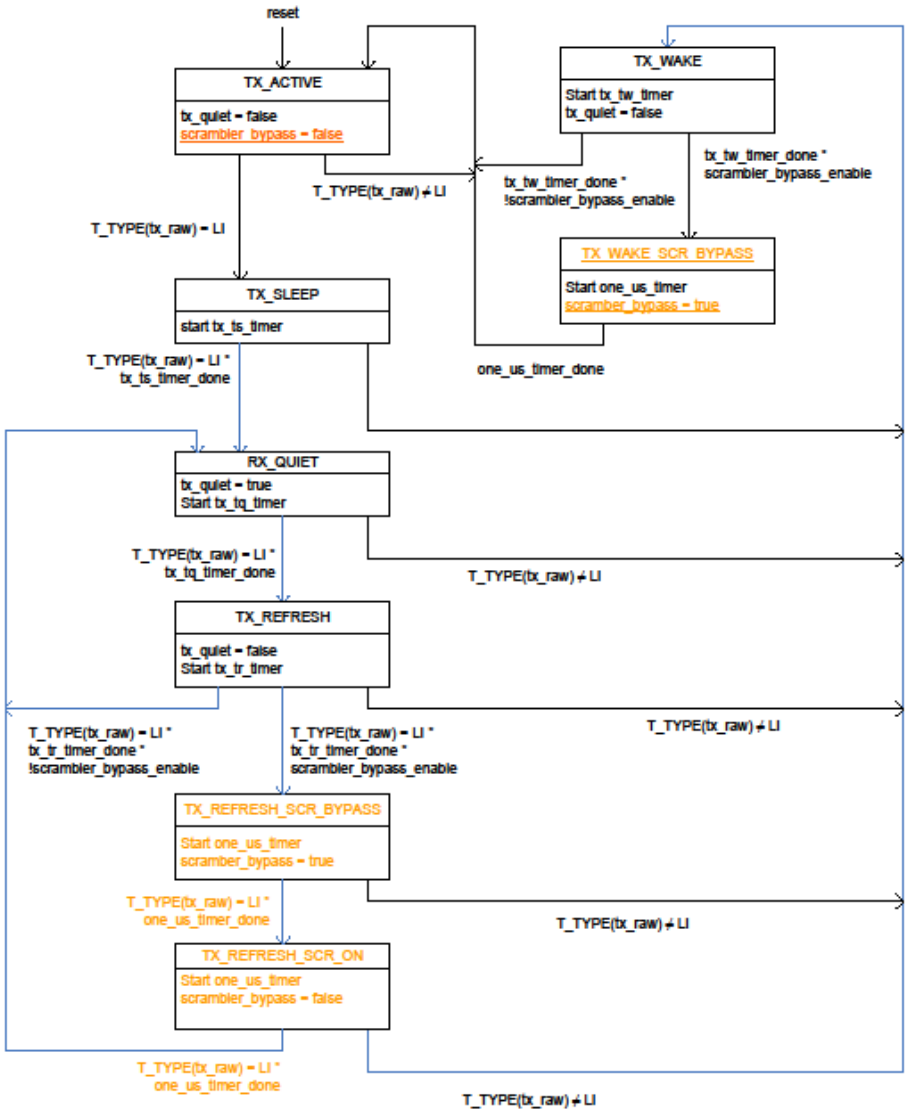
Revised CL49 LPI State diagrams and changes to the PCS scrambler.

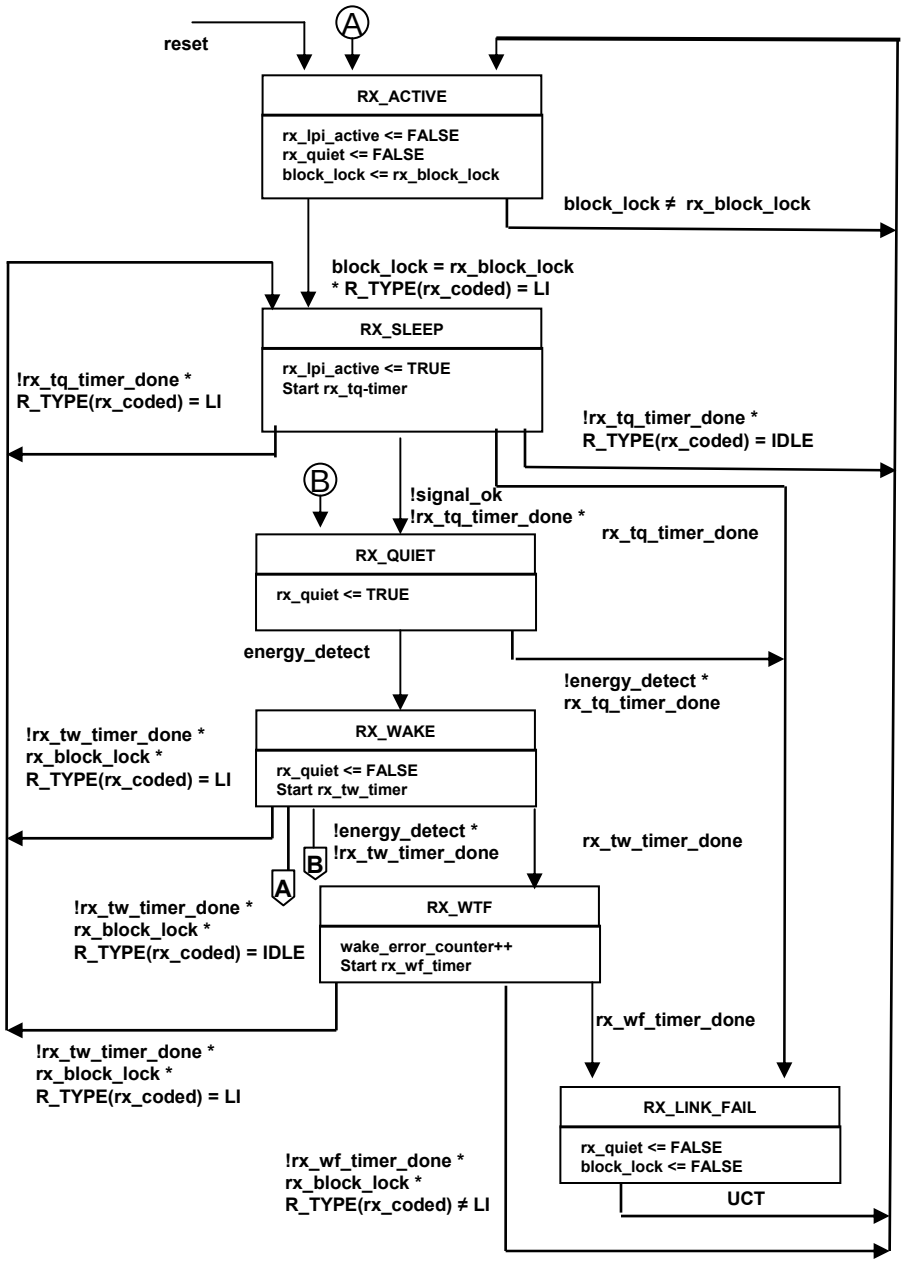
Matt Brown and Velu Pillai

IEEE 802.3az, Chicago, September 2009

Revised CL49 LPI TX State diagram



Revised CL49 LPI RX State diagram



Comment # 223

- Comment 223 is concerned that systematically resetting the PCS scrambler increases vulnerability to malicious packets.
- Fix for this is to bypass the scrambler rather than resetting it.
- The scrambler will continue to operate and the scrambling state will be indeterminate when wake is complete

- Specific changes to CL 49 to support scrambler bypass are provided in accompanying text document.

Cl **49** *SC* **49.2.6** *P* **141** *L* **1** # 223

Gustlin, Mark Cisco

Comment Type **TR** *Comment Status* **D**

It seems to me that resetting the scrambler to all 0s each time the link comes out of LPI is dangerous and will allow malicious users to send killer packets. The original scrambler for 10GE was chose as a very long polynomial to prevent attacks.
Walker's presentation shows a Mean Time to Jamming of 29 years, but that is without resetting the scrambler.
http://grouper.ieee.org/groups/802/3/10G_study/public/jan00/walker_1_0100.pdf

When you reset the scrambler often, that means someone could construct a packet to reverse the scrambler, and if this packet is sent immediately after LPI for instance, it could reverse the scrambler and bring down the link.

SuggestedRemedy

Either find another way to sync up the FEC after LPI or do an analysis that shows the possibility of jamming the scrambling even though it is being reset is not significant.

Proposed Response *Response Status* **W**

PROPOSED ACCEPT IN PRINCIPLE.

Scrambler_reset is no longer needed by the FEC sublayer.

Delete scrambler_reset and all associated specifications.

Proposed edits

September 23, 2009
Matt Brown, Applied Micro
IEEE 802.3az Interim Meeting

Edits to Clause 49 to replace scrambler reset with scrambler bypass.
This is necessary to resolve Comment #223.

In figure 49.4, change scrambler_reset to scrambler_bypass.
In figure 49-5, show bypass mux at output of scrambler.
Currents scrambler diagram

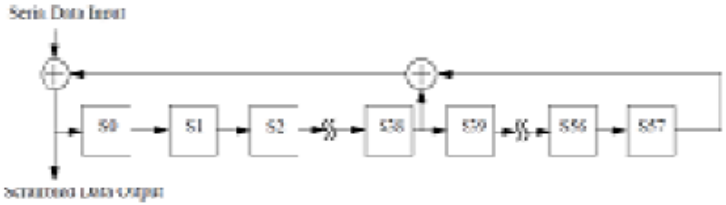


Figure 49–5—Scrambler

Change EEE text after scrambler from...

To aid block synchronization in the receiver when the optional Low Power Idle function is supported, the registers of scrambler shall be held at logic zero while scrambler_reset is TRUE.

To ...

"To aid block synchronization in the receiver when the optional Low Power Idle function is supported, the scrambler input will bypass the scrambler while scrambler_bypass is TRUE."

On page 144, definition changes from...

scrambler_reset
If the optional Low Power Idle function is implemented, this boolean variable is used to bypass the scrambler in order to assist rapid synchronization following Low Power Idle. When set to TRUE, all of the bits of the scrambler delay line are reset. The PHY shall set scrambler_reset_enable = TRUE if FEC is in use.

To... scrambler_bypass
If the optional low power idle function is implemented, this Boolean variable is used to bypass the Tx PCS scrambler in order to assist rapid synchronization following low power idle. When set to TRUE, the PCS will pass the unscrambled data from the scrambler input rather than the scrambled data from the scrambler output. The scrambler will continue to operate normally shifting input data into the delay line. When scrambler_bypass is set to false the PCS will pass scrambled data from the scrambler output.

And change ...

scrambler_reset_enable
A boolean variable used to indicate to the transmit LPI state machine that the scrambler reset option is required.

To... scrambler_bypass_enable
A Boolean variable used to indicated to the transmit LPI state machine that the scrambler reset option is required. This variable is always set to TRUE if the Clause 74 FEC sub-layer is enabled and is otherwise set to FALSE. [The latter sentence addresses comments requesting this clarification.]

In Figure 49-16
Change "scrambler_reset" to "scrambler_bypass" in TX_ACTIVE and SCR_RESET states
Change "scrambler_reset_enable" to "scrambler_bypass_enable" in transitions from TX_WAKE state.
Rename "SCR_RESET" state to "SCR_BYPASS".

Change 3rd sentence in paragraph at bottom of page 149 from ...

Following a period of low power idle, the receiver is required to achieve block synchronization within the wakeup time specified (See Figure 49–17). The implementation of the block synchronization state machine should use techniques to ensure that block lock is achieved with minimal numbers of slip attempts. For PHYs that include the scrambler reset function, the receiver may use the knowledge that the link partner's transmitter has reset the scrambler as part of the wake sequence. The idle sequence following this event will form a fixed pattern for the duration of the wake period.

To...
When the Clause 74 FEC is enabled, the receiver may use the knowledge that the link partner's transmitter will bypass the scrambler as part of the wake sequence.

In Table 49-3...

Table 49–3—Receiver LPI timing parameters

Parameter	Description	Min	Max	Units
T_{QR}	The time the receiver waits for signal detect while in the RX_QUIET state before asserting rx_fault	2	3	ms
T_{WR}	Time to wake remote link partner's receiver. (for PHYs that set scrambler_reset_enable = FALSE)	11	12	µs
T_{WR}	Time to wake remote link partner's receiver. (for PHYs that set scrambler_reset_enable = TRUE)	13	14	µs
T_{WTF}	Wake time fault recovery time	1	1	ms

Change "for PHYs that set scrambler_reset_enable=FALSE" to "when scrambler_bypass_enable = FALSE".

Change "for PHYs that set scrambler_reset_enable=TRUE" to "when scrambler_bypass_enable = TRUE".

Update PICS LP-02, LP-03.

The End.