Synchronization State Machine Modification

IEEE 802.3bp - Plenary Meeting - March 2015

William Lo, Marvell

MARVELL®

Objectives

▸ **Allow link up between forced 1000BASE-T1 PHY and Auto-Negotiation Enabled PHY**

▸ **Some house keeping for forced 1000BASE-T1 mode**

▸ **Consistent transition into training regardless of path taken**

▸ **Keep exchange of SEND_S half duplex**

House keeping

▶ **If Auto-Negotiation is not implemented need registers to**

- **Manually set master / slave**
- **Manually tell the PHY that 1000BASE-T1 is selected**

▶ **Add following registers to control**

Bit(s)	Name	Description	R/W
1.2304.4	Master/Slave	1 = Master 0 = Slave	R/W
1.2304.3:0	PHY Type	BASE-T1 PHY speed when Auto-negotiation is disabled 0000 = Reserved 0001 = Reserved 0010 = 1000BASE-T1 Else = Reserved	R/W

▶ **Registers ignored if Auto-Negotiation is enabled**

▶ **See Lo_McClellan_3bp_01a_0315.pdf page 1 for detail text**

- **1.2304.4 is force_config variable**
- **1.2304.3:0 is force_phy_type variable**

Current Synchronization State Machine

▸ **Forced slave is always silent**
  ▪ **No way for link partner to know the slave PHY is there.**
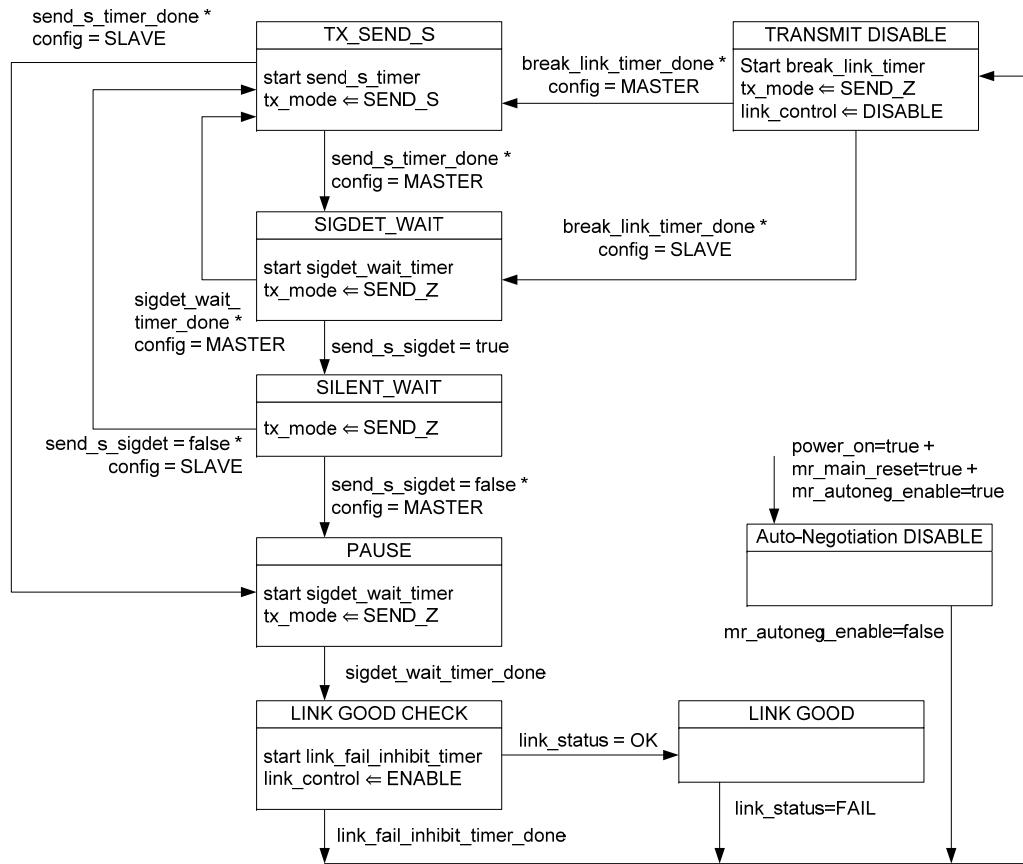
Proposed Updates

▶ **Slave is not silent all the time (LONG_SEND_S)**

▶ **Orderly path to allow transition from Auto-Neg to Sync State Machine (Blue states)**
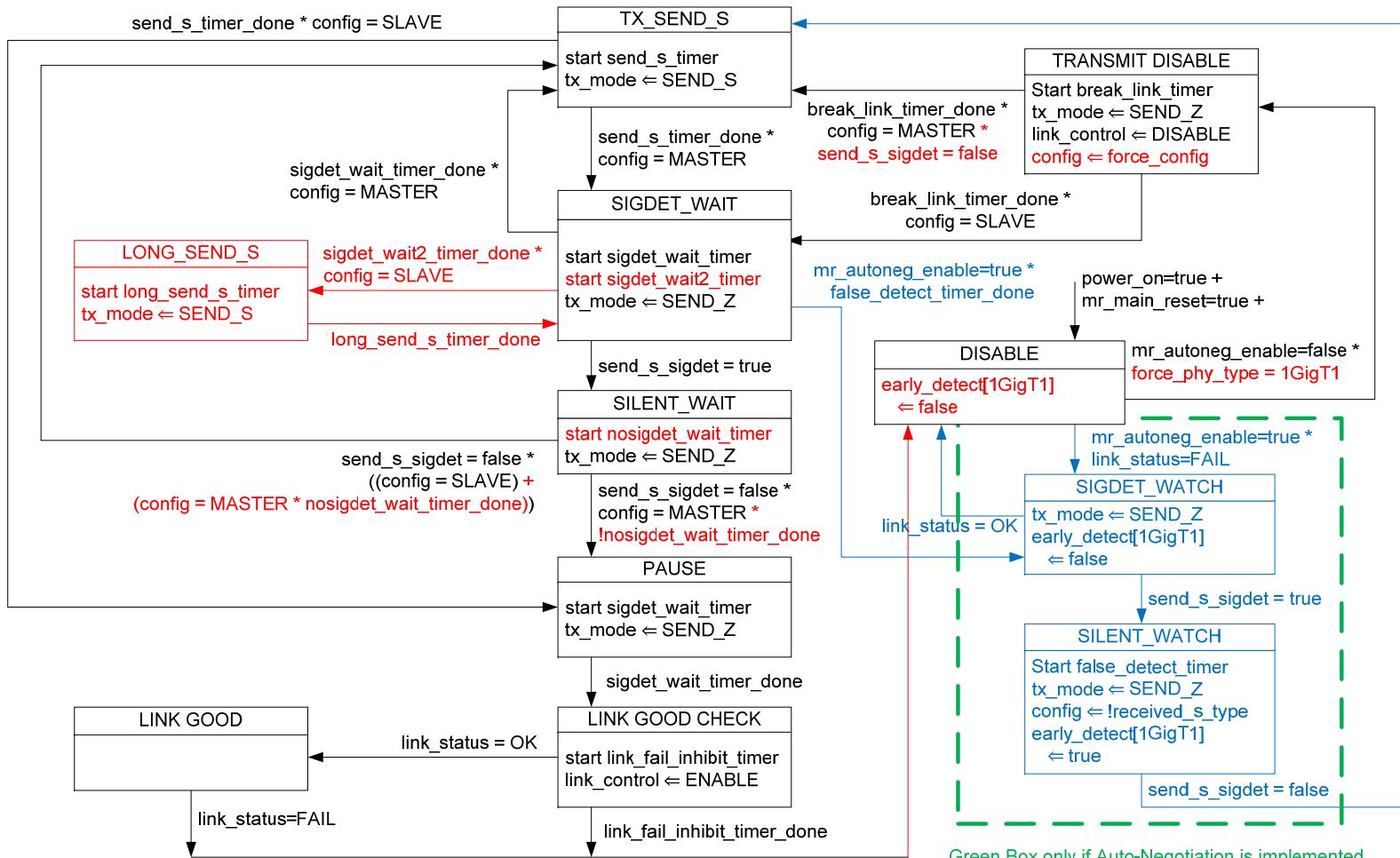
Timers for synchronization

▶ **Preserves half duplex nature of SEND_S**

▶ **Slave sends longer duration SEND_S only for the purpose to wake Auto-Negotiation enabled link partner**

▶ **Synchronize only on normal duration SEND_S (both master and slave)**

Timer	Duration	Description
send_s_timer	1.0us +/- 0.04us	Normal SEND_S duration
sigdet_wait_timer	4.0us +/- 0.04us	Time between end of one SEND_S and beginning of next SEND_S from master if slave does not respond
long_send_s_timer	3.0us +/- 0.04us	Long SEND_S duration from slave used to wakeup link partner
sigdet_wait2_timer	10.0us +/- 0.04us	Time between end of one SEND_S and beginning of next SEND_S from slave if master does not respond
nosigdet_wait_timer	1.7us +/- 0.04us	Timer for master to test whether normal or long duration SEND_S is sent by slave
false_detect_timer	10.0us +/- 0.04us	Exit timer if false detection triggered.

MARVELL®

Scenario: Force one side, Auto-Negotiation on other side

▸ **SIGDET_WATCH: Watch for SEND_S**

▸ **SILENT_WATCH: Watch for SEND_S to end**

▸ **TX_SEND_S: Transmit SEND_S and continue as normal**

▸ **Consistent start up – Slave always transmits final SEND_S**
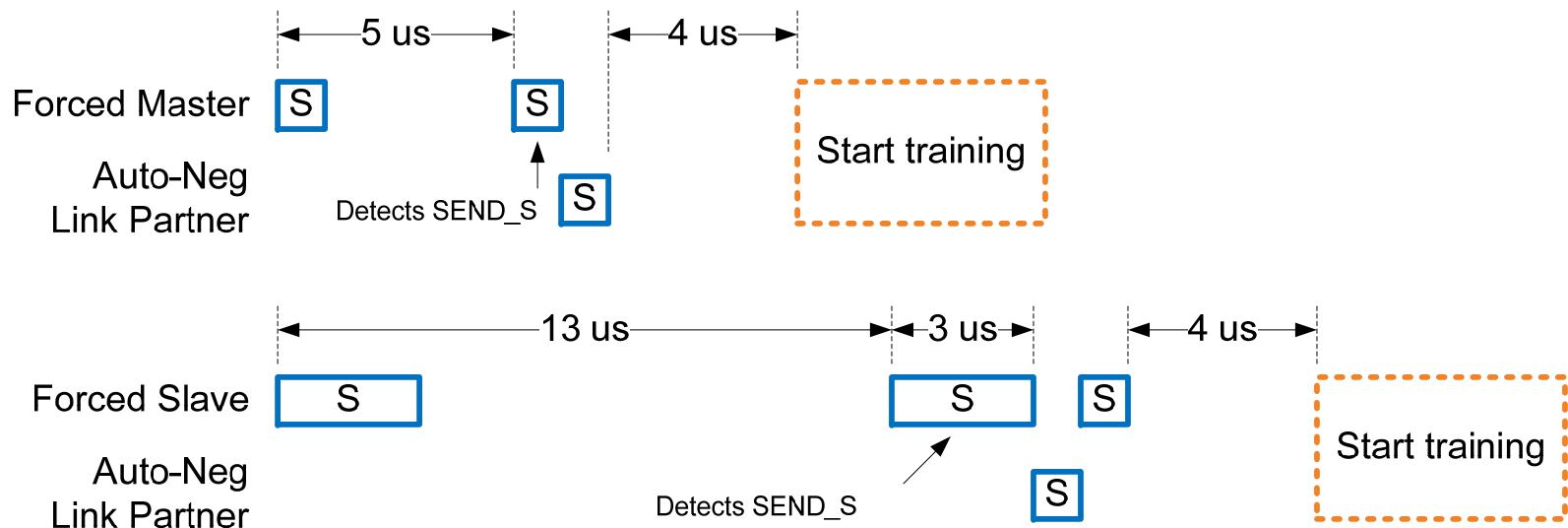
Scenario: Both sides forced, no SEND_S collision

▶ **Slave sees master's SEND_S first and responds with its own SEND_S**

▶ **Long duration SEND_S does not interfere**

▶ **Slave always the last to transmit SEND_S**
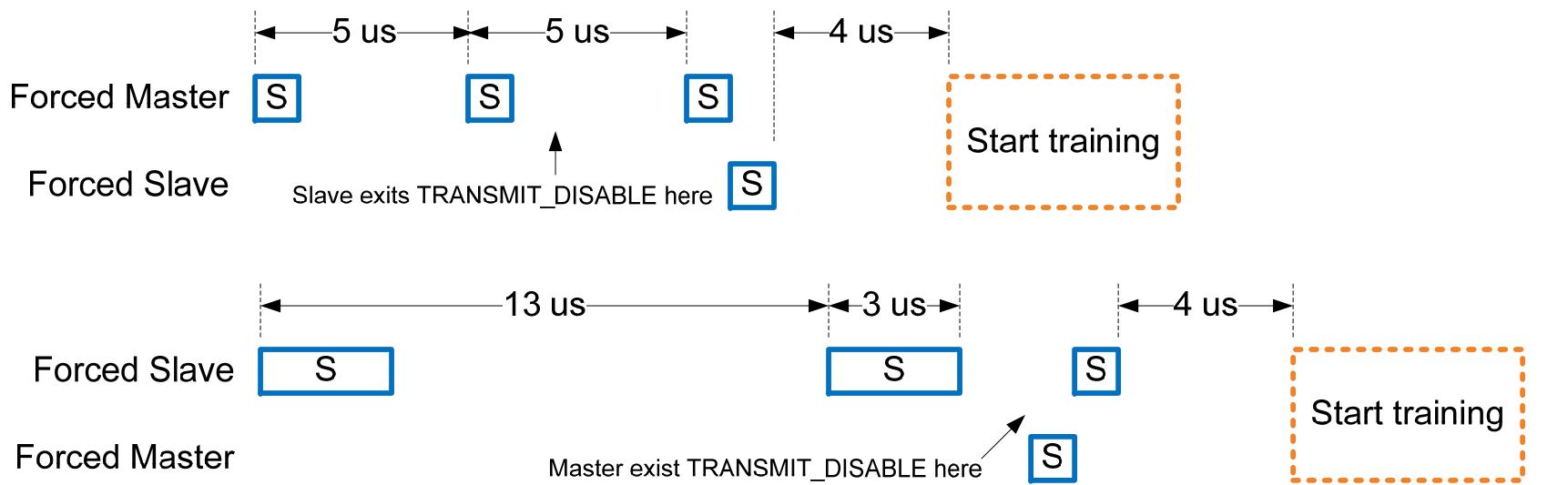
Scenario: Both sides forced, SEND_S collision

▶ **Both side initiates SEND_S simultaneously**

▶ **Master Path:**

- **TX_SEND_S – Send normal duration SEND_S**
- **SIGDET_WAIT – sees slave's SEND_S**
- **SILENT_WAIT – nosigdet_wait_timer expires**
- **TX_SEND_S – long duration SEND_S from slave doesn't count**

▶ **Slave Path:**

- **Long_SEND_S – Send long duration SEND_S**

- **SIGDET_WAIT - waits for master's SEND_S**
- **SILENT_WAIT – waits for master to be quiet**
- **TX_SEND_S – Send normal duration SEND_S**

←4 us→

Forced Slave   [S]   [S]

Start training

Forced Master   [S]   [S]

nosig_wait_timer expired

Forced Slave   [S]

Not possible. Master does not exit TRANSMIT_DISABLE state until send_s_sigdet = false

Forced Master   [S]
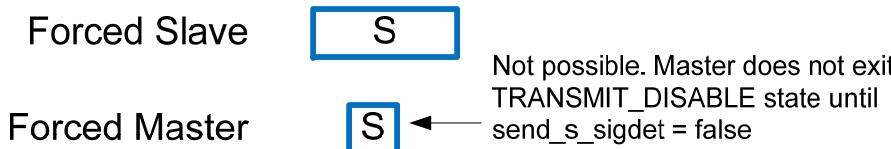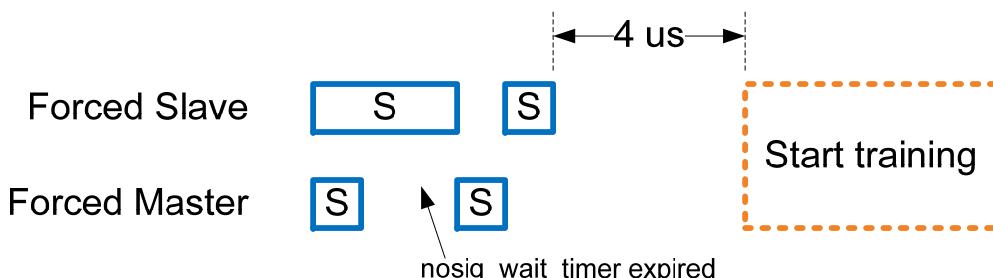
▶ **Collisions always start simultaneously**

▶ **Cannot transmit SEND_S unless there is no signal detect**

9

MARVELL®

Other Scenarios

▶ **False detection:**

- **SIGDET_WATCH – false detection**

- **SILENT_WATCH - starts false_detect_timer**

- **TX_SEND_S – Send normal duration SEND_S**

- **SIGDET_WAIT – no resolution,**
  false_detect_timer expires

- **SIGDET_WATCH – Auto-Neg re-eanbled**

Auto-Neg   False detection | S | ←——— 10 us ———→   Resume Auto-Neg if sync not established.
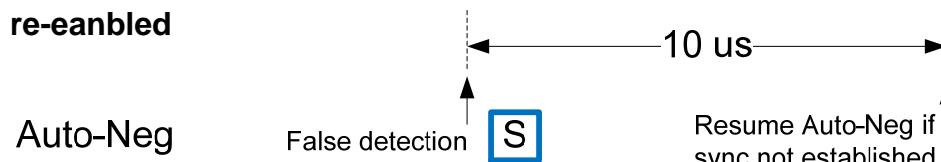
▶ **Auto-Negotiation on both sides:**

- **SIGDET_WATCH – watching link partner**

- **DISABLE - enters here when link is up via Auto-Negotatiation**

  Can shut down SEND_S detection circuit

MARVELL®

Auto-Negotiation Arbitration State Machine

▶ **early_detect[1GigT1] = true**

- **Disengages Auto-Negotiation**
- **Synchronization state machine fully controls synchronization**

ABILITY DETECT

transmit_ability ⇐ true
mr_lp_autoneg_able ⇐ false
toggle_tx ⇐ mr_adv_ability[12]
ability_match ⇐ false
acknowledge_match ⇐ false
tx_link_code_word[48:1] ⇐ mr_adv_ability[48:1]
mr_page_rx ⇐ false
base_page ⇐ true
ack_finished ⇐ false
consistency_match ⇐false

TRANSMIT DISABLE

Start break_link_timer
link_control_[all] ⇐ DISABLE
transmit_disable ⇐ true
mr_page_rx ⇐ false
mr_autoneg_complete ⇐ false
mr_next_page_loaded ⇐ false

break_link_timer_done

ability_match=true ∗
nonce_match=true

(acknowledge_match=true ∗
(consistency_match=false +
(ack_nonce_match=false ∗
base_page=true))) +
an_receive_idle=true

ability_match=true ∗ nonce_match=false

ACKNOWLEDGE DETECT

IF(base_page=true) THEN
tx_link_code_word[10:6] ⇐ rx_nonce[4:0]
transmit_ability ⇐ true
transmit_ack ⇐ true
mr_lp_autoneg_able ⇐ true
link_control_[all] ⇐ DISABLE

acknowledge_match=true ∗
(ack_nonce_match=true +
base_page=false) ∗
consistency_match=true

power_on=true +
mr_main_reset=true +
mr_restart_negotiation=true +
mr_autoneg_enable=false +
early_detect[1GigT1]=true

COMPLETE ACKNOWLEDGE

complete_ack ⇐ true
transmit_ability ⇐ true
transmit_ack ⇐ true
toggle_rx ⇐ rx_link_code_word[12]
toggle_tx ⇐ !toggle_tx
mr_page_rx ⇐ true
np_rx ⇐ rx_link_code_word[NP]
mr_lp_adv_abiliy ⇐ rx_link_code_word[48:1]

ability_match=true ∗
((toggle_rx ^ ability_match_word[12])=1)

Auto-Negotiation ENABLE

mr_page_rx ⇐ false
mr_autoneg_complete ⇐ false
transmit_disable ⇐ true

mr_autoneg_enable=true

ack_finished=true ∗
tx_link_code_word[NP]=0 ∗
np_rx=0

ack_finished=true ∗
mr_next_page_loaded=true ∗
((tx_link_code_word[NP]=1) + (np_rx=1))

AN GOOD

an_link_good ⇐ true
mr_autoneg_complete ⇐ true

link_status_[HCD]=OK

AN GOOD CHECK

link_control_[notHCD] ⇐ DISABLE
link_control_[HCD] ⇐ ENABLE
an_link_good ⇐true
start link_fail_inhibit_timer

((link_status_[HCD]=FAIL +
link_status_[HCD]=OK) ∗
link_fail_inhibit_timer_done) +
incompatible_link = true

NEXT PAGE WAIT

transmit_ability ⇐ true
mr_page_rx ⇐ false
base_page ⇐ false
tx_link_code_word[48:13] ⇐ mr_np_tx[48:13]
tx_link_code_word[12] ⇐ toggle_tx
tx_link_code_word[11:1] ⇐ mr_np_tx[11:1]
ack_finished ⇐ false
mr_next_page_loaded ⇐ false

link_status_[HCD]=FAIL

an_receive_idle=true

11

MARVELL®

New variables 97.6.1.1 and timers 97.6.1.2

Variables

force_config   Master/slave configuration when Auto-Negotiation is disabled
                      Values   MASTER
                                    SLAVE
force_phy_type   BASE-T1 Type configuration when Auto-Negotiation is disabled
                      Values   1GigT1: Select 1000BASE-T1 type
early_detect[1GigT1] See 98.5.1
received_s_type   SEND_S type of link partner. If received_s_type is MASTER then !received_s_type is SLAVE
                      and vice versa.
                      Values   MASTER
                                    SLAVE

Timers

long_send_s_timer   Timer used to control the long duration SEND_S is transmitted by slave
                      The timer shall expire 3.0us ±0.04 us after being started.
signal_wait2_timer   Timer used by the slave to control the wait time between transmitting long duration SEND_S.
                      The timer shall expire 10.0us ±0.04 us after being started.
nosigdet_wait_s_timer Timer used to by the master to test whether the SEND_S from the slave is normal or long
                      duration.
                      The timer shall expire 1.7us ±0.04 us after being started.
false_detect_timer   Exit timer if false SEND_S.detection is triggered
                      The timer shall expire 10.0us ±0.04 us after being started.

New variables 98.5.1

Variables

early_detect[x]   Early detection that link can operate in x type.

                                  Values   True: detection that link can operate in x type.

                                                   False: no detection that link can operate in x type.

THANK YOU