ENT-AN1164
ResilientRing™ Configuration

August 2018

Microsemi
a Microchip company

Contents

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 1.3

The following is a summary of the changes in revision 1.3 of this document.

- Figure 2 and 3 were updated. For more information, see figures Master PHY Switching to Local Clock and Slave PHY Becomes the Timing Master.
- The section Communicating R-R Support was updated. For more information, see Communicating R-R Support.
- Additional information was added to the section R-R Notes. For more information, see R-R Notes.

1.2 Revision 1.2

The following is a summary of the changes in revision 1.2 of this document.

- The Application Note number was added to the document title.
- The explanation of the methodology in the ResilientRing™ application was clarified. For more information, see Methodology.
- The bit value indicating the PHY master was corrected from 0 to 1. For more information, see R-R Notes.

1.3 Revision 1.1

In revision 1.1 of this document, the programming section was added. For more information, see Program Control of ResilientRing™.

1.4 Revision 1.0

Revision 1.0 was the first publication of this document.

2 ResilientRing™ Configuration

This document describes the Ring Resiliency feature available in many Microsemi Gigabit Ethernet PHYs.

2.1 1000BASE-T Master/Slave Issue in Synchronous Ethernet

2.1.1 Current Scenario

1000BASE-T operation requires master/slave configuration. Timing for all nodes in a 1000BASE-T synchronous Ethernet system is derived from a single grandmaster. The timing information is passed down from grandmaster to all the nodes along multiple paths in a ring topology. If an upstream link goes down, the timing synchronization is lost on certain nodes. Typically, the link is dropped and re-linked again to change the timing reference and master/slave configuration, and re-acquire timing lock. This causes traffic interruption and several seconds of network downtime.

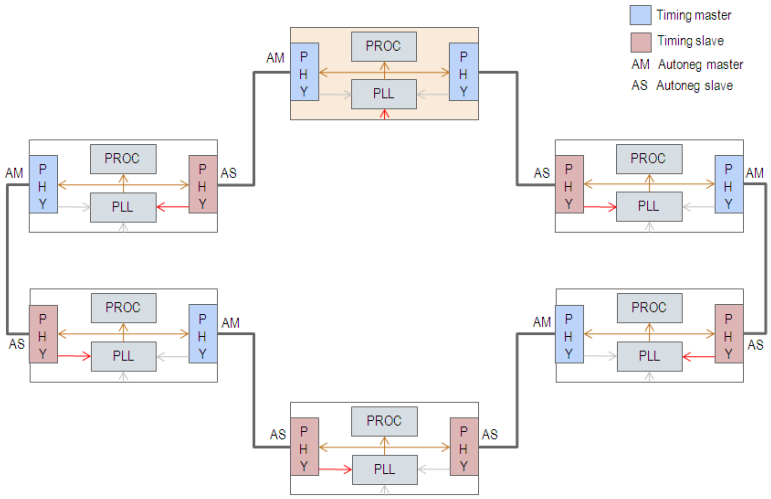
2.1.2 Microsemi Solution

Microsemi offers a solution that restores the timing synchronization during link break down without dropping the link. The solution does not change the master/slave configuration but changes the timing reference between master and slave PHYs.

2.2 ResilientRing™ Application

The following figure shows a typical system prior to link drop.

Figure 1 • Typical System Prior to Link Drop



2.2.1 Methodology

The master PHY transmitter sends data according to the local clock (as shown in figure Master PHY Switching to Local Clock), and initiates timing recovery in the receiver. The slave PHY instructs the node to switch its local timing reference to a recovered clock from the other PHY in the box (by means of an interruption to its own recovered clock), freezes timing recovery, and locks the frequency of the clock for the transmitter (as shown in figure Slave PHY Becomes the Timing Master). The master PHY makes a smooth transition of its transmit path from the local clock to the recovered clock after timing lock is achieved (as shown in figure Master PHY After Timing Lock).

Figure 2 • Master PHY Switching to Local Clock

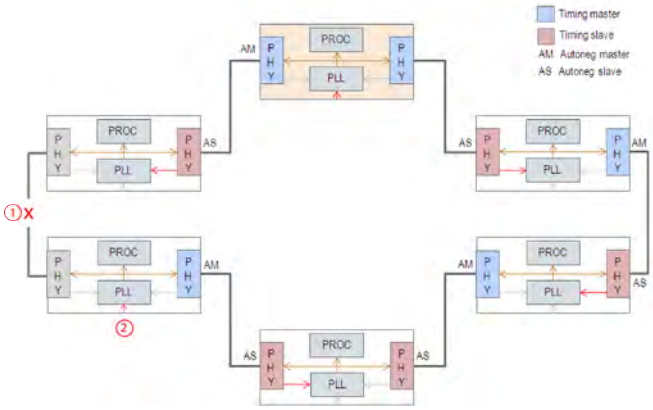


Figure 3 • Slave PHY Becomes the Timing Master

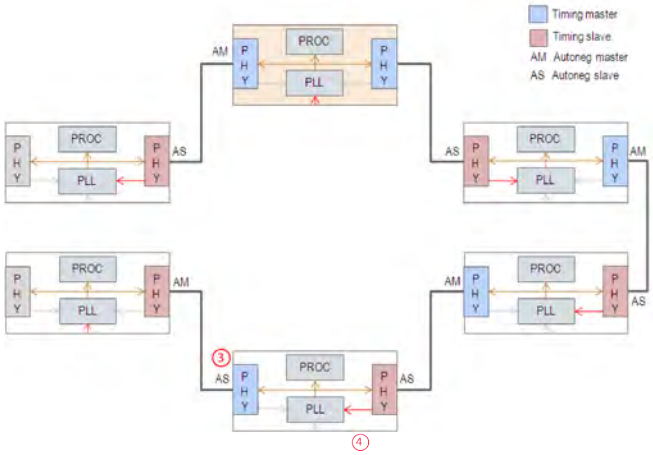
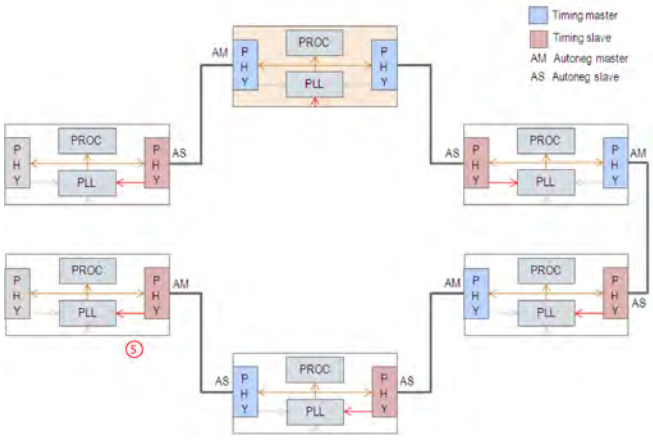


Figure 4 • Master PHY After Timing Lock



This methodology is possible in a synchronous Ethernet system because local clocks in each node are synchronized to the grandmaster clock.

2.3 Program Control of ResilientRing™

To use ResilientRing (R-R), the two connected PHYs must have the following characteristics:

- Microsemi PHYs supporting R-R
- Connected at 1000BASE-T
- R-R notification enabled before the last Auto-Negotiation (ANEG)

2.3.1 Enabling R-R Notification

Set register 30E2 bits 15 and 14 both to 1.

Bit 15 will turn on Master timing recovery.

Bit 14 will enable Microsemi proprietary ANEG communication of R-R support.

```
smiwrite (<PHY#>, 31, 0x2);         // Change register base to extension 2 page
temp = smiread (<PHY#>, 30) ;       // Read register 30E2
temp |= 0xc000;                     // Set bits 15 and 14
smiwrite(<PHY#>, 30, temp);         // Update register 30E2
smiwrite (<PHY#>, 31, 0x0);         // Change register base to main page
```

2.3.2 Communicating R-R Support

To communicate R-R support, an ANEG or restart ANEG must occur after register 30E2.14 has been set to 1 in each PHY at both ends of the link.

```
temp = smiread (<PHY#>, 0);         // Read register 0, assume base is main page
temp |= 0x0200;                     // Set bit 9
smiwrite(<PHY#>, temp);             // Restart ANEG cycle
smiwrite (<PHY#>, 31, 0x0);         // Set register page to main page
```

2.3.3 Checking if Link Partner Supports R-R

After the ANEG, check register 30E2.13. If set to 1 then the link partner supports R-R.

```
smiwrite (<PHY#>, 31, 0x2);         // Change register base to extension 2 page
If (smiread (PHY, 30) & 0x2000)
    link_partner_supports_r-r = TRUE;
smiwrite (<PHY#>, 31, 0x0);         // Change register base to main page
```

2.3.4 Swapping of Master-Slave Relationship with R-R

To cause a swap of master and slave relationship, set register 30E2.0 to 1.

While the swap is in process, bits 30E2.4 and 30E2.5 will not be equal. Software must wait for bits 4 and 5 to become equal before proceeding.

```
If (link_partner_supports_r-r == TRUE)
     {
     smiwrite (<PHY#>, 31, 0x2);          // Change register base to extension 2 page

     temp = smiread (<PHY#>, 30);         // Read register 30E2
     temp |= 0x1;                         // Set bit 0 to swap
     smiwrite (<PHY#>, 30, temp);         // Start swap

     timeoutstart (3000);                 // Start timer for 3 seconds
     error = TIMEOUT_R-R_swap;            // Set error code to timeout occurred on swap
     while ( !timeout() )                 // Exit loop
         {
         temp = smiread(<PHY#>, 30)       // Read register 30E2
         if ((temp & 0x30 == 0x30) ||
             (temp & 0x30 == 0x00))       // Check if bit 4 == bit 5
             {
             error = OK;                  // Set error code to pass
             break;                       // Exit while loop
             }
           }  // End while
     smiwrite (<PHY#>, 31, 0x0)           // Change register base to main page
     }
else
     error = R-R_NOT_SUPPORTED;           // Can't do R-R not supported by link partner
```

2.3.5 R-R Notes

- When status bits 30E2.4 and 30E2.5 equal 0, then the PHY is the timing slave.
- When status bits 30E2.4 and 30E2.5 equal 1, then the PHY is the timing master.
- Register 10.14 slave/master state will not be changed after a R-R swap.
- Having EEE enabled will slow R-R swaps.
- R-R swaps happen after the current packet is finished being sent.
- PPM difference between link partners will slow R-R swap, for example.
  - 200 ppm will take about 2 seconds.
  - 50 ppm will take about 0.5 second.
- Interpacket gap must be long enough for R-R signaling, IPG= 12 recommended.
- R-R swap supported in forced 1000BT.
- R-R feature is only supported between Microsemi Gigabit Ethernet PHYs.

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

VPPD-02862