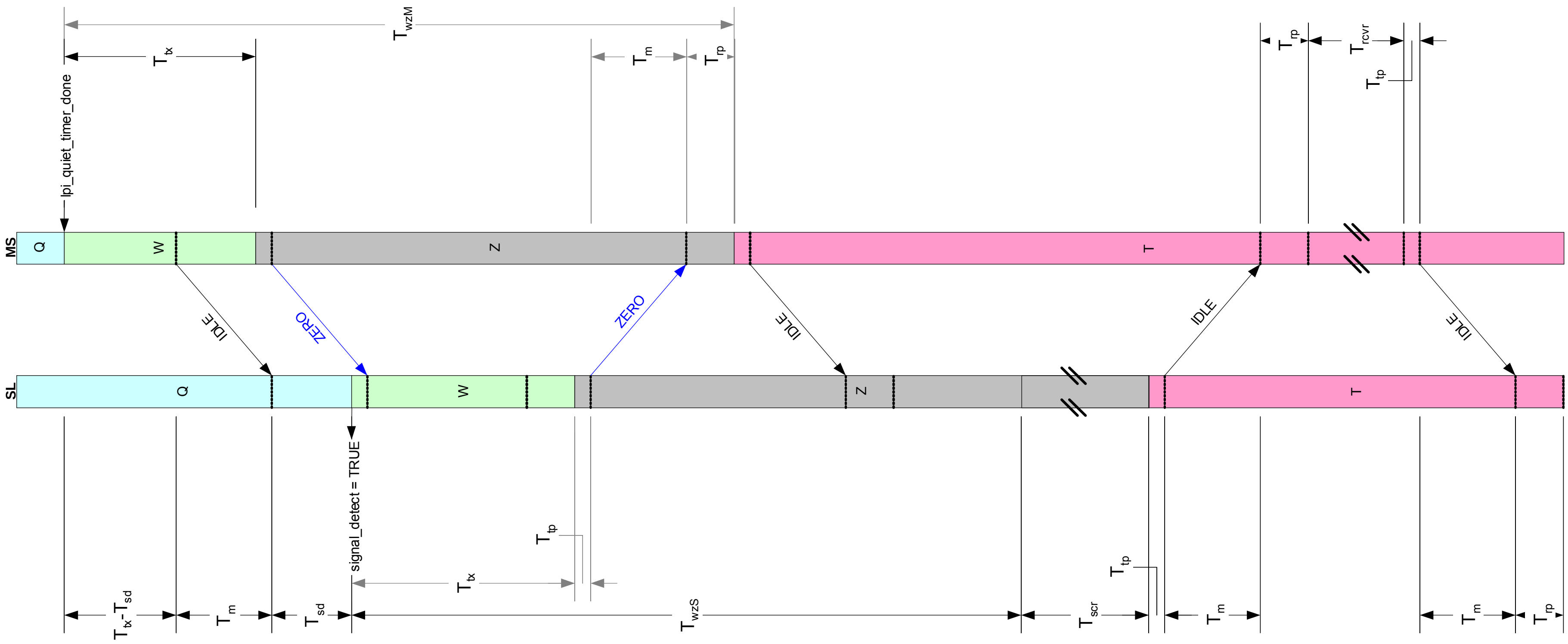
T_{tp}	PHY transmit latency	T_{tx}	lpi_waketx_timer (wake transmit duration)
T_m	Cabel propagation delay	T_{wz}	lpi_wakemz_timer (wakeup stabilization time)
T_{rp}	PHY receive latency	T_{scr}	Scrambler acquisition time
T_{sd}	Signal detect assertion time	T_{rcvr}	Receiver acquisition time



Timing analysis: Master wakes up first

Legend

T_{tp}	PHY transmit latency	T_{tx}	lpi_waketx_timer (wake transmit duration)
T_m	Cabel propagation delay	T_{wz}	lpi_wakemz_timer (wakeup stabilization time)
T_{rp}	PHY receive latency	T_{scr}	Scrambler acquisition time
T_{sd}	Signal detect assertion time	T_{rcvr}	Receiver acquisition time



Timing equations

lpi_wakemz_timer

$$T_{wz} \geq T_{tx} - T_{sd} + T_m + T_{sd} + T_{tx} + T_{tp} + T_m + T_{rp} = T_p + T_m + 2 \cdot T_{tx}^{\max}$$

$$(where \quad T_p = T_{tp} + T_m + T_{rp})$$

Slave wake-up time

$$T_w^s = T_{tx}^s - T_{sd}^m + T_m + T_{sd}^m + T_{wz}^m + T_p + T_{scr} + T_p + T_{rcvr} + T_p = T_{tx}^s + T_m + T_{wz}^m + T_p + T_{scr} + T_p + T_{rcvr} + T_p$$

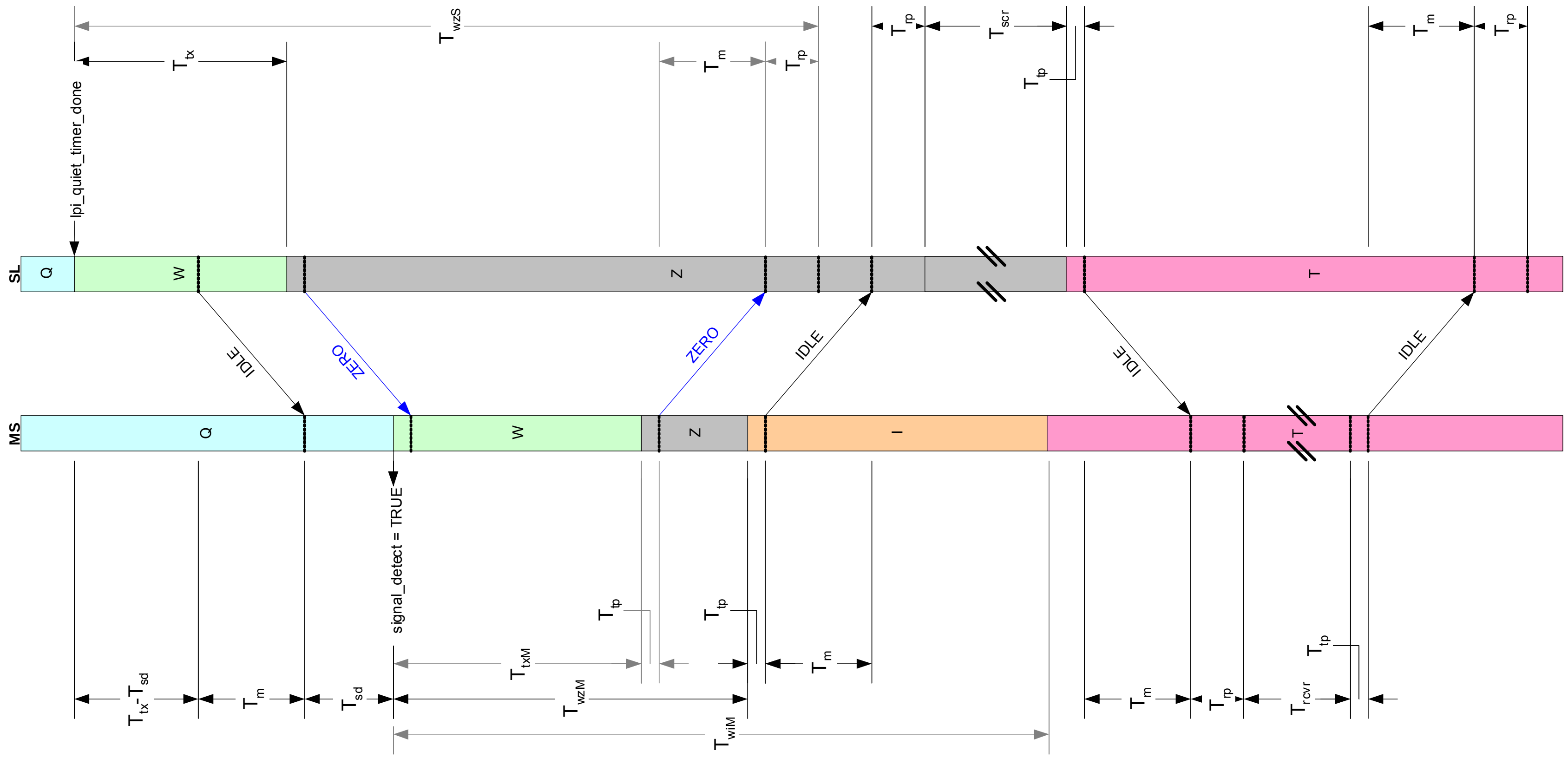
Master wake-up time

$$T_w^m = T_{tx}^m - T_{sd}^s + T_m + T_{sd}^s + T_{wz}^s + T_{scr} + T_p + T_{rcvr} + T_p = T_{tx}^m + T_m + T_{wz}^s + T_{scr} + T_p + T_{rcvr} + T_p$$

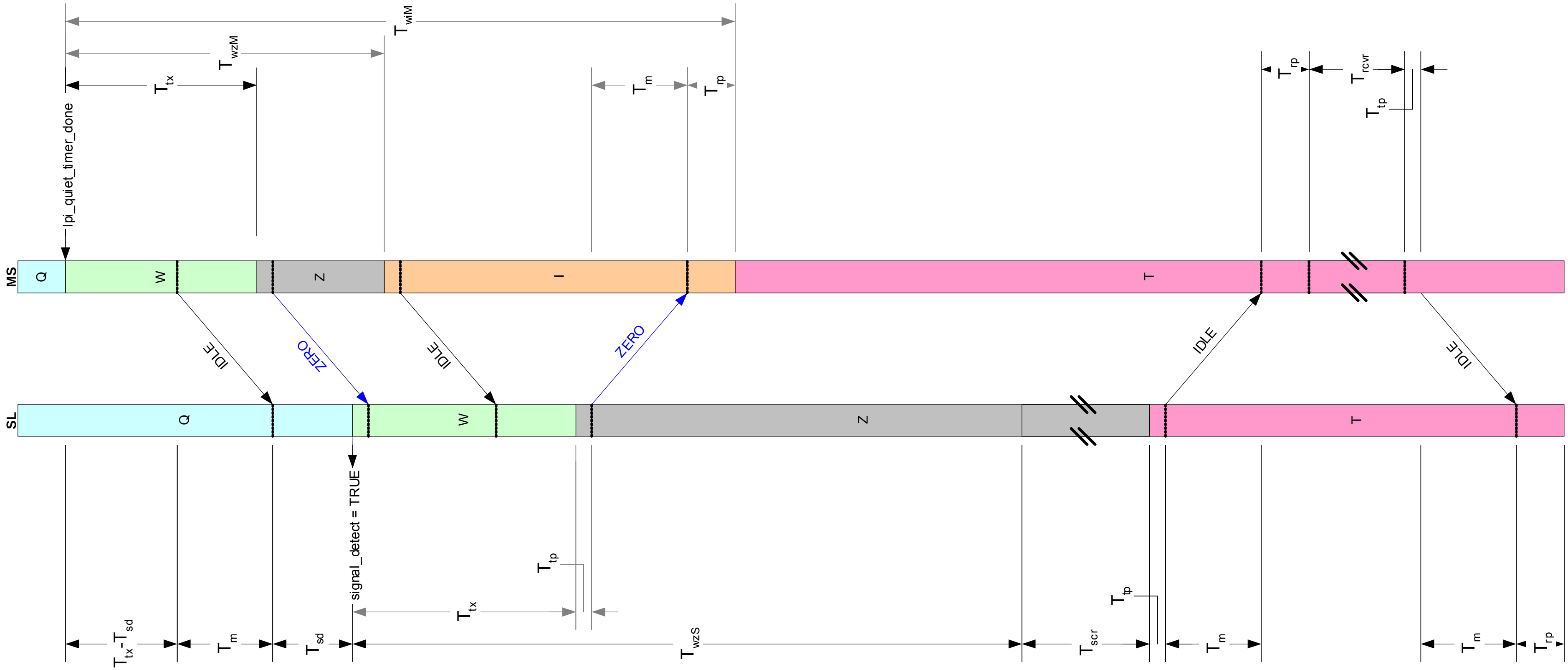
Timing analysis observations

- Assuming master and slave timing parameters are equal, the slave wake-up time is larger (one propagation delay, T_p)
- The slave wake-up time is limited by the master lpi_wakemz_timer
- If that is reduced, the wake-up time could be also reduced

Asymmetric lpi_wakemz_timer: Slave wakes up first



Asymmetric lpi_wakemz_timer: Master wakes up first



Asymmetric lpi_wakemz_timer conclusions

- MASTER lpi_wakemz_timer could be made smaller than the SLAVE
  - Maximum value shown below is 2 microseconds
- "I" state signifies the time the MASTER must just send IDLE symbols to allow the SLAVE to acquire scrambler lock
  - T_{wi} is not a critical parameter
  - It could be added as a new state in the PHY Control state diagram to clarify the intended function
- Sample values (units are bit times)

T_m	550	
T_p	878	
$T_{tx,max}$	1,400	
$T_{wz}(S)$	4,228	
$T_{wz}(M)$	2,000	
T_{scr}	3,000	
T_{rcvr}	1,000	
	Symmetric	**Asymmetric**
$T_w(S)$	12,812	10,584
$T_w(M)$	11,934	11,934

Questions?