TSE MAC Driver API Reference

Technical Note

FPGA-TN-02341-1.0

December 2023

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Contents

Contents....3
Acronyms in This Document....6
1. Introduction....7
1.1. Purpose....7
1.2. Audience....7
1.3. Driver Versioning....7
1.3.1. Driver Version....7
1.3.2. IP Version....7
1.4. Driver and IP Compatibility....7
2. API Description....8
2.1. Ethernet_init()....8
2.2. Ethernet_packet_handle()....8
2.3. Ethernet_set_mac_address()....8
2.4. Ethernet_get_mac_address()....9
2.5. Ethernet_set_multicast_address()....9
2.6. Ethernet_get_multicast_address()....9
2.7. Ethernet_statistics_counter_register_read()....9
2.8. Ethernet_tx_rx_status_reg_read()....10
2.9. Ethernet_mode_reg_read()....10
2.10. Ethernet_tx_rx_control_reg_set()....10
2.11. Ethernet_set_speed()....10
3. Function Call Flow Diagrams....11
3.1. Ethernet_init()....11
3.2. Ethernet_packet_handle()....12
3.3. Ethernet_set_mac_address()....12
3.4. Ethernet_get_mac_address()....13
3.5. Ethernet_set_multicast_address()....13
3.6. Ethernet_get_multicast_address()....14
3.7. Ethernet_statistics_counter_register_read()....14
3.8. Ethernet_tx_rx_status_reg_read()....15
3.9. Ethernet_mode_reg_read()....15
3.10. Ethernet_tx_rx_control_reg_set()....16
3.11. Ethernet_set_speed()....17
4. API Data Structures....18
4.1. tsemac_reg_type_t....18
4.2. tsemac_handle_t....21
5. API Enum....22
5.1. speed_mode_set....22
5.2. statistics_counter_reg....22
6. API Variables....25
7. API Macros....26
7.1. Success and Failure....26
7.2. Frame Length....26
7.3. IPG Time....26
7.4. Maximum Packet size....26
7.5. Control Register Macros....26
7.6. Mode Register Macros....26
7.7. Shift value....26
7.8. Index value....26
References....27
Technical Support Assistance....28
Revision History....29

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Figures

Figure 3.1. ethernet_init() 11
Figure 3.2. ethernet_packet_handle() 12
Figure 3.3. ethernet_set_mac_address() 12
Figure 3.4. ethernet_get_mac_address() 13
Figure 3.5. ethernet_set_multicast_address() 13
Figure 3.6. ethernet_get_multicast_address() 14
Figure 3.7. ethernet_statistics_counter_register_read() 14
Figure 3.8. ethernet_tx_rx_status_reg_read() 15
Figure 3.9. ethernet_mode_reg_read() 15
Figure 3.10. ethernet_tx_rx_control_reg_set() 16
Figure 3.11. ethernet_set_speed() 17

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Tables

Table 4.1. tsemac_handle_t Parameters 21
Table 5.1. speed_mode_set Enum Variables 22
Table 5.2. statistics_counter_reg Enum Variables 22
Table 6.1. Variable Description 25

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AHB	Advance High Performance
APB	Advanced Peripheral Bus
API	Application Programming Interface
AXI	Advanced extensible Interface
CRC	Cyclic Redundancy Check
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
MAC	Media Access Controller
SDK	Software Development Kit
TSEMAC	Tri-Speed Ethernet Media Access Controller

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

1. Introduction

Tri-Speed Ethernet Media Access Controller (TSEMAC) IP core is a complex core containing all necessary logic and interfacing and clocking infrastructures necessary to integrate an external industry-standard Ethernet PHY with an internal processor efficiently with minimal overhead.

The TSEMAC IP core supports the ability to transmit and receive data between standard interfaces such as APB, AHB-Lite or AXI4-lite, and an Ethernet network. The main function of the TSEMAC IP is to ensure that the Media Access rules specified in the 802.3 IEEE standard are met while transmitting a frame of data over Ethernet. On the receiving side, the TSEMAC extracts different components of a frame and transfers them to higher applications through the AXI4-stream interface.

Refer to the Tri-Speed Ethernet MAC IP Core – Lattice Radiant Software (FPGA-IPUG-02084) user guide for more details about the IP core.

1.1. Purpose

TSEMAC and its SDK is a set of application programming interfaces (APIs) that provide access to specific Lattice hardware and software capabilities. This document is intended to act as a reference guide for developers by providing details of the C language driver APIs and function call flows.

1.2. Audience

The intended audience for this document includes embedded system designers and embedded software developers using Lattice MachXO2™, MachXO3D™, MachXO3L™, MachXO3LF™, CrossLink™-NX, Certus™-NX, CertusPro™-NX, Mach™-NX, MachXO5™-NX, and Lattice Avant™ devices. The technical guide assumes readers have expertise in embedded systems and FPGA technologies.

1.3. Driver Versioning

1.3.1. Driver Version

Driver version: 1.0.0

1.3.2. IP Version

IP version: 1.4.2

1.4. Driver and IP Compatibility

Driver version	IP version
TSEMAC	1.4.2
RISC-V-RX	2.2.0
PLL	1.7.0
OSC	1.4.0
APB Interconnect	1.2.0
Memory	2.0.0
AXI4 Interconnect	1.2.0
AXI4 to APB Bridge	1.1.0
GPIO	1.6.1
UART	1.3.0

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2. API Description

2.1. Ethernet_init()

This API initializes the TSEMAC IP with a speed of 10/100/1000 Mbps and configures Tx MAC and Rx MAC to receive any frame address and multicast and broadcast frames.

```
uint8_t ethernet_init(tsemac_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.	0: failure 1: success
In	speed mode	Part of the structure used to set the speed 10/100/1000 Mbps for TSE MAC.	
In	tx_rx_ctrl_var	• Part of the tsemac_handle_t structure. • Used for setting the control register to receive short pause frames and to receive the multicast and broadcast frames.	
In	enable_tx_mac	Part of the tsemac_handle_t structure used for enabling the Tx MAC.	
In	enable_rx_mac	Part of the tsemac_handle_t structure used for enabling the Rx MAC.	

2.2. Ethernet_packet_handle()

This API handles the transmitting and receiving of Ethernet packet by using shared memory and data mover. The same packet is looped back by the TSEMAC IP into shared memory.

```
void ethernet_packet_handle(tsemac_handle_t *handle,unsigned int *src_packet,unsigned int
*dest_packet)
```

In/Out	Parameter	Description	Returns
In/Out	src_packet	• Parameter used for both input and output. • Represents the Ethernet packet to transmit to TSEMAC.	None
In/Out	dest_packet	• Parameter used for both input and output. • Represents the Ethernet packet received from TSEMAC.	

2.3. Ethernet_set_mac_address()

This API sets the six-byte MAC address for a source or destination device for TSEMAC.

```
unsigned char ethernet_set_mac_address(tsemac_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.	0: failure 1: success
In	mac_upper	Part of the tsemac_handle_t structure used to set the first four bytes of the mac address for TSEMAC.	
In	mac_lower	Part of the tsemac_handle_t structure used to set the last two bytes of the mac address for TSEMAC.	

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.4. Ethernet_get_mac_address()

This API reads the six-byte MAC address from MAC address word 0 and MAC address word 1 register of TSEMAC.

```
unsigned char ethernet_get_mac_address(tsemac_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.	0: failure 1: success

2.5. Ethernet_set_multicast_address()

This API sets the 8-byte multicast address for multicast frame from the 64-bit hash table. The first four bytes of the 64-bit hash table is stored into Multicast Table Word 0 register and the last four bytes of the 64-bit hash table is stored into Multicast Table Word 1 register.

```
unsigned char ethernet_set_multicast_address(tsemac_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.	0: failure 1: success
In	multicast_upper	• A 4-byte variable of the tsemac_handle_t structure. • Used to set the first four bytes of multicast address for TSEMAC.	
In	multicast_lower	• A 4-byte variable of the tsemac_handle_t structure. • Used to set the last four bytes of multicast address for TSEMAC.	

2.6. Ethernet_get_multicast_address()

This API reads the eight-byte multicast address from Multicast Table Word 0 and Multicast Table Word 1 register of TSEMAC.

```
unsigned char ethernet_get_multicast_address(tsemac_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.	0: failure 1: success

2.7. Ethernet_statistics_counter_register_read()

This API reads the statistics counter register of TSEMAC when multiple frames are transmitted or received.

```
unsigned int ethernet_statistics_counter_register_read(tsemac_handle_t handle)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.	total_count: The count of the total frames transmitted or received after the statistics counter register is read.
In	stat_counter_reg_enum	• Part of the tsemac_handle_t structure. • Represents the index position of the statistics counter register.	

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

2.8. Ethernet_tx_rx_status_reg_read()

This API reads the transmit and receive status register of TSEMAC.

```
unsigned int ethernet_tx_rx_status_reg_read(tsemac_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.	Transmit and receive status register read value.

2.9. Ethernet_mode_reg_read()

This API reads the TSEMAC mode register.

```
unsigned int ethernet_mode_reg_read(tsemac_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.	Mode register read value.

2.10. Ethernet_tx_rx_control_reg_set()

This API sets the bit for the control register.

```
unsigned char ethernet_tx_rx_control_reg_set(tsemac_handle_t *handle, unsigned char bit_pos)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.	0: failure 1: success
In	bit_pos	• Represents the bit position that the control register sets.	

2.11. Ethernet_set_speed()

This API sets the bit for the control register.

```
unsigned char ethernet_set_speed(tsemac_handle_t *handle)
```

In/Out	Parameter	Description	Returns
In	adr	• Part of tsemac_handle_t structure. • Represents the base address of TSEMAC.	0: failure 1: success
In	speed_mode	• Part of the structure used to set the speed 10/100/1000 Mbps for TSE MAC.	

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3. Function Call Flow Diagrams

3.1. Ethernet_init()

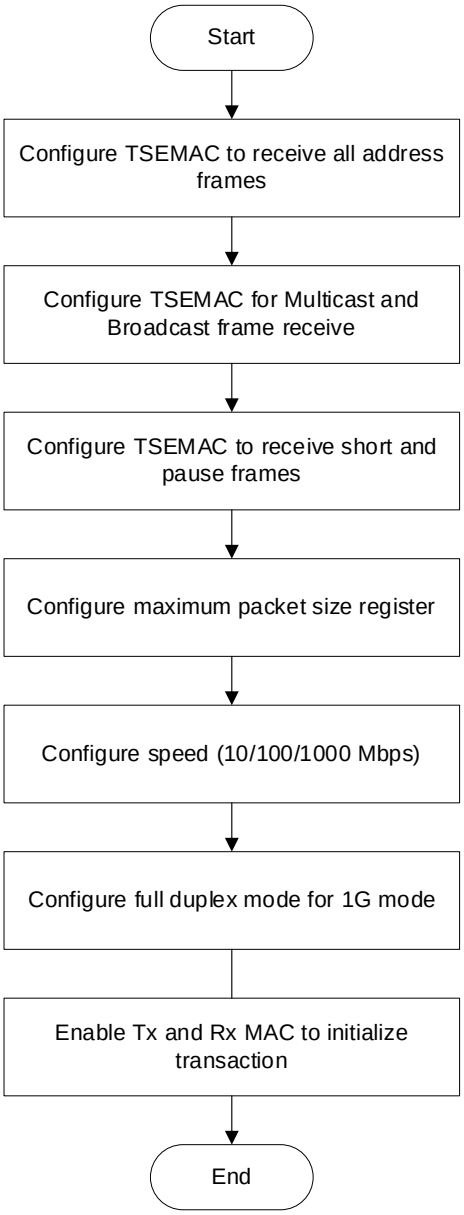

Figure 3.1. ethernet_init()

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3.2. Ethernet_packet_handle()

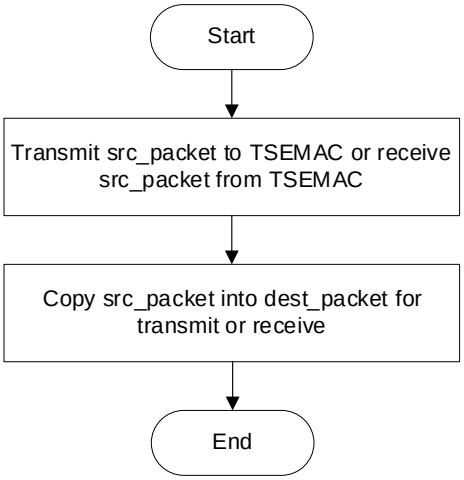

Figure 3.2. ethernet_packet_handle()

3.3. Ethernet_set_mac_address()

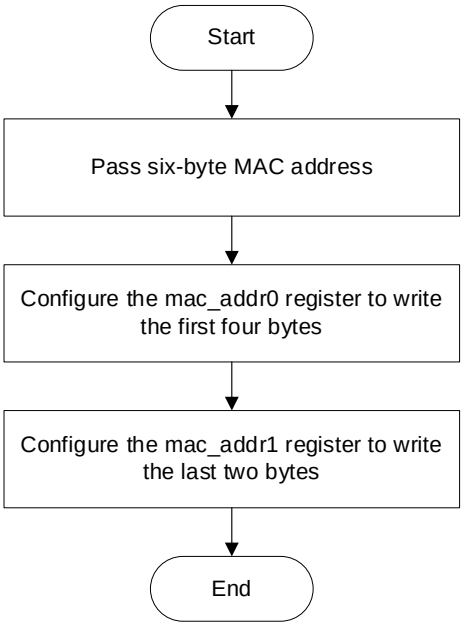

Figure 3.3. ethernet_set_mac_address()

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3.4. Ethernet_get_mac_address()

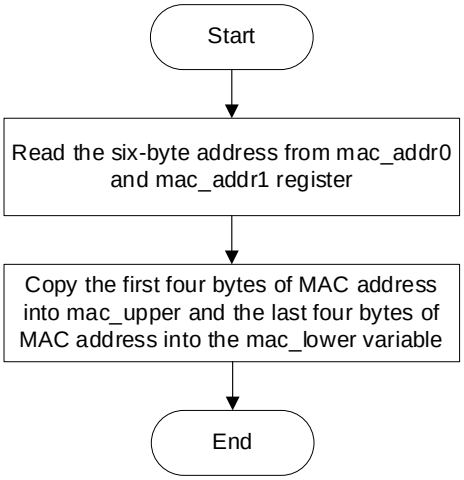

Figure 3.4. ethernet_get_mac_address()

3.5. Ethernet_set_multicast_address()

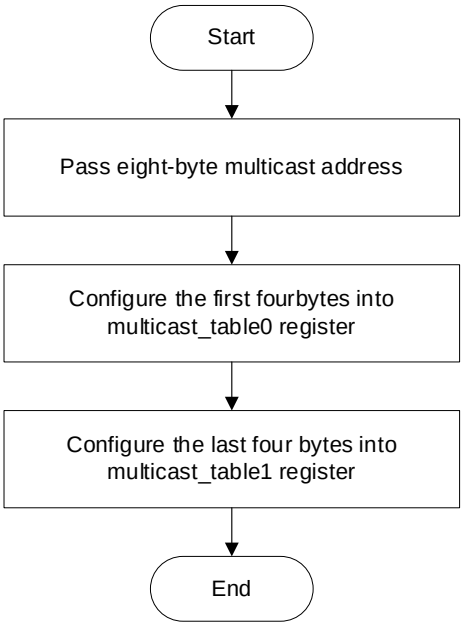

Figure 3.5. ethernet_set_multicast_address()

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3.6. Ethernet_get_multicast_address()

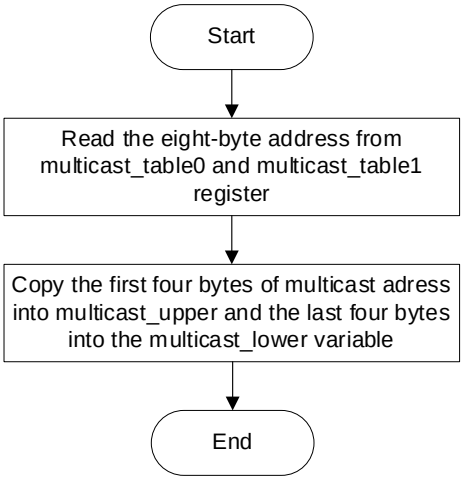

Figure 3.6. ethernet_get_multicast_address()

3.7. Ethernet_statistics_counter_register_read()

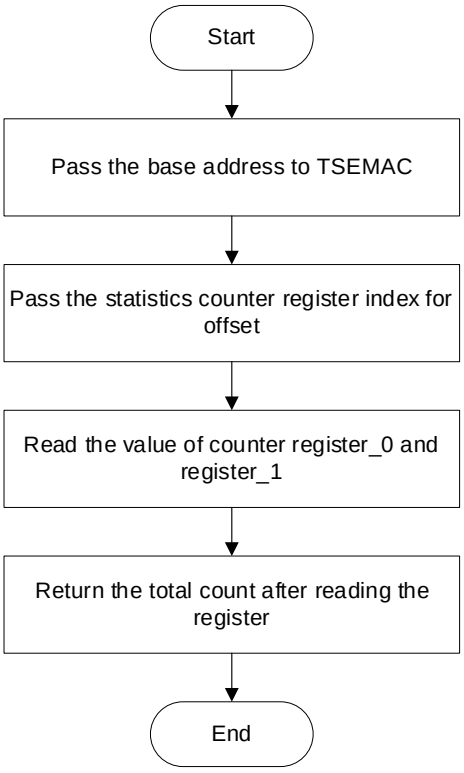

Figure 3.7. ethernet_statistics_counter_register_read()

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3.8. Ethernet_tx_rx_status_reg_read()

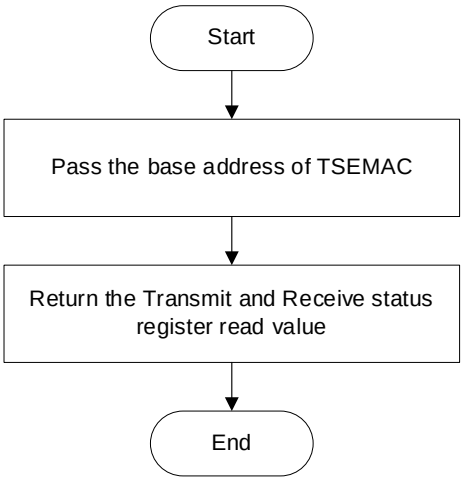

Figure 3.8. ethernet_tx_rx_status_reg_read()

3.9. Ethernet_mode_reg_read()

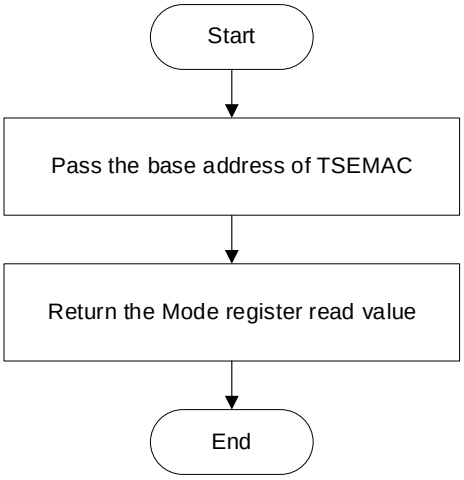

Figure 3.9. ethernet_mode_reg_read()

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3.10. Ethernet_tx_rx_control_reg_set()

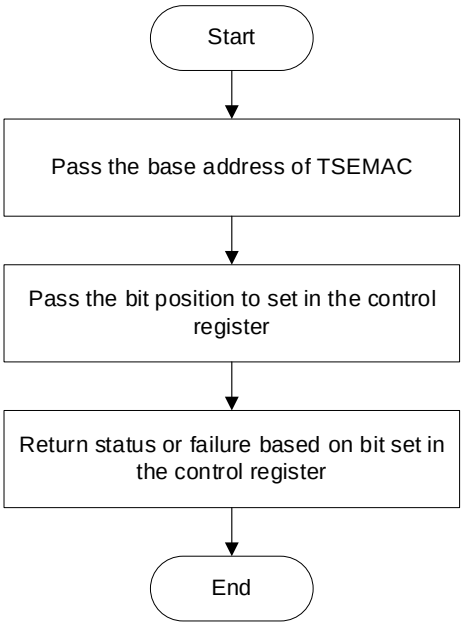

Figure 3.10. ethernet_tx_rx_control_reg_set()

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3.11. Ethernet_set_speed()

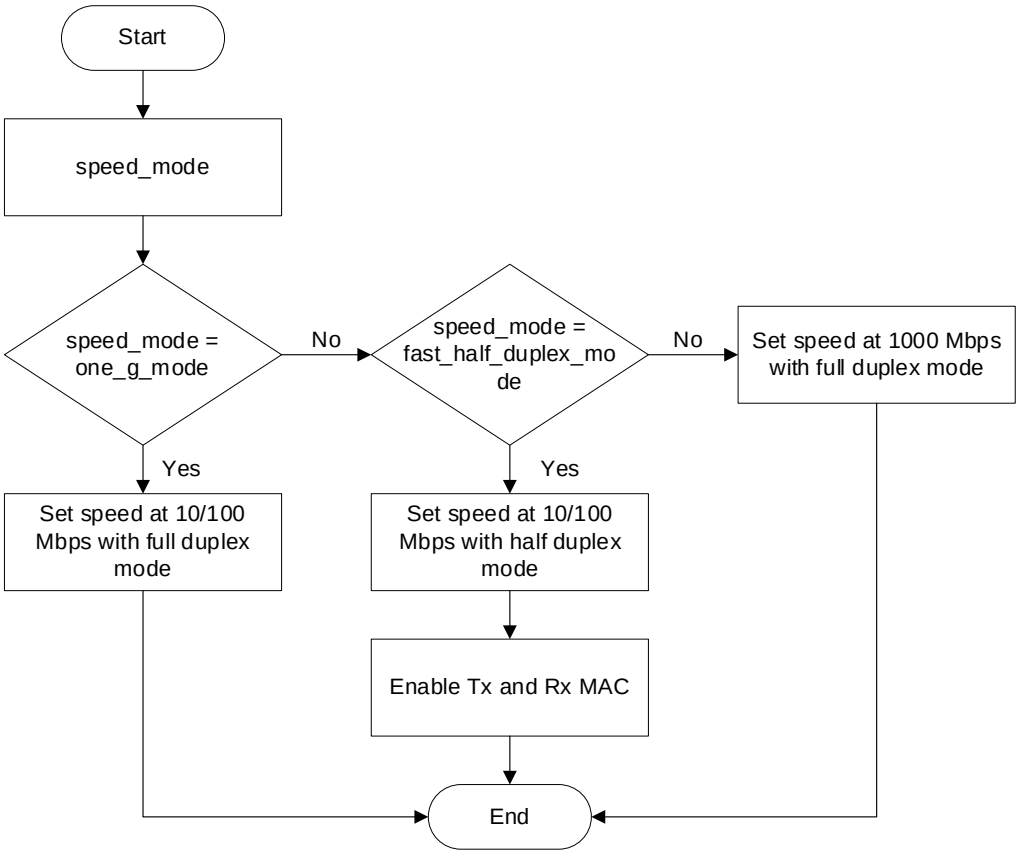

Figure 3.11. ethernet_set_speed()

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4. API Data Structures

4.1. tsemac_reg_type_t

This structure assigns offsets for the TSEMAC register.

```
volatile unsigned int mode_reg;          //0x0000
volatile unsigned int tx_rx_ctrl;
volatile unsigned int max_packet_size;
volatile unsigned int ipg;
volatile unsigned int mac_addr0;         //0x0010
volatile unsigned int mac_addr1;
volatile unsigned int tx_rx_status;
volatile unsigned int vlan_tag;
volatile unsigned int gmii_mgmt_ctrl;    //0x0020
volatile unsigned int gmii_mgmt_data;
volatile unsigned int multi_cast0;
volatile unsigned int multi_cast1;
volatile unsigned int pause_opcode;      //0x0030
volatile unsigned int tx_fifo_afull;
volatile unsigned int tx_fifo_aempty;
volatile unsigned int rx_fifo_afull;
volatile unsigned int rx_fifo_aempty;    //0x0040
volatile unsigned int interrupt_status;
volatile unsigned int interrupt_enable;        //0x0048
volatile unsigned int tx_stat_unicst_0;
volatile unsigned int tx_stat_unicst_1;
volatile unsigned int tx_stat_multcst_0;
volatile unsigned int tx_stat_multcst_1;       //0x0058
volatile unsigned int tx_stat_brdcst_0;
volatile unsigned int tx_stat_brdcst_1;
volatile unsigned int tx_stat_badfcs_0;
volatile unsigned int tx_stat_badfcs_1;        //0x0068
volatile unsigned int tx_stat_jumbo_0;
volatile unsigned int tx_stat_jumbo_1;
volatile unsigned int tx_stat_under_run_0;
volatile unsigned int tx_stat_under_run_1;     //0x0078
volatile unsigned int tx_stat_pause_0;
volatile unsigned int tx_stat_pause_1;
volatile unsigned int tx_stat_vlan_tg_0;
volatile unsigned int tx_stat_vlan_tg_1; //0x0088
volatile unsigned int tx_stat_frm_length_0;
volatile unsigned int tx_stat_frm_length_1;
volatile unsigned int tx_stat_deferred_trans_0;
volatile unsigned int tx_stat_deferred_trans_1; //0x0098
volatile unsigned int tx_stat_excess_deferred_trans_0;
volatile unsigned int tx_stat_excess_deferred_trans_1;
volatile unsigned int tx_stat_late_col_0;
volatile unsigned int tx_stat_late_col_1;             //0x00A8
volatile unsigned int tx_stat_excess_col_0;
volatile unsigned int tx_stat_excess_col_1;
volatile unsigned int tx_stat_num_early_col_0;
```

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

```
volatile unsigned int tx_stat_num_early_col_1;  //0x00B8
volatile unsigned int tx_stat_shrt_frm_dis_fcs_0;
volatile unsigned int tx_stat_shrt_frm_dis_fcs_1;
volatile unsigned int tx_stat_ptp_1588_frm_0;
volatile unsigned int tx_stat_ptp_1588_frm_1;         //0x00C8
volatile unsigned int tx_stat_frm_64_0;
volatile unsigned int tx_stat_frm_64_1;
volatile unsigned int tx_stat_frm_65_127_0;
volatile unsigned int tx_stat_frm_65_127_1;           //0x00D8
volatile unsigned int tx_stat_frm_128_255_0;
volatile unsigned int tx_stat_frm_128_255_1;
volatile unsigned int tx_stat_frm_256_511_0;
volatile unsigned int tx_stat_frm_256_511_1;          //0x00E8
volatile unsigned int tx_stat_frm_512_1023_0;
volatile unsigned int tx_stat_frm_512_1023_1;
volatile unsigned int tx_stat_frm_1024_1518_0;
volatile unsigned int tx_stat_frm_1024_1518_1;  //0x00F8
volatile unsigned int tx_stat_frm_1519_2047_0;
volatile unsigned int tx_stat_frm_1519_2047_1;
volatile unsigned int tx_stat_frm_2048_4095_0;
volatile unsigned int tx_stat_frm_2048_4095_1;  //0x108
volatile unsigned int tx_stat_frm_4096_9216_0;
volatile unsigned int tx_stat_frm_4096_9216_1;
volatile unsigned int tx_stat_frm_9217_16383_0;
volatile unsigned int tx_stat_frm_9217_16383_1; //0x118
volatile unsigned int rx_stat_frm_length_0;
volatile unsigned int rx_stat_frm_length_1;
volatile unsigned int rx_stat_vlan_tg_0;
volatile unsigned int rx_stat_vlan_tg_1;        //0x128
volatile unsigned int rx_stat_pause_0;
volatile unsigned int rx_stat_pause_1;
volatile unsigned int rx_stat_ctrl_0;
volatile unsigned int rx_stat_ctrl_1;                 //0x138
volatile unsigned int rx_stat_unsp_opcode_0;
volatile unsigned int rx_stat_unsp_opcode_1;
volatile unsigned int rx_stat_dribb_nibb_0;
volatile unsigned int rx_stat_dribb_nibb_1;           //0x148
volatile unsigned int rx_stat_brdcst_0;
volatile unsigned int rx_stat_brdcst_1;
volatile unsigned int rx_stat_multcst_0;
volatile unsigned int rx_stat_multcst_1;        //0x158
volatile unsigned int rx_stat_unicst_0;
volatile unsigned int rx_stat_unicst_1;
volatile unsigned int rx_stat_rcvd_ok_0;
volatile unsigned int rx_stat_rcvd_ok_1;        //0x168
volatile unsigned int rx_stat_length_error_0;
volatile unsigned int rx_stat_length_error_1;
volatile unsigned int rx_stat_crc_error_0;
volatile unsigned int rx_stat_crc_error_1;            //0x178
volatile unsigned int rx_stat_pkt_ignore_0;
volatile unsigned int rx_stat_pkt_ignore_1;
volatile unsigned int rx_stat_previous_carrier_event_0;
```

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

```
volatile unsigned int rx_stat_previous_carrier_event_1;      //0x188
volatile unsigned int rx_stat_ptp1588_frm_0;
volatile unsigned int rx_stat_ptp1588_frm_1;
volatile unsigned int rx_stat_ipg_viol_0;
volatile unsigned int rx_stat_ipg_viol_1;       //0x198
volatile unsigned int rx_stat_shrt_frm_0;
volatile unsigned int rx_stat_shrt_frm_1;
volatile unsigned int rx_stat_lng_frm_0;
volatile unsigned int rx_stat_lng_frm_1; //0x1A8
volatile unsigned int rx_stat_frm_undersize_0;
volatile unsigned int rx_stat_frm_undersize_1;
volatile unsigned int rx_stat_frm_fragement_0;
volatile unsigned int rx_stat_frm_fragement_1; //0x1B8
volatile unsigned int rx_stat_frm_jabber_0;
volatile unsigned int rx_stat_frm_jabber_1;
volatile unsigned int rx_stat_frm_64_good_crc_0;
volatile unsigned int rx_stat_frm_64_good_crc_1; //0x1C8
volatile unsigned int rx_stat_frm_1518_good_crc_0;
volatile unsigned int rx_stat_frm_1518_good_crc_1;
volatile unsigned int rx_stat_frm_64_0;
volatile unsigned int rx_stat_frm_64_1;         //0x1D8
volatile unsigned int rx_stat_frm_65_127_0;
volatile unsigned int rx_stat_frm_65_127_1;
volatile unsigned int rx_stat_frm_128_255_0;
volatile unsigned int rx_stat_frm_128_255_1;    //0x1E8
volatile unsigned int rx_stat_frm_256_511_0;
volatile unsigned int rx_stat_frm_256_511_1;
volatile unsigned int rx_stat_frm_512_1023_0;
volatile unsigned int rx_stat_frm_512_1023_1;    //0x1F8
volatile unsigned int rx_stat_frm_1024_1518_0;
volatile unsigned int rx_stat_frm_1024_1518_1;
volatile unsigned int rx_stat_frm_1519_2047_0;
volatile unsigned int rx_stat_frm_1519_2047_1;  //0x208
volatile unsigned int rx_stat_frm_2048_4095_0;
volatile unsigned int rx_stat_frm_2048_4095_1;
volatile unsigned int rx_stat_frm_4096_9216_0;
volatile unsigned int rx_stat_frm_4096_9216_1;  //0x218
volatile unsigned int rx_stat_frm_9217_16383_0;
volatile unsigned int rx_stat_frm_9217_16383_1;
```

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

4.2. tsemac_handle_t

This structure is used to declare different types of parameters used for the TSEMAC API. Table 4.1 shows the available parameters and description.

Table 4.1. tsemac_handle_t Parameters

Parameter	Description
speed_mode	• Part of the tsemac_handle_t structure. • Used for setting the speed for TSEMAC.
adr	• Part of the tsemac_handle_t structure. • Represents the base address of TSEMAC.
frame_length	• Part of the tsemac_handle_t structure. • Represents the Ethernet packet length to be transmitted to TSEMAC.
mac_upper	• Part of the tsemac_handle_t structure. • Represents the variable to set and get first four bytes of MAC address for TSEMAC.
mac_lower	• Part of the tsemac_handle_t structure. • Represents the variable to set and get last two bytes of MAC address for TSEMAC.
multicast_upper	• Part of the tsemac_handle_t structure. • Represents the variable to set and get first four bytes of 64-bit hash for TSEMAC.
multicast_lower	• Part of the tsemac_handle_t structure. • Represents the variable to set and get last four bytes of 64-bit hash for TSEMAC.
tx_rx_ctrl_var	• Part of the tsemac_handle_t structure. • Represents the transmit and receive control register configuration for TSEMAC.
enable_tx_mac	• Part of the tsemac_handle_t structure. • Used to enable the Tx MAC.
enable_rx_mac	• Part of the tsemac_handle_t structure. • Used to enable the Rx MAC.

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

5. API Enum

5.1. speed_mode_set

This enum sets the different speeds and modes for TSEMAC. Table 5.1. speed_mode_set Enum Variables shows the speed_mode_set enum variables and description.

Table 5.1. speed_mode_set Enum Variables

Variable	Description
fast_half_duplex_mode	This variable is used to set TSEMAC to half duplex mode and speed at 10/100 Mbps.
fast_full_duplex_mode	This variable is used to set TSEMAC to full duplex mode and speed at 10/100 Mbps.
one_g_mode	This variable is used to set TSEMAC to full duplex mode and speed at 1000 Mbps.

5.2. statistics_counter_reg

This enum is used to assign the offset for the statistics counter register by passing enum values. Table 5.2 shows the statistics_counter_reg enum variables and description.

Table 5.2. statistics_counter_reg Enum Variables

Variable	Description
tx_unicast_frame_counter_reg	This variable is used to read the total unicast frame transmit.
tx_multicast_frame_counter_reg	This variable is used to read the total multicast frame transmit.
tx_broadcast_frame_counter_reg	This variable is used to read the total broadcast frame transmit.
tx_badfcs_frame_counter_reg	This variable is used to read the total bad fcs frame transmit.
tx_jumbo_frame_counter_reg	This variable is used to read the total jumbo frame transmit.
tx_under_run_frame_counter_reg	This variable is used to read the total FIFO under run.
tx_pause_frame_counter_reg	This variable is used to read the total pause frame transmit.
tx_vlan_tag_frame_counter_reg	This variable is used to read the total vlan tagged frame transmit.
tx_frame_length_counter_reg	This variable is used to read the total frame length.
tx_deferred_transmission_counter_reg	This variable is used to read the total deferred frame while transmit.
tx_excessive_deferred_transmission_counter_reg	This variable is used to read the total excessive frame deferred while transmitting.
tx_late_collision_counter_reg	This variable is used to read the total number of collisions.
tx_excessive_collision_counter_reg	This variable is used to read the total number of excessive collisions.
tx_num_early_collision_counter_reg	This variable is used to read the total number of early collisions.
tx_short_frame_dis_fcs_counter_reg	This variable is used to read the total short frames transmitted when FCS field is disabled.
tx_ptp_1588_frame_counter_reg	This variable is used to read the total ptp frames transmitted.
tx_frame_length_64_counter_reg	This variable is used to read the total frames transmitted with length of 64.
tx_frame_length_65_127_counter_reg	This variable is used to read the total frames transmitted with of 65 to 127.
tx_frame_length_128_255_counter_reg	This variable is used to read the total frames transmitted with of 128 to 255.
tx_frame_length_256_511_counter_reg	This variable is used to read the total frames transmitted with of 256 to 511.
tx_frame_length_512_1023_counter_reg	This variable is used to read the total frames transmitted with of 512 to 1023.
tx_frame_length_1024_1518_counter_reg	This variable is used to read the total frames transmitted with of 1024 to 1518.
tx_frame_length_1519_2047_counter_reg	This variable is used to read the total frames transmitted with of 1519 to 2047.

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Variable	Description
tx_frame_length_2048_4095_counter_reg	This variable is used to read the total frames transmitted with of 2048 to 4095.
tx_frame_length_4096_9216_counter_reg	This variable is used to read the total frames transmitted with of 4096 to 9216.
tx_frame_length_9217_16383_counter_reg	This variable is used to read the total frames transmitted with of 9217 to 16383.
rx_frame_length_counter_reg	This variable is used to read the total frame length received.
rx_vlan_tag_frame_counter_reg	This variable is used to read the total vlan tag frames received.
rx_pause_frame_counter_reg	This variable is used to read the total pause frames received.
rx_control_frame_counter_reg	This variable is used to read the total control frames received.
rx_unsupported_opcode_reg	This variable is used to read the total unsupported opcode frames received.
rx_dribble_nibble_counter_reg	This variable is used to read the total dribble nibble frames received.
rx_broadcast_frame_counter_reg	This variable is used to read the total broadcast frames received.
rx_multicast_frame_counter_reg	This variable is used to read the total multicast frames received.
rx_unicast_frame_counter_reg	This variable is used to read the total unicast frames received.
rx_frame_received_ok_counter_reg	This variable is used to read the total frames received.
rx_frame_received_length_error_counter_reg	This variable is used to read the total error in received frame length.
rx_frame_received_crc_error_counter_reg	This variable is used to read the total crc error in received frame.
rx_frame_received_packet_ignored_counter_reg	This variable is used to read the total number of packets ignored.
rx_frame_received_carrier_event_counter_reg	This variable is used to read the total number of packets received with carrier event detected.
rx_ptp_1588_frame_counter_reg	This variable is used to read the total ptp frames received.
rx_frame_received_ipg_violation_counter_reg	This variable is used to read the total frames received with ipg violation.
rx_received_short_frame_counter_reg	This variable is used to read the total short frames received.
rx_received_long_frame_counter_reg	This variable is used to read the total long frames received.
rx_received_undersize_frame_counter_reg	This variable is used to read the total frames received of length less than 64 bytes.
rx_received_frame_fragments_counter_reg	This variable is used to read the total frame received of length less than 64 bytes and has either an FCS error or an Alignment error.
rx_received_frame_jabber_counter_reg	This variable is used to read the total frame length of more than 1518 bytes and has either an FCS error or an Alignment error.
rx_received_frame_length64_good_crc_counter_reg	This variable is used to read the total received frames with length of less than 64 bytes and has a good CRC.
rx_received_frame_length1518_good_crc_counter_reg	This variable is used to read the total received frames with length of more than 1518 bytes and has a good CRC.
rx_frame_length_64_counter_reg	This variable is used to read the total frames received with length of 64.
rx_frame_length_65_127_counter_reg	This variable is used to read the total frames received with length of 65 to 127.
rx_frame_length_128_255_counter_reg	This variable is used to read the total frames received with length of 128 to 255.
rx_frame_length_256_511_counter_reg	This variable is used to read the total frames received with length of 256 to 511.
rx_frame_length_512_1023_counter_reg	This variable is used to read the total frames received with length of 512 to 1023.
rx_frame_length_1024_1518_counter_reg	This variable is used to read the total frames received with length of 1024 to 1518.
rx_frame_length_1519_2047_counter_reg	This variable is used to read the total frames received with length of 1519 to 2047.
rx_frame_length_2048_4095_counter_reg	This variable is used to read the total frames received with length of 2048 to 4095.

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Variable	Description
rx_frame_length_4096_9216_counter_reg	This variable is used to read the total frames received with length of 4096 to 9216.
rx_frame_length_9217_16383_counter_reg	This variable is used to read the total frames received with length of 9217 to 16383.

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

6. API Variables

Table 6.1 shows the variables and their description.

Table 6.1. Variable Description

Variable	Description
handle	• Used in the TSEMAC API. • A structure variable that consists of different variables.
status	Returns success or failure from the API.

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

7. API Macros

7.1. Success and Failure

```
#define     SUCCESS                 1
#define     FAILURE                 0
```

7.2. Frame Length

```
#define     FRAME_LENGTH            64
```

7.3. IPG Time

```
#define     IPG_TIME                12
```

7.4. Maximum Packet size

```
#define     MAX_PACKET_SIZE         1500
```

7.5. Control Register Macros

```
#define     SET_FULL_DUPLEX_MODE    5
#define     SET_HALF_DUPLEX_MODE    5
```

7.6. Mode Register Macros

```
#define     SPEED_10_OR_100_MBPS    0
#define     SPEED_1G                0
```

7.7. Shift value

```
#define     SIXTEEN_BIT             16
```

7.8. Index value

```
#define     ZERO                    0
#define     ONE                     1
```

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

References

- Tri-Speed Ethernet MAC IP Core – Lattice Radiant Software (FPGA-IPUG-02084) user guide
- Avant-E web page
- Avant-G web page
- Avant-X web page
- Certus-NX web page
- CertusPro-NX web page
- CrossLink-NX web page
- Mach-NX web page
- MachXO2 web page
- MachXO3D web page
- MachXO5-NX web page
- Lattice Solutions IP Cores web page
- Lattice Radiant Software FPGA web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Revision History

Revision 1.0, December 2023

Section	Change Summary
	Initial release.

© 2023 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

www.latticesemi.com