
Supporting Materials for comments #181, 188, 190, 194, 195, 196, 197, 201, 202

Jeff Slavick

10 July, 2017

Comment #181

<i>CI</i> 080	<i>SC</i> 80.5	<i>P</i> 101	<i>L</i> 4	# <input type="text" value="181"/>
Slavick, Jeff		Broadcom Limited		
<i>Comment Type</i>	T	<i>Comment Status</i>	D	<i><late></i>
Table 80-7 is missing columns for new baud rates we've added.				
<i>Suggested Remedy</i>				
See presentation				
<i>Proposed Response</i>	<i>Response Status</i> W			

[Editor's note: This comment was received after the Working Group ballot closed.]

Comment #181 Resolution

- 100G-CR2 and 100G-DR add two new signaling rates, so we need to update the dynamic skew table with the new rates. No update to static skew (Table 80-6) is needed since it's based on PCS lane rate which hasn't changed.
- Update Table 80-7 for the new signaling rates with the information to the right.

Table 80-7			
		Maximum Skew Variation for 26.5625 Gbaud PMD lane (UI) ^e	Maximum Skew Variation for 53.125 Gbaud PMD lane (UI) ^f
SP0	0.2	n/a	n/a
SP1	0.2	5	n/a
SP2	0.4	11	21
SP3	0.6	16	32
SP4	3.4	90	181
SP5	3.6	96	191
SP6	3.8	101	n/a
SP7	0.2	n/a	n/a
PCS	4	n/a	n/a
FEC Tx	0.4	11	n/a
FEC Rx	4	106	n/a
PCS+FEC	0.4	n/a	n/a
e. The symbol ~ indicates 37.647ps at PMD lane signalling 26.5625Gbd			
f. The symbol ~ indicates 18.8235ps at PMD lane signalling 53.125Gbd			

Comment #188

CI 136 SC 136.9.3.1.3 P 218 L 28 # 188
Slavick, Jeff Broadcom Limited

Comment Type **T** *Comment Status* **D** <late>

136.9.3.1.3 states that when Figure 136-9 enters the OUT_OF_SYNC state the TxEq should be set according to Table 136-12 (Preset1 = NoEq). However, in Figure 136-9 there is no "load" of that Equalization value.

Suggested Remedy

Add a call to "UPDATE_IC" into the OUT_OF_SYNC state before the ENCODE_STS call.

Proposed Response *Response Status* **W**

[Editor's note: This comment was received after the Working Group ballot closed.]

Comment #188 Issue

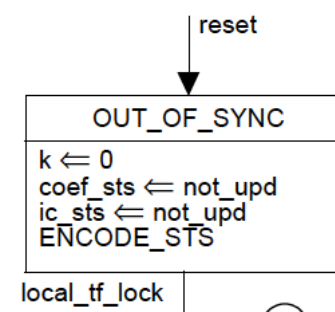
- In Figure 136-9 the OUT_OF_SYNC state just updates the status field, but does not change the local EQ. But 136.9.3.1.3 states the EQ should be set to a particular value.

136.9.3.1.3 Coefficient initialization

When the Coefficient update state diagram (Figure 136-9) is in either the OUT_OF_SYNC state or the NEW_IC state, the coefficients of the transmit equalizer shall be configured to values within the ranges specified in Table 136-12 (according to the value of the variable `ic_req`). These requirements apply upon the assertion of Initial condition status of “Updated”.

Table 136-12—Coefficient initial conditions

Coefficient update state	ic_req	$c(-2)$	$c(-1)$	$c(0)$	$c(1)$
OUT_OF_SYNC	N/A	0 ± 0.025	0 ± 0.05	1 ± 0.05	0 ± 0.05
NEW_IC	preset 1	0 ± 0.025	0 ± 0.05	1 ± 0.05	0 ± 0.05
	preset 2	0 ± 0.025	0 ± 0.05	0.75 ± 0.05	-0.25 ± 0.05
	preset 3	0 ± 0.025	-0.25 ± 0.05	0.75 ± 0.05	0 ± 0.05



Comment #188 Resolution

- Add UPDATE_IC to OUT_OF_SYNC (before ENCODE_STS) into Figure 136-9
- Change 136.8.11.4 to be
 - When a change in the initial condition request bits is detected or a reset occurs, the coefficients are configured in a manner consistent with the following algorithm.

```
if reset or mr_restart_training
  Set coefficients to OUT_OF_SYNC
  ic_sts = not_upd
else
  if ic_req = ind_ctl
    ic_sts = not_upd
  else
    if ic_req = preset 1
      set coefficients to preset 1
    else if ic_req = preset 2
      set coefficients to preset 2
    else if ic_req = preset 3
      set coefficients to preset 3
    end if
  end if
end if
```

Comments #194 & 190

Cl 136 SC 136.8.11.3.3 P 205 L 13 # 194
Slavick, Jeff Broadcom Limited

Comment Type T Comment Status D <late>

The Receiver Frame Lock bit is forced to 0, only during a training startup condition. It should also be forced low when in the TIMEOUT and TRAINING_FAILED states. local_tf_lock is forced low, but that does not force an ENCODE_STS to occur (the function that currently encodes transmitted status bit for receiver frame lock based on local_tf_lock)

SuggestedRemedy

Change the 2nd paragraph of 136.8.11.3.3 to be "Receiver frame lock shall be set to 0 when the variable training is false and it shall not be set to 1 until training and local_tf_lock are both true."
Remove "local_tf_lock is mapped to the receiver frame lock bit," from the definition of ENCODE_STS

Cl 136 SC 136.8.11.7.1 P 208 L 40 # 190
Slavick, Jeff Broadcom Limited

Comment Type T Comment Status D <late>

local_tf_lock should be solely based on the state of Figure 136-8. Another comment is changing the definition of the transmitted Receiver Frame Lock status bit to handle the case when training fails. So we no longer need to force this local status bit low in training failure. While in training failure it's a useful debug status bit to know if you do have training frame lock or not.

SuggestedRemedy

Change the definition of local_tf_lock to be: "Boolean variable that is true when the training frame marker positions have been identified and is false otherwise."

Proposed Response Response Status W

[Editor's note: This comment was received after the Working Group ballot closed.]

Comments #194 Issue

- **136.8.11.3.3 Receiver frame lock**

- When the receiver frame lock bit is set to 1, the receiver is indicating that it has identified training frame marker positions and is in a state where the response time requirements specified in 136.8.11.6 are met. Receiver frame lock shall be set to 0 upon entering TRAINING mode and shall not be set to 1 until `local_tf_lock` is true.

- **ENCODE_STS**

- Encodes portions of the status field of transmitted training frames. `local_tf_lock` is mapped to the receiver frame lock bit, `k` is mapped to the coefficient select echo bits, `coef_sts` is mapped to the coefficient status bits, `ic_sts` is mapped to the initial condition status bit, `local_rx_ready` is mapped to the receiver ready bit, and `tp_mode` is mapped to modulation and precoding bits (see 136.8.11.3.2).

- **local_tf_lock**

- Boolean variable that is true if the value of training is true and training frame marker positions have been identified and is false otherwise.

Comment #194 Issue cont.

- ENCODE_STS is mapping the local_tf_lock variable to Status frame being transmitted. There's some extra requirements placed on the transmitted Status frame in 136.8.11.3.3 to try and ensure it's low when you start-up training and if you enter TIMEOUT or TRAINING_FAILURE (via forcing local_tf_lock low by setting training false).
- However, there is no call to ENCODE_STS in Figure 136-7 to cause the transmitted Status frame to be updated when training becomes false.

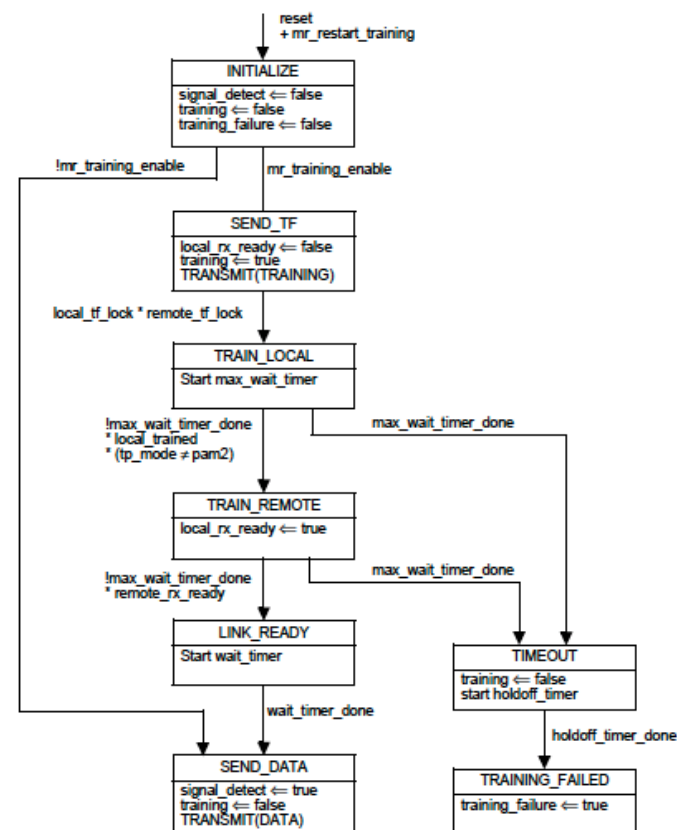


Figure 136-7—PMD control state diagram

Comment #194 resolution

- In 136.8.11.7.2 update ENCODE_STS function as follows
 - ENCODE_STS
 - Encodes portions of the status field of transmitted training frames.
~~local_tf_lock is mapped to the receiver frame lock bit~~, k is mapped to the coefficient select echo bits, coef_sts is mapped to the coefficient status bits, ic_sts is mapped to the initial condition status bit, local_rx_ready is mapped to the receiver ready bit, and tp_mode is mapped to modulation and precoding bits (see 136.8.11.3.2).
- Change 2nd sentence in the definition of
136.8.11.3.3 Receiver frame lock

When the receiver frame lock bit is set to 1, the receiver is indicating that it has identified training frame marker positions and is in a state where the response time requirements specified in 136.8.11.6 are met.

Receiver frame lock shall be set to 0 upon entering TRAINING mode and shall not be set to 1 until local_tf_lock is true.
- To: Receiver frame lock shall be set to 0 when the variable training is false and it shall not be set to 1 until training and local_tf_lock are both true.

Comment #190 resolution

- Change definition of

`local_tf_lock`

Boolean variable that is true if the value of training is true and training frame marker positions have been identified and is false otherwise.

- To: Boolean variable that is if the training frame marker positions have identified and false otherwise.

Comment #194 & 190 resolution effect

- Transmitted receiver frame lock status indicator is based solely on description in 136.8.11.3.3 which will force the bit in the status frame bit to be 0 when Figure 136-7 is in the INITIALIZE, TIMEOUT or TRAINING_FAILURE states, low until local_tf_lock becomes true in the SEND_TF state and is otherwise managed by the local device in TRAIN_LOCAL, TRAIN_REMOTE and SEND_RDY states based upon it's ability to comply with 2ms response time.
- By not forcing local_tf_lock low when in TIMEOUT or TRAINING_FAILURE it allows for frame lock to be maintained locally when one side has timed out, but other is still running training (has yet to timeout). Which provides some debug capability and aligns with how Clause 72 acts.

Comment #195

CI 136 SC 136.8.11.7.2 P 209 L 54 # 195
Slavick, Jeff Broadcom Limited

Comment Type **T** *Comment Status* **D** *<late>*

ENCODE_STS maps local_rx_ready into the transmitted status frame. But it's only called when the remote partner modifies his transmitted frame. So when the local side transitions to TRAIN_REMOTE state, until the partner sends a change in his Control Frame they won't see that you're done. The local side should push it's local_rx_ready status to the partner based on it's state.

SuggestedRemedy

Remove ", local_rx_ready is mapped to the receiver ready bit" from the definition of ENCODE_STS
To help clarify which variable is mapped into the transmitted status bit add "(local_rx_ready = true)" to the end of 2nd sentence of 136.8.11.3.1, and add "(local_rx_ready = false)" to the end of the 3rd sentence.

Proposed Response *Response Status* **W**

[Editor's note: This comment was received after the Working Group ballot closed.]

Comment #195 Issue

- ENCODE_STS

- Encodes portions of the status field of transmitted training frames. local_tf_lock is mapped to the receiver frame lock bit, k is mapped to the coefficient select echo bits, coef_sts is mapped to the coefficient status bits, ic_sts is mapped to the initial condition status bit, local_rx_ready is mapped to the receiver ready bit, and tp_mode is mapped to modulation and precoding bits (see 136.8.11.3.2).

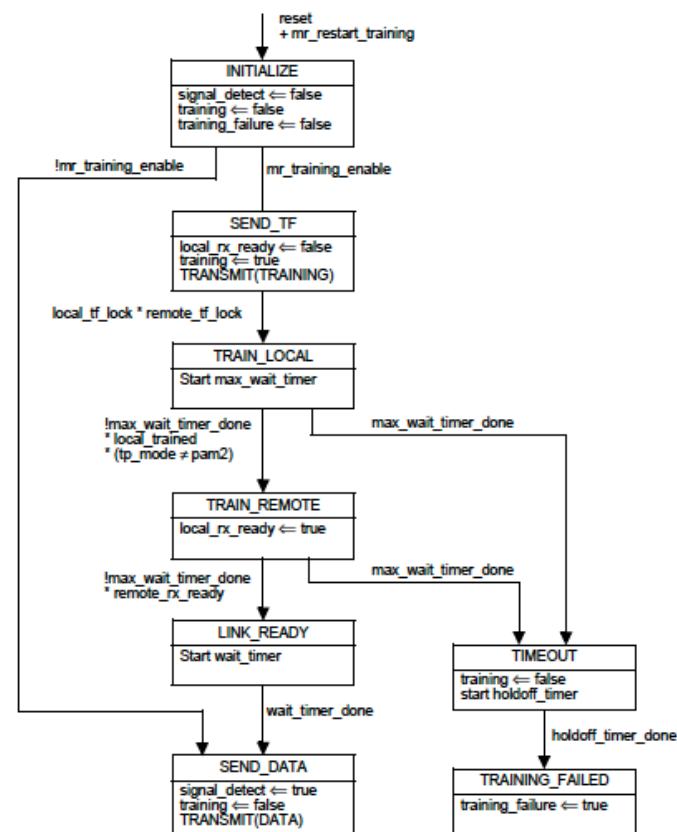


Figure 136-7—PMD control state diagram

- Local_rx_ready is updated in training FSM. But ENCODE_STS is not called by that FSM. So there's no locally initiated process to update the transmitted Status field when local_rx_ready asserts.

Comment #195 Resolution

- In 136.8.11.7.2 update ENCODE_STS function as follows
 - ENCODE_STS
 - Encodes portions of the status field of transmitted training frames. local_tf_lock is mapped to the receiver frame lock bit, k is mapped to the coefficient select echo bits, coef_sts is mapped to the coefficient status bits, ic_sts is mapped to the initial condition status bit, ~~local_rx_ready is mapped to the receiver ready bit,~~ and tp_mode is mapped to modulation and precoding bits (see 136.8.11.3.2).
- Change 136.9.8.11.3.1 to read
 - The receiver ready bit is used to signal the local receiver state to the link partner. When this bit is 1, it indicates that the local receiver has completed training and is prepared to receive data (**local_rx_ready is true**). When this bit is 0, it indicates that the local receiver is requesting that training continue (**local_rx_ready is false**).

Comment #196

CI 136 **SC 136.8.11.7.2** **P 210** **L 1** # **196**
Slavick, Jeff Broadcom Limited

Comment Type **T** *Comment Status* **D** <late>

tp_mode is based on received frames. ENCODE_STS updates the status field of the transmitted frames, which should not be based on tp_mode, but based on our local logic acting upon the received tp_mode_req from the far end.

Suggested Remedy

Remove ", and tp_mode is mapped to modulation and precoding bits (see 136.8.11.3.2)" from the definition of ENCODE_STS

The current definition in 136.8.11.3.2 sufficiently defines how the local status bits are set.

Proposed Response *Response Status* **W**

[Editor's note: This comment was received after the Working Group ballot closed.]

Comment #196 Issue

- ENCODE_STS

- Encodes portions of the status field of transmitted training frames. local_tf_lock is mapped to the receiver frame lock bit, k is mapped to the coefficient select echo bits, coef_sts is mapped to the coefficient status bits, ic_sts is mapped to the initial condition status bit, local_rx_ready is mapped to the receiver ready bit, and tp_mode is mapped to modulation and precoding bits (see 136.8.11.3.2).

tp_mode

Enumerated variable that corresponds to the “modulation and precoding status” bits in the status field of received training frames. It may be assigned one of the following values: pam2, pam4, pam4 with precoding.

- Tp_mode is based on the status field we **receive**, which is status of the remote Transmitter. The Status field we send must be based on the state of the local transmitter.
- ENCODE_STS updates the Status field we send but is not called when a change in modulation and precoding request is received.

Comment #196 Resolution

- In 136.8.11.7.2 update ENCODE_STS function as follows
 - ENCODE_STS
 - Encodes portions of the status field of transmitted training frames. local_tf_lock is mapped to the receiver frame lock bit, k is mapped to the coefficient select echo bits, coef_sts is mapped to the coefficient status bits, ic_sts is mapped to the initial condition status bit, local_rx_ready is mapped to the receiver ready bit, ~~and tp_mode is mapped to modulation and precoding bits (see 136.8.11.3.2).~~
- No change is need in 136.8.11.3.2, current text covers how to set the status fields.

136.8.11.3.2 Modulation and precoding status

The modulation and precoding status bits acknowledge the modulation and precoding request bits received from the link partner. When a change in the received modulation and precoding request bits are detected, the format of the training pattern is changed accordingly (see 136.8.11.1.3). The modulation and precoding status bits are then updated to confirm that the change to the format of the training pattern was completed.

Comment #197

CI 136 SC 136.8.11.7.5 P 213 L 4 # 197
Slavick, Jeff Broadcom Limited

Comment Type **T** *Comment Status* **D** *<late>*

The training FSM maybe run through a reset sequence without the PMD being reset. When that happens, the Frame_lock and Coeff_Update FSM blocks should also be placed into their Reset states.

SuggestedRemedy

Add "+ mr_restart_training" to the entry condition of OUT_OF_FRAME in Figure 136-8 and to the entry condition of "OUT_OF_SYNC" in Figure 136-9

Proposed Response *Response Status* **W**

[Editor's note: This comment was received after the Working Group ballot closed.]

Comment #197 Issue

- In Figures 136-7,8,9 the condition to enter the “RESET” state is not consistent. When you reset the training process only “mr_restart_training = true” all FSM related to training should be placed into a reset state.

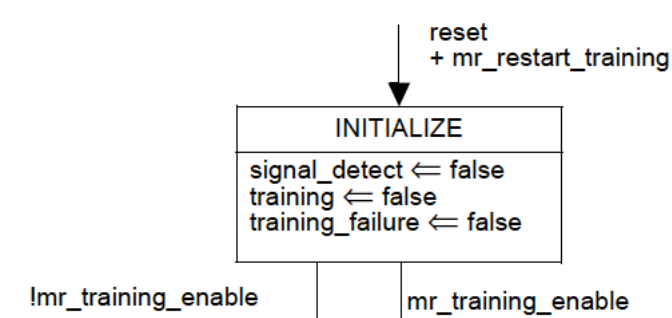


Figure 136-7

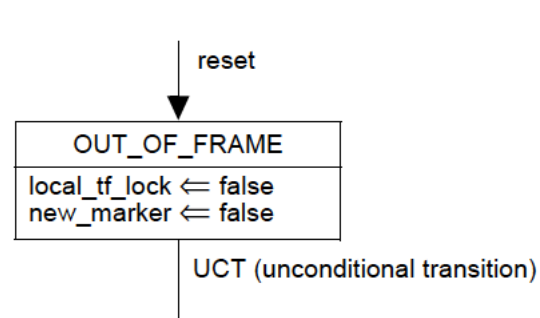


Figure 136-8

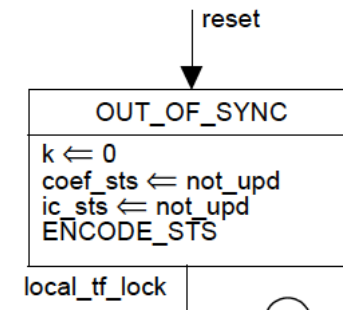


Figure 136-9

Comment #197 Resolution

- To align all 3 FSM reset states
 - Add “+ mr_restart_training” to the entry condition to OUT_OF_FRAME in Figure 136-8
 - Add “+ mr_restart_training” to the entry condition to OUT_OF_SYNC in Figure 136-9

Comments #201 & 202

CI 136 SC 136.8.11.5 P 206 L 9 # 201
Slavick, Jeff Broadcom Limited

Comment Type T Comment Status D <late>

Update to the coefficients is only done upon a change in request bits, not a change in select.

SuggestedRemedy

Remove "coefficient select or" from the first sentence of 136.8.11.5

CI 136 SC 136.8.11.5 P 206 L 9 # 202
Slavick, Jeff Broadcom Limited

Comment Type T Comment Status D <late>

The algorithm in this section does not set `coef_sts = not_upd`, that is done in Figure 136-9. But when you change the `coef_req` from INC/DEC/NOEQ to HOLD that is a change in the coefficient request bits. So now you have the FSM (Figure 136-9) and this algorithm fighting each other. This algorithm would set `coef_sts = UPDATED` for HOLD (but not change the setting) while the FSM sets the response properly to NOT_UPDATED

SuggestedRemedy

Change "When a change to the coefficient select or coefficient request bits is detected" to "When a request to update a coefficient setting is detected in the coefficient request bits" to limit the scope of this code to execute only when you go from HOLD -> INC/DEC/NOEQ

Comment #201 & 202 Issue

- 136.8.11.5 reads as below, and currently reacts to both a select and request change (any change).

When a change to the coefficient select or coefficient request bits is detected, the corresponding coefficient update is performed in a manner consistent with the following algorithm.

```
if k in k_list
    if coef_req = INCREMENT
        ck_ask = c(k) + ck_stp
    else if coef_req = DECREMENT
        ck_ask = c(k) - ck_stp
    else if coef_req = NO EQUALIZATION
        if k = 0
            ck_ask = 1
        else
            ck_ask = 0
        end if
    else
        ck_ask = c(k)
    end if

    if ck_ask > ck_max
        c(k) = ck_max
        if CHECK_EQ(ck_ask,k)
            coef_sts = COEFFICIENT AT LIMIT AND EQUALIZATION LIMIT
        else
            coef_sts = COEFFICIENT AT LIMIT
        end if
    else if ck_ask < ck_min
        c(k) = ck_min
        if CHECK_EQ(ck_ask,k)
            coef_sts = COEFFICIENT AT LIMIT AND EQUALIZATION LIMIT
        else
            coef_sts = COEFFICIENT AT LIMIT
        end if
    else if CHECK_EQ(ck_ask,k)
        coef_sts = EQUALIZATION LIMIT
    else
        c(k) = ck_ask
        coef_sts = UPDATED
    end if
else
    coef_sts = COEFFICIENT NOT SUPPORTED
end if
```

Comment #201 & 202 Issue cont.

- On coef_req change to HOLD, 136.8.11.5 algorithm stays coef_sts = UP while Figure 136-9 says coef_sts = NUP (WAIT state).
- On a coef_sel change, coef_sts would become UP per 136.8.11.5 while 136-9 says it's NUP (NEW_INDEX state). Changing the “selector” field should only modify the coefficient select echo field and not coef_sts field.

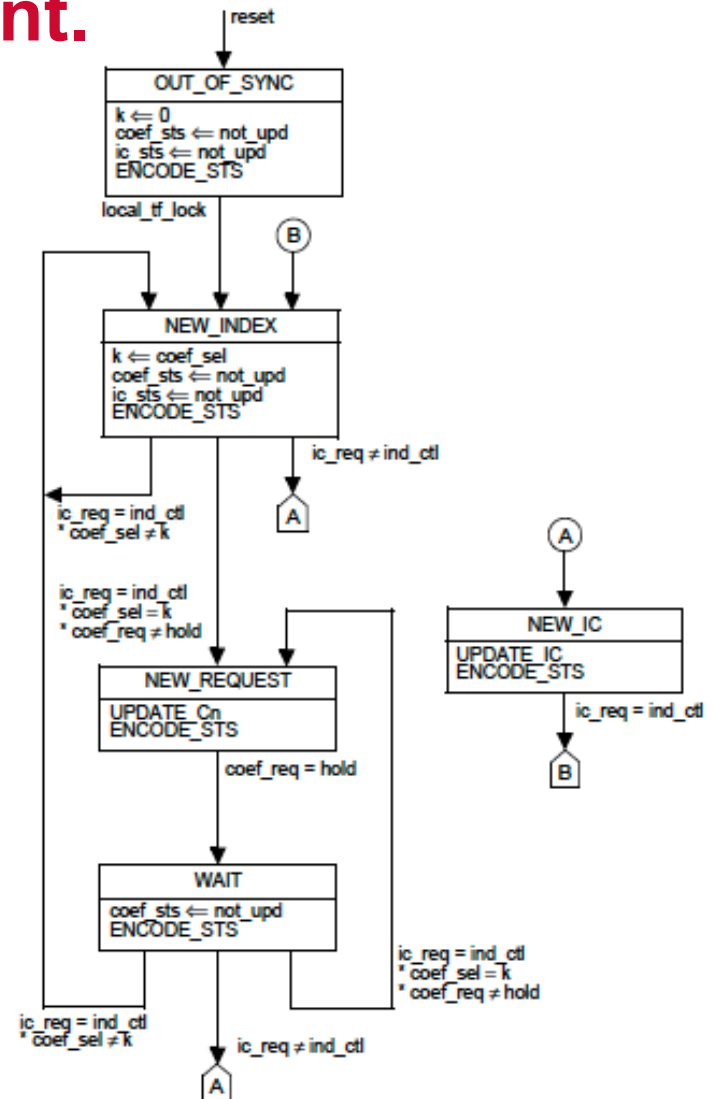


Figure 136-9—Coefficient update state diagram

Comment #201 & 202 Resolution

- The algorithm in 136.8.11.5 should only run when UPDATE_Cn is called in Figure 136-9. This occurs when coef_req is changed from HOLD -> INC/DEC/NOEQ. Changing coef_req back to HOLD and changes to coef_sel do not cause UPDATE_Cn to be executed.
- **Change 136.8.11.5 to read**
 - When a request to update a coefficient setting is detected in the coefficient request bits, the corresponding coefficient update is performed in a manner consistent with the following algorithm.