RH850/U2A-EVA Group

Ethernet I/F Connection Guide

Introduction

This application note describes the basic procedure for connection of Ethernet interface, which are implemented in the RH850/U2A. For detailed specifications and settings, refer to User's Manual: Hardware.

Aim of this document and software is to provide supplemental information for the function on RH850/U2A. It is not intended to implement in the design for mass production.

There is no guarantee to update in this document and software to reflect the latest manual, errata, technical update and development environment. You are fully responsible for the incorporation or any other use of the information of this document in the design of your product or system, and please refer to latest manual, errata, technical update and development environment.

Target Device

- RH850/U2A-EVA Group
  - ➢ RH850/U2A16 516BGA

Target integrated development environment

- CS+ (by Renesas Electronics Corporation)
- Version : V.8.03.00
- Device file : R7F702300.DVF.V1.20

Evaluation board

- Main Board : Y-RH850-X1X-MB-T1-V1
- Piggyback Board : Y-RH850-U2A-516PIN-PB-T1-V1

Reference Document

- RH850/U2A-EVA User's Manual(Rev.1.00): R01UH0864EJ0100

At the release time of this application note the following manual version available.
The Hardware User's Manual provides information about functional and electrical behavior of the device.

Contents

1. Ethernet AVB Overview

1.1 Overview

The RH850/U2A has the Ethernet AVB (ETNB) module with the following number of units. The ETNB has one channel interface per unit. The ETNB0 is for Fast Ethernet that supports up to 100Base-TX, and the ETNB1 is for Gigabit Ethernet that supports up to 1000Base-T.

Figure 1-1 lists the channel configuration of the ETNB units.

Product Name	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516 pins)	RH850/U2A16 (373 pins)	RH850/U2A16 (292 pins)	RH850/U2A8 (373 pins)	RH850/U2A8 (292 pins)
Number of Units	1	1	1	1	1	1
Name	ETNBn (n = 0)					

Table 25.2 Number of Units (1 Gbps Ether)

Product Name	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516 pins)	RH850/U2A16 (373 pins)	RH850/U2A16 (292 pins)	RH850/U2A8 (373 pins)	RH850/U2A8 (292 pins)
Number of Units	1	1	1	1	1	1
Name	ETNBn (n = 1)					

Table 25.3 Index

Index	Description
n	Throughout this section, the individual ETNB units of Fast Ethernet and Gigabit Ethernet are identified by the index “n”. ETNB0 is for Fast Ethernet and ETNB1 is for Gigabit Ethernet.

Figure 1-1 ETNB Unit Channel Configuration

The ETNB module consists of the following function units.

・MAC Controller (E-MAC)

・DMA Transfer Controller (AVB-DMAC)

The block diagram of the ETNB0 is shown in **Figure 1-2**, and that of the ETNB1 in **Figure 1-3**.

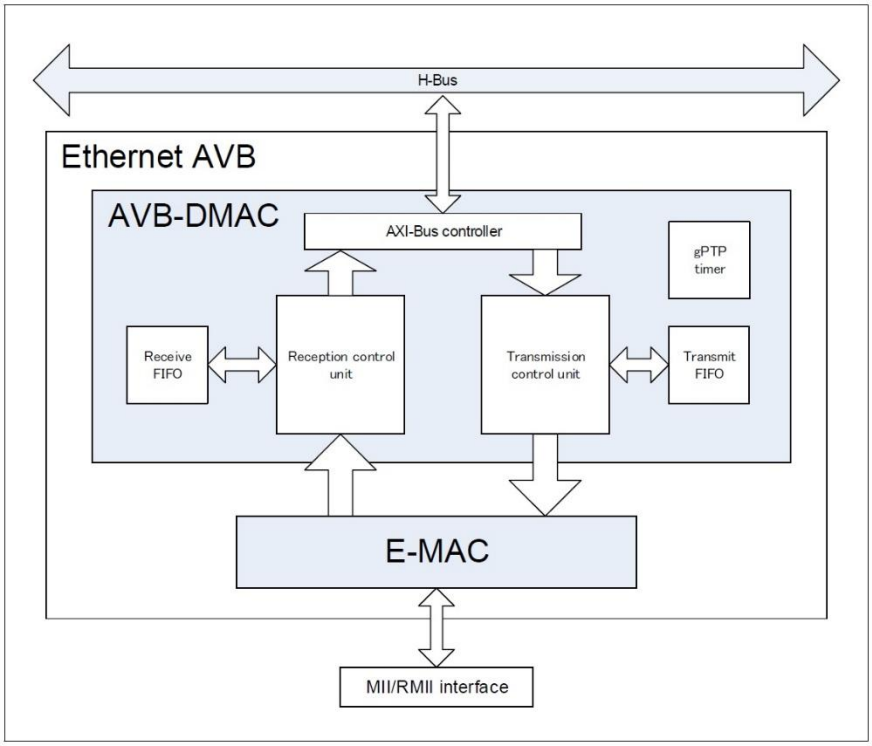

Figure 1-2 Block Diagram of ETNB0

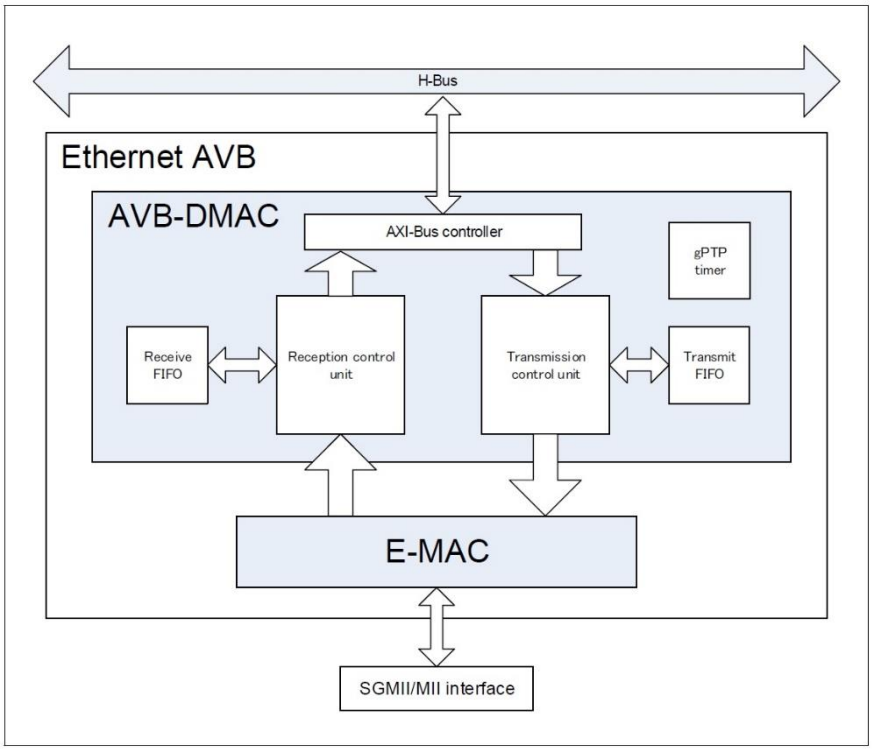

Figure 1-3 Block Diagram of ETNB1

1.2 E-MAC Overview

The E-MAC structure is shown in **Figure 1-4**. It supports a MII/RMII/SGMII, which provides an interface format for the externally connected PHY-LSI. (The ETNB0 supports the MII/RMII, and the ETNB1 supports the MII/SGMII.) The E-MAC constructs Ethernet frames from the data written to the transmission FIFO and transmits these frames to the MII/RMII/SGMII. It also performs CRC checking of Ethernet frames received from the MII/RMII/SGMII and writes these frames to the reception FIFO.

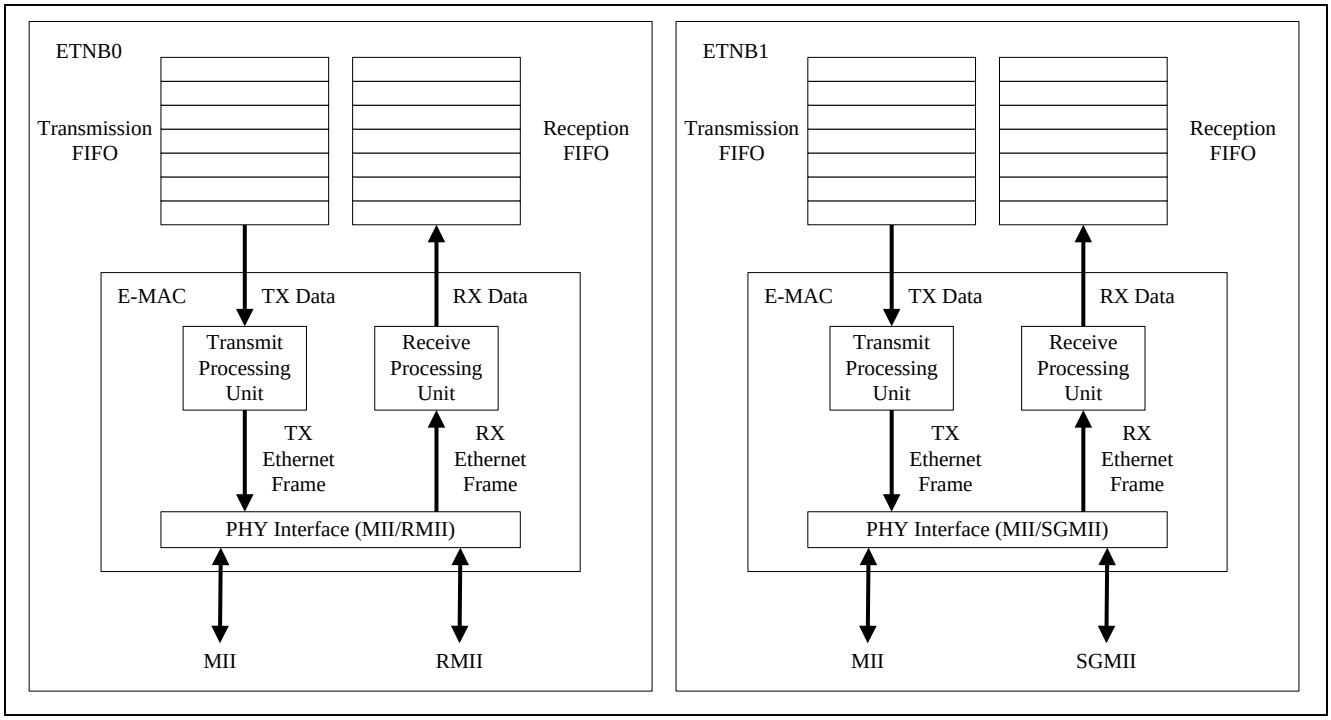

Figure 1-4 E-MAC Structure

1.3 AVB-DMAC Overview

The AVB-DMAC structure is shown in **Figure 1-5**. Using its direct memory access (DMA) function, the AVB-DMAC handles the DMA transfer of Ethernet frame data for reception and transmission between the destinations in the URAM (*1) and the FIFO buffers. Data cannot be directly read from or written to the FIFO buffers.

To handle DMA transfer, the AVB-DMAC requires information that includes the addresses for storage of data for transmission and received data. The information is referred to as descriptors. The AVB-DMAC reads data for transmission from the storage area for data to be transmitted and writes received data to the storage area for received data according to the information described in descriptors. The descriptors are placed in the URAM. Arranging multiple descriptors in descriptor lists allows the continuous reception or transmission of multiple Ethernet frames.

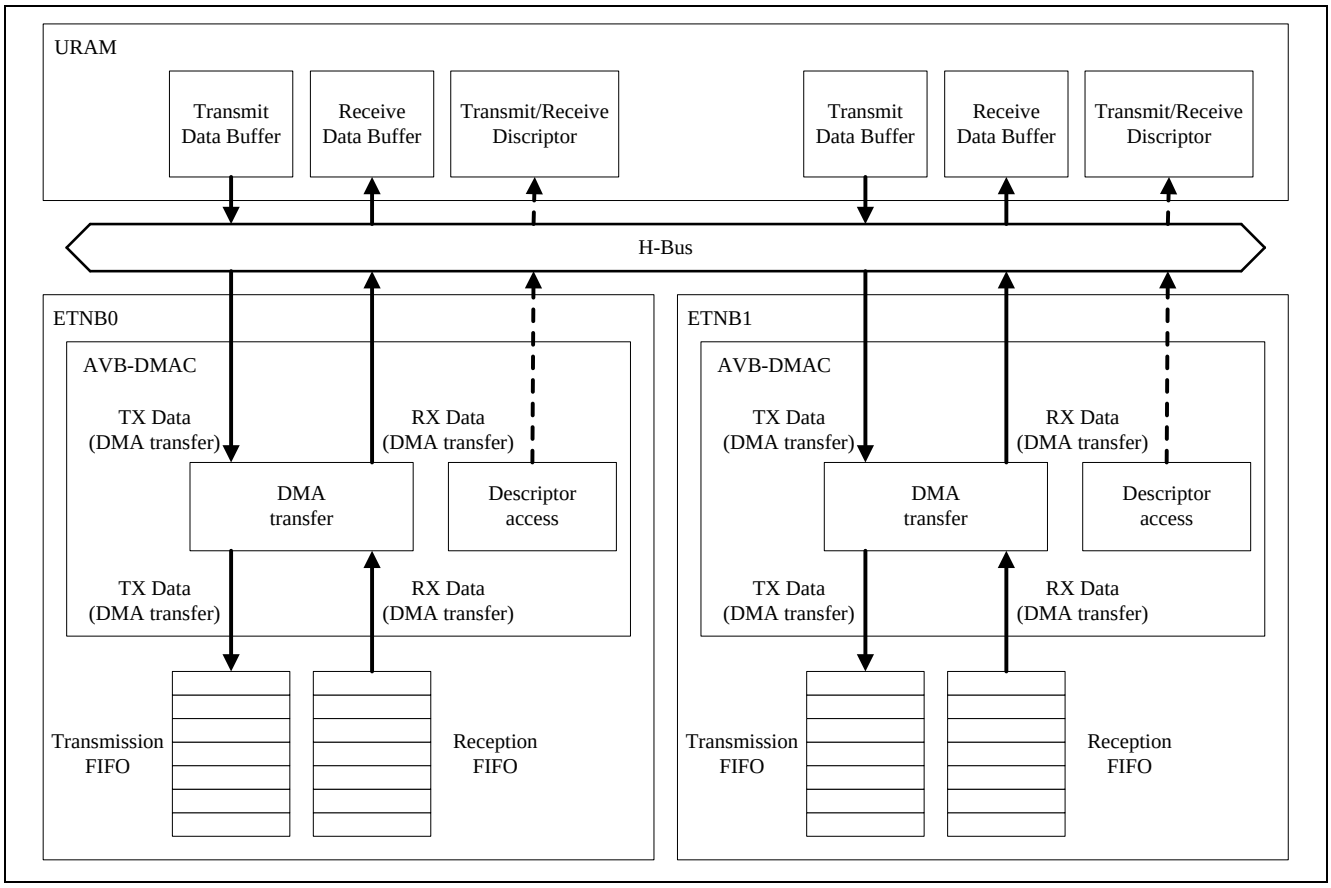

Figure 1-5 AVB-DMAC Structure

2. Ethernet AVB sample program

2.1 Overview

Figure 2-1 shows the OSI reference model and layer supported by this sample code which supports both initialization ETNB module and initialization PHY-LSI. If communicate with other device(e.g. evaluation board), it is necesary to implement upper layer protocol (e.g. TCP/IP).

OSI Reference model			Example	
Layer 7	Application layer		HTTP, FTP	Layer to be built by user
Layer 6	Presentation layer			
Layer 5	Session layer			
Layer 4	Transport layer		TCP, UDP	
Layer 3	Network layer		IP	
Layer 2	Data link layer	LLC	Ethernet	
		MAC		Layer supported by this sample code
Layer 1	Physical layer			

Figure 2-1 OSI reference model and rayer supported by this sample code

2.2 Operating environment

Figure 2-2 shows operation environment of sample code. LAN cable conecction is only for checking the link status.

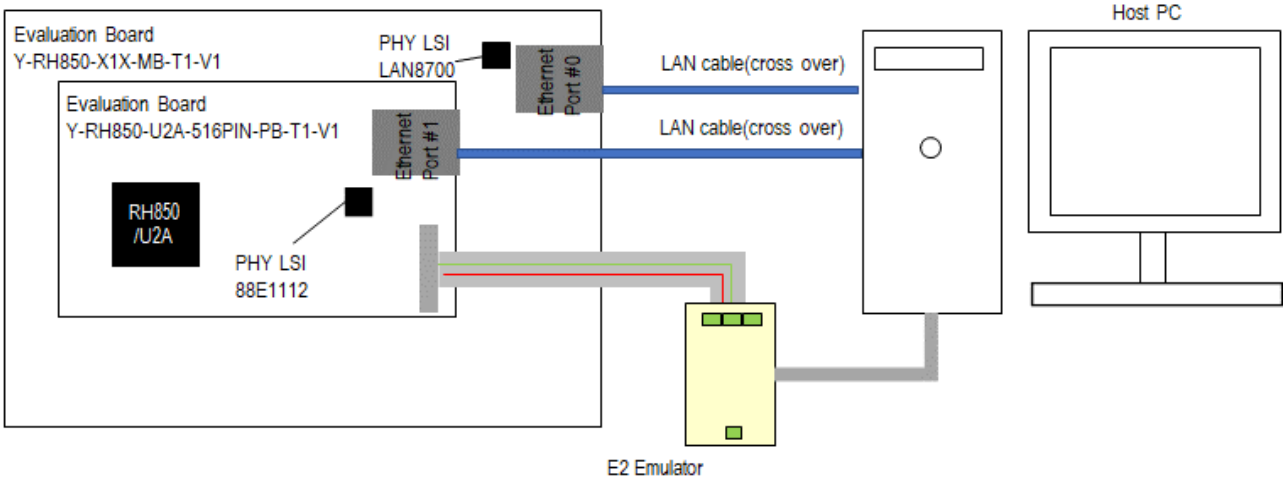

Figure 2-2 operation environment of sample code

2.3 Example of connection

This section indicates the connection operation of MII (100Base-TX) , SGMII(1000BASE-TX) communication, taking the sample code as an example.

If the connection was successful, the speed is indicated on the Ethernet Status window of Windows10 as Figure **2-3** shows.

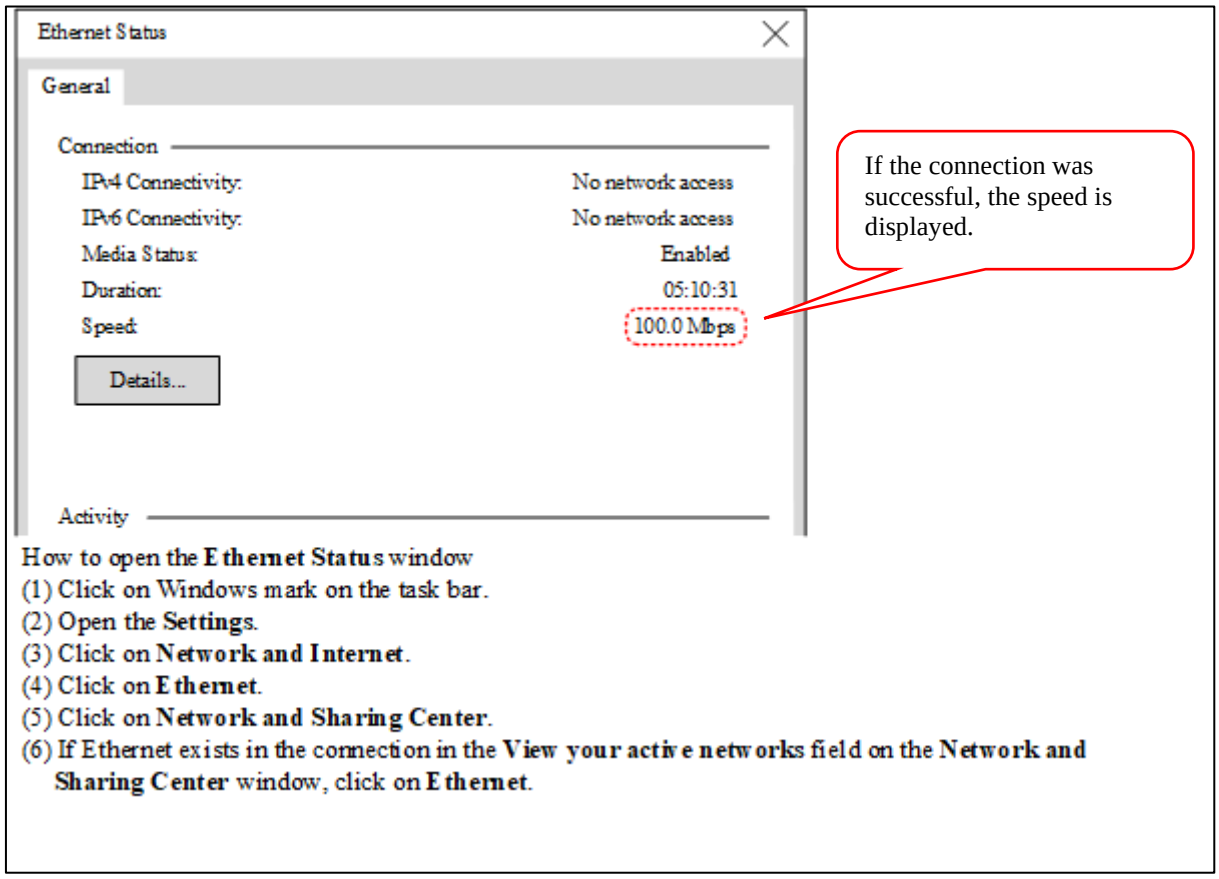

Figure 2-3 Successfully Connected Ethernet Status Window Example

3. Initialization Procedure

For the use of the Ethernet AVB (ETNB), initialization and settings of the E-MAC, the AVB-DMAC, and the descriptors are required.

Figure 3-1 shows the initialization processing, taking the sample code as an example.

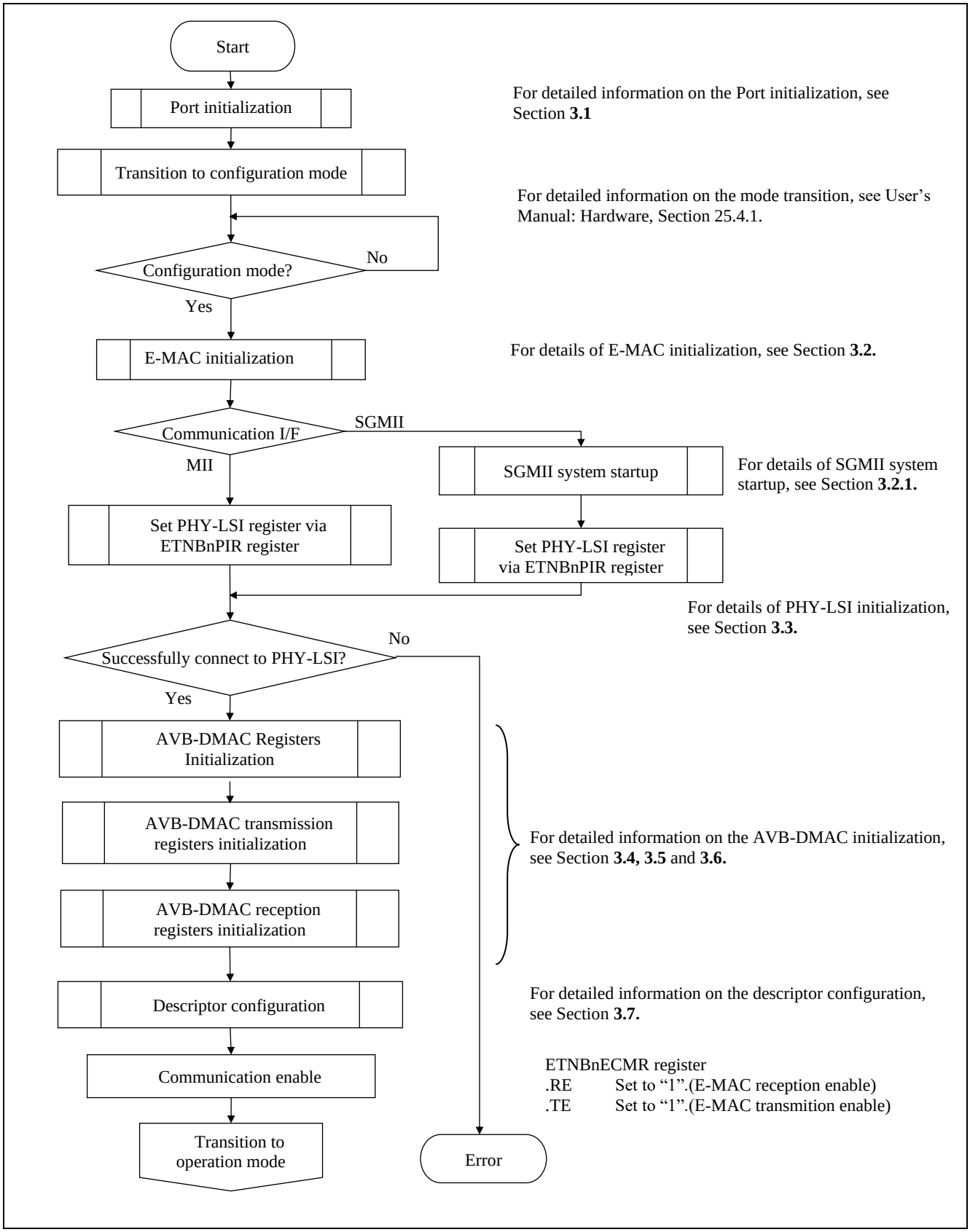

Figure 3-1 Ethernet AVB Initialization Procedure

3.1 Port Initialization Example

3.1.1 Case ETNB0(MII)

This section describes the port initialization for each signal connected in MII mode. This sample code assumes the use of the LAN8700 manufactured by Microchip Technology included on the evaluation board (Main Board): Y-RH850-X1X-MB-T1-V1.**Figure 3-2** shows the port for each signal used in MII mode.

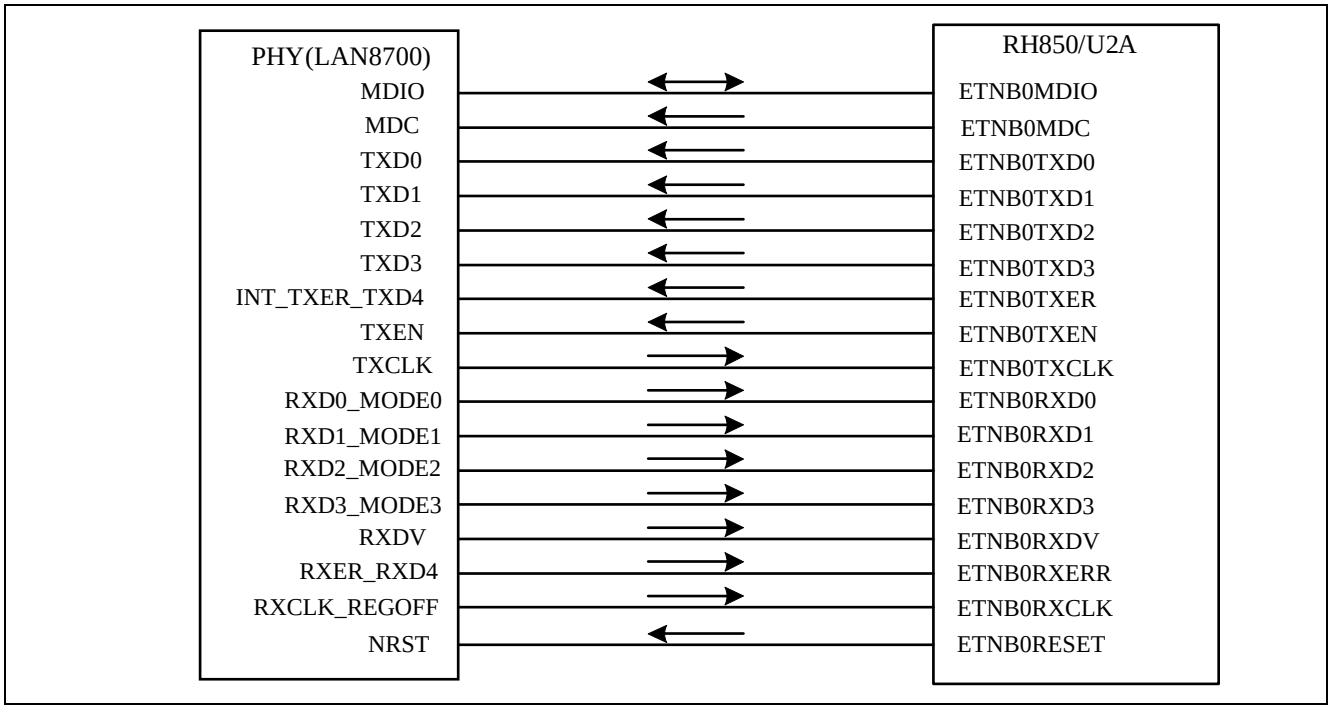

Figure 3-2 Signal Connection Example in MII mode

Table **3-1**,Table **3-2**, and Table **3-3** list examples of initialization for each port.

Table 3-1 ETNB0 Port Initialization Example [MII] (1/3)

Function (Signal) Name	Port Number	I/O	Description
ETNB0TXCLK	P10_1	I	Transmission clock : 25MHz clock at 100Mbps and 2.5MHz clock at 10Mbps are generated by the PHY.
			Input mode (PM10.PM10_1=1) Software I/O control (PIPC10.PIPC10_1=0) Alternative mode (PMC10.PMC10_1=1) Alternative mode 2 (PFCAE10.PFCAE10_1=0, PFCE10.PFCE10_1=0, PFC10.PFC10_1=1)
ETNB0TXEN	P20_14	O	Transmission permitted : Indicates that the effective transmission data is in TXD 0 to 4.
			Output mode (PM20.PM20_14=0) Software I/O control (PIPC20.PIPC20_14=0) Alternative mode (PMC20.PMC20_14=1) Alternative mode 2 (PFCAE20.PFCAE20_14=0, PFCE20.PFCE20_14=0, PFC20.PFC20_14=1)
ETNB0TXD0	P20_9	O	Transmission data bit 0 : One of the transmission data bits 0 to 3 permitted by Ethernet PHY
			Output mode (PM20.PM20_9=0) Software I/O control (PIPC20.PIPC20_9=0) Alternative mode (PMC20.PMC20_9=1) Alternative mode 2 (PFCAE20.PFCAE20_9=0, PFCE20.PFCE20_9=0, PFC20.PFC20_9=1) Port output buffer characteristics specified as HIGH (PDSC20.PDSC20_9=1)

Table 3-2 ETNB0 Port Initialization Example [MII] (2/3)

Function (Signal) Name	Port Number	I/O	Description
ETNB0TXD1	P20_10	O	Transmission data bit 1 : One of the transmission data bits 0 to 3 permitted by Ethernet PHY
			Output mode (PM20.PM20_10=0) Software I/O control (PIPC20.PIPC20_10=0) Alternative mode (PMC20.PMC20_10=1) Alternative mode 2 (PFCAE20.PFCAE20_10=0, PFCE20.PFCE20_10=0, PFC20.PFC20_10=1) Port output buffer characteristics specified as HIGH (PDSC20.PDSC20_10=1)
ETNB0TXD2	P20_12	O	Transmission data bit 2 : One of the transmission data bits 0 to 3 permitted by Ethernet PHY
			Output mode (PM20.PM20_12=0) Software I/O control (PIPC20.PIPC20_12=0) Alternative mode (PMC20.PMC20_12=1) Alternative mode 2 (PFCAE20.PFCAE20_12=0, PFCE20.PFCE20_12=0, PFC20.PFC20_12=1) Port output buffer characteristics specified as HIGH (PDSC20.PDSC20_12=1)
ETNB0TXD3	P20_13	O	Transmission data bit 3 : One of the transmission data bits 0 to 3 permitted by Ethernet PHY
			Output mode (PM20.PM20_13=0) Software I/O control (PIPC20.PIPC20_13=0) Alternative mode (PMC20.PMC20_13=1) Alternative mode 2 (PFCAE20.PFCAE20_13=0, PFCE20.PFCE20_13=0, PFC20.PFC20_13=1) Port output buffer characteristics specified as HIGH (PDSC20.PDSC20_13=1)
ETNB0TXER	P20_8	O	Transmission error : The case of 10Base-T is ignored.
			Output mode (PM20.PM20_8=0) Software I/O control (PIPC20.PIPC20_8=0) Alternative mode (PMC20.PMC20_8=1) Alternative mode 2 (PFCAE20.PFCAE20_8=0, PFCE20.PFCE20_8=0, PFC20.PFC20_8=1)
ETNB0RXCLK	P10_2	I	Reception clock : 25MHz clock at 100Mbps and 2.5MHz clock at 10Mbps are generated by the PHY.
			Input mode (PM10.PM10_2=1) Software I/O control (PIPC10.PIPC10_2=0) Alternative mode (PMC10.PMC10_2=1) Alternative mode 2 (PFCAE10.PFCAE10_2=0, PFCE10.PFCE10_2=0, PFC10.PFC10_2=1)
ETNB0RXDV	P10_7	I	Reception Data Enable : Indicates that the decoded reception data lies in RXD 0 to 3.
			Input mode (PM10.PM10_7=1) Software I/O control (PIPC10.PIPC10_7=0) Alternative mode (PMC10.PMC10_7=1) Alternative mode 2 (PFCAE10.PFCAE10_7=0, PFCE10.PFCE10_7=0, PFC10.PFC10_7=1)
ETNB0RXD0	P10_3	I	Received data bit 0 : One of the reception data bits 0 to 3 permitted by Ethernet PHY
			Input mode (PM10.PM10_3=1) Software I/O control (PIPC10.PIPC10_3=0) Alternative mode (PMC10.PMC10_3=1) Alternative mode 2 (PFCAE10.PFCAE10_3=0, PFCE10.PFCE10_3=0, PFC10.PFC10_3=1)

Table 3-3 ETNB0 Port Initialization Example [MII] (3/3)

Function (Signal) Name	Port Number	I/O	Description
ETNB0RXD1	P10_4	I	Received data bit 1 : One of the reception data bits 0 to 3 permitted by Ethernet PHY
			Input mode (PM10.PM10_4=1) Software I/O control (PIPC10.PIPC10_4=0) Alternative mode (PMC10.PMC10_4=1) Alternative mode 2 (PFCAE10.PFCAE10_4=0, PFCE10.PFCE10_4=0, PFC10.PFC10_4=1)
ETNB0RXD2	P10_5	I	Received data bit 2 : One of the reception data bits 0 to 3 permitted by Ethernet PHY
			Input mode (PM10.PM10_5=1) Software I/O control (PIPC10.PIPC10_5=0) Alternative mode (PMC10.PMC10_5=1) Alternative mode 2 (PFCAE10.PFCAE10_5=0, PFCE10.PFCE10_5=0, PFC10.PFC10_5=1)
ETNB0RXD3	P10_6	I	Received data bit 3 : One of the reception data bits 0 to 3 permitted by Ethernet PHY
			Input mode (PM10.PM10_6=1) Software I/O control (PIPC10.PIPC10_6=0) Alternative mode (PMC10.PMC10_6=1) Alternative mode 2 (PFCAE10.PFCAE10_6=0, PFCE10.PFCE10_6=0, PFC10.PFC10_6=1)
ETNB0RXER	P10_0	I	Reception error : Indicates that an error was detected in frame being transmitted from Ethernet PHY.
			Input mode (PM10.PM10_0=1) Software I/O control (PIPC10.PIPC10_0=0) Alternative mode (PMC10.PMC10_0=1) Alternative mode 2 (PFCAE10.PFCAE10_0=0, PFCE10.PFCE10_0=0, PFC10.PFC10_0=1)
ETNB0MDC	P20_6	I	Management clock : For the serial interface
			Output mode (PM20.PM20_6=0) Software I/O control (PIPC20.PIPC20_6=0) Alternative mode (PMC20.PMC20_6=1) Alternative mode 2 (PFCAE20.PFCAE20_6=0, PFCE20.PFCE20_6=0, PFC20.PFC20_6=1)
ETNB0MDIO	P20_3	I/O	Management data I/O : I/O of Serial interface data
			Direct I/O control (PIPC20.PIPC20_3=1) Alternative mode (PMC20.PMC20_3=1) Alternative mode 2 (PFCAE20.PFCAE20_3=0, PFCE20.PFCE20_3=0, PFC20.PFC20_3=1)
ETNB0RESET	P20_0	O	External Reset : Pins for External Reset
			Output mode (PM20.PM20_0=0) Port mode (PMC20.PMC20_14=0)

3.1.2 Case ETNB1(SGMII)

This section describes the port initialization for each signal connected in SGMII mode. This sample code assumes the use of the PHY-device 88E1112 manufactured by Marvell Semiconductor included on the evaluation board (Piggy Back Board): Y-RH850-U2A-516PIN-PB-T1-V1.

Figure **3-3** shows the port for each signal used in SGMII mode.

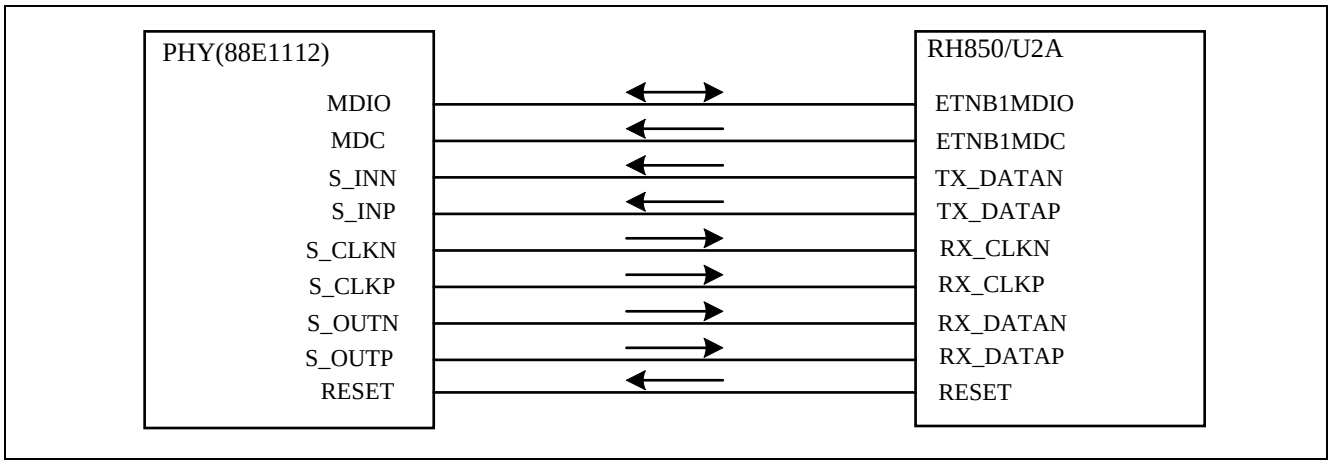

Figure 3-3 Signal Connection Example in SGMII mode

Table **3-4** lists initialization examples of each port.

Table 3-4 ETNB1 Port Initialization Example [SGMII]

Function (Signal) Name	Port Number	I/O	Function / Description
TX_DATAN	--------	O	SGMII transmit serial data outputs (Negative) Needs no port settings. It has a dedicated pin.
TX_DATAP	--------	O	SGMII transmit serial data outputs (Positive) *2 Needs no port settings. It has a dedicated pin. *2
RX_CLKN	--------	I	 Needs no port settings. It has a dedicated pin.
RX_CLKP	--------	I	 Needs no port settings. It has a dedicated pin.
RX_DATAN	--------	I	SGMII transmit serial data inputs (Negative) Needs no port settings. It has a dedicated pin. *2
RX_DATAP	--------	I	SGMII transmit serial data inputs (Positive) Needs no port settings. It has a dedicated pin. *2
REFCLK	--------	I	SGMII Reference clock input Needs no port settings. It has a dedicated pin.
ETNB1MDC	P3_6	O	Management Lock Output mode (PM3.PM3_6=0) Software I/O control (PIPC3.PIPC3_6=0) Alternative mode (PMC3.PMC3_6=1) Alternative mode 3 (PFCAE6.PFCAE3_6=0, PFCE6.PFCE3_6=1, PFC6.PFC3_6=0)
ETNB1MDIO	P3_7	I/O	Management data I/O Output mode (PM3.PM3_7=0) Software I/O control (PIPC3.PIPC3_7=1) Alternative mode (PMC3.PMC3_7=1) Alternative mode 3 (PFCAE3.PFCAE3_7=0, PFCE3.PFCE3_7=1, PFC3.PFC3_7=0)
RESET	P9_0	O	External Reset Output mode (PM9.PM9_0=0) Port mode (PMC9.PMC9_0=0)

*2 SGMII transmitter polarity inversion is defined OPBT14. Details of polarity refer to U2A-EVA Group User's Manual Hardware : 51.12.20 "OPBT 14".

3.2 E-MAC Initialization Example

Figure **3-4** shows the flow of the E-MAC initialization, taking the sample code as an example.

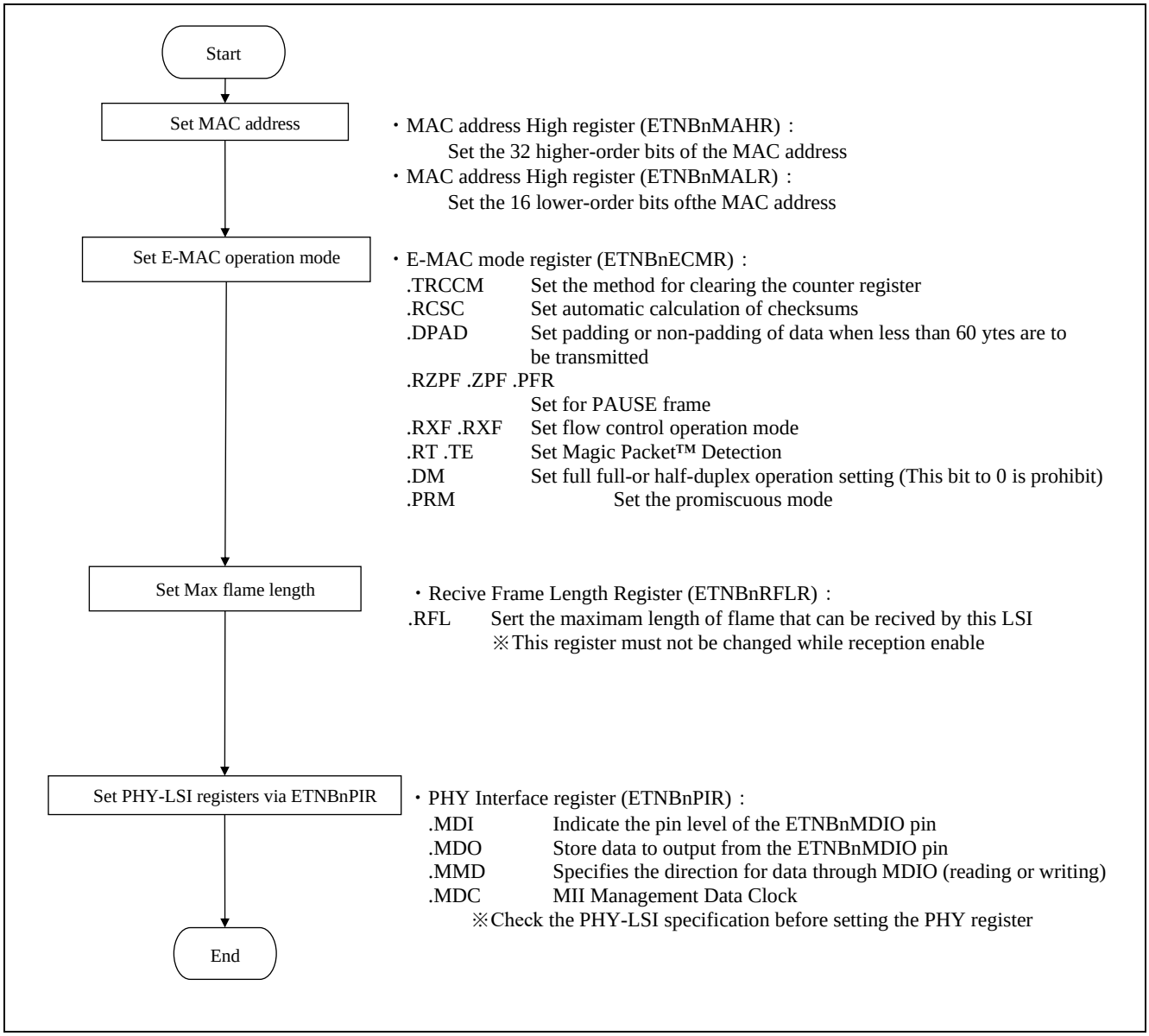

Figure 3-4 Example of E-MAC Initialization Flow

Table **3-5** lists examples of the register settings for the E-MAC initialization.

Table 3-5 Example of Register Settings during E-MAC Initialization

Register Name	Setting Value	Function
ETNBnMAHR	0x00800F00	The 32 higher-order bits of the MAC address (00:80:0F:00)
ETNBnMALR	0x00000001	The 16 lower-order bits of the MAC address (00:01)
ETNBnECMR	0x00200003	Counter Clear Mode : Select 0 CLEAR when writing
		Checksum Calculation : Select the check sum automatic calculation
		Disable PAUSE Frame Receive with TIME=0
		Disable PAUSE Frame Transmit and Response with TIME=0
		Data padding : Select Data Padding Disabled.
		Mode selection : Select Full-duplex Mode
		Promiscuous Mode : Select Promiscuous Mode Operation
ETNBnRFLR.	0x000005F9	Receive Frame Data Length (1,529 bytes)
ETNBnECSIPR	0x00000000	Disable E-MAC interrupt

3.2.1 SGMII System startup

Figure **3-5** show the E-MAC initialization flow, taking the sample code as an example.

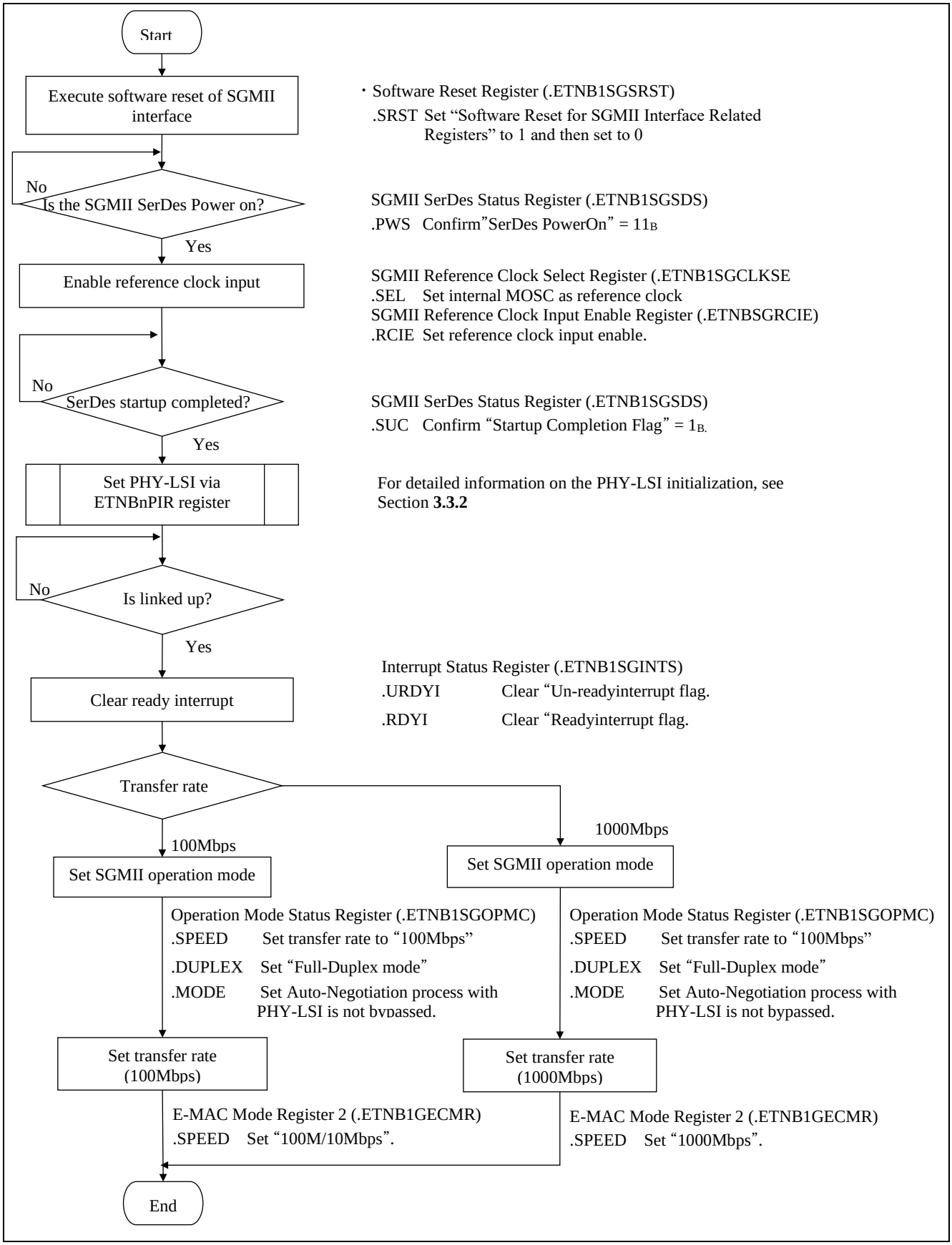

Figure 3-5 Example of E-MAC Initialization Flow [SGMII]

Table **3-6** lists examples of the register settings for each function during the SGMII initialization.

Table 3-6 Example of Register Settings for E-MAC Initialization [SGMII]

Register Name	Setting Value	Function
ETNB1SGSRST	0x00000000	Reset Release
ETNB1SGCLKSEL[*3]	0x01	External clock (20MHz)
ETNB1SGRCIE	0x01	Reference Clock Input Enable
ETNB1SGINTS	0x00000000	No Un-ready Interrupt (The status is cleared.)
		No Ready Interrupt (The status is cleared.)
ETNB1SGOPMC	0x0000000B (1Gbps communication)	Transfer Rate =1000Mbps
		Full-duplex mode
		Auto-Negotiation process with PHY-LSI is bypassed. (Disable)
	0x00000007 (100Mbps communication)	Transfer Rate =100Mbps
		Full-duplex mode
		Auto-Negotiation process with PHY-LSI is bypassed. (Disable)
ETNB1GECMR	0x00000001 (1Gbps communication)	Transfer Rate Setting =1000Mbps
	0x00000000 (100Mbps communication)	Transfer Rate Setting =100Mbps
ETNB1ECMR	0x00000002	Mode selection: Select full-duplex mode

[*3] If the MOSC is used for reference clock source of SGMII, it has to be attached 20MHz crystal.

3.3 PHY-LSI Initialization Example

Ethernet PHY registers are accessed via the ETNBnPIR register. ETNBnPIR is used as a serial interface conforming to the MII frame format specified in IEEE802.3u.

The format of an MII management frame refer to U2A-EVA Group User's Manual Hardware : "25.4.13.2 MII Management Frame Format". Figure **3-6** and Figure **3-7** show the frame structure and the timing.

The program accesses MII registers via ETNBnPIR. Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. The MII register access timing refer to U2A-EVA Group User's Manual Hardware : "25.4.13.3 MII Register Access Procedure". (The timing will differ depending on the PHY-LSI type.)

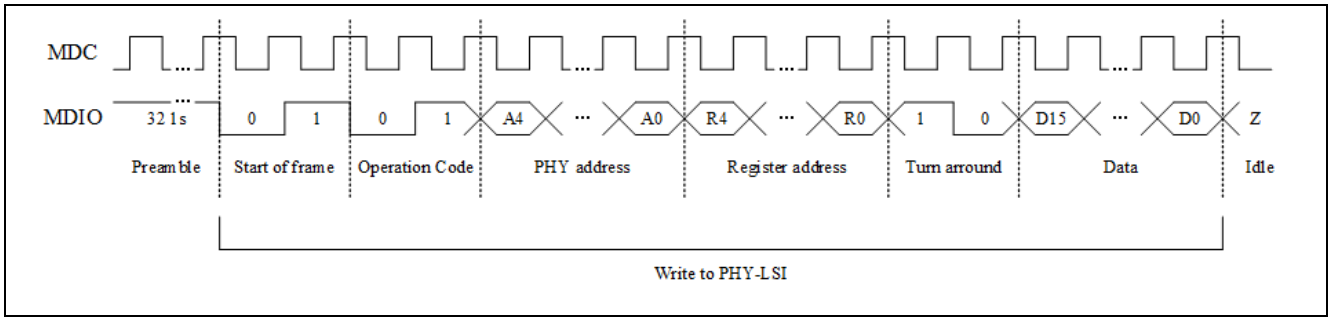

Figure 3-6 MDIO Timing and Frame Structure (Wtite Cycle)

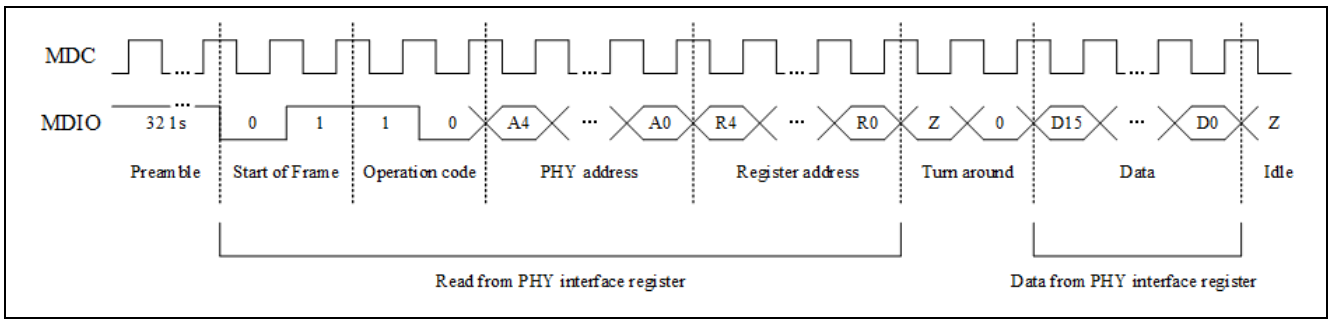

Figure 3-7 MDIO Timing and Frame Structure (Read Cycle**)**

3.3.1 Case ETNB0(MII)

The PHY-LSI should be initialized corresponding to the PHY. This sample code assumes the use of the PHY device LAN8700 manufactured by Microchip Technology included on the evaluation board (Main Board): Y-RH850-X1X-MB-T1-V1. **Figure 3-8** shows an example of the PHY-LSI (LAN8700 MII) initialization.

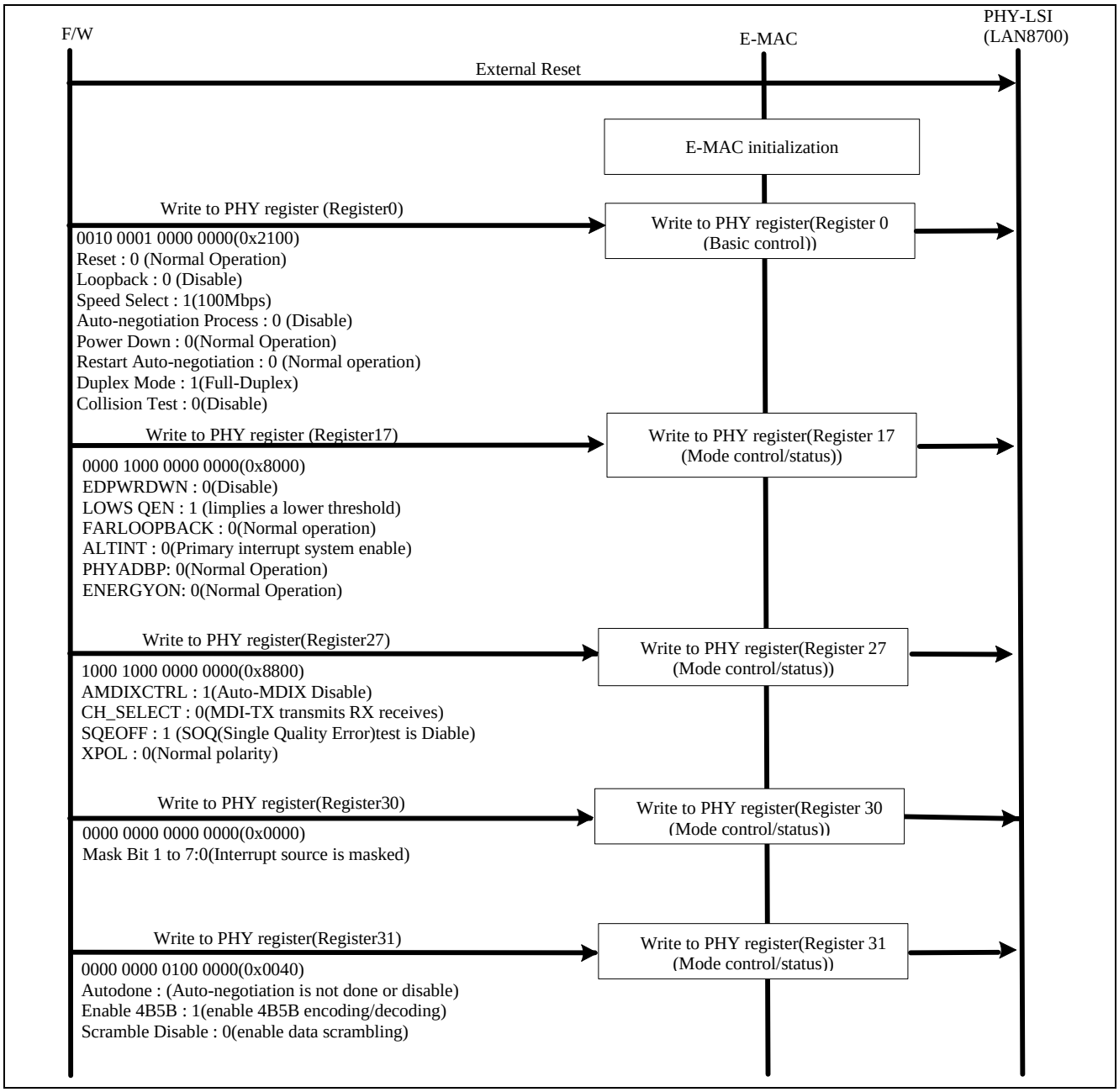

Figure 3-8 Example of PHY-LSI (LAN8700 MII) Initialization Sequence

3.3.2 Case ETNB1(SGMII)

The PHY-LSI should be initialized corresponding to the PHY device. This sample code assumes the use of the PHY device 88E1112 manufactured by Marvell Semiconductor included on the evaluation board (Piggy Back Board): Y-RH850-U2A-516PIN-PB-T1-V1. Figure **3-9** shows the PHY-LSI initialization.

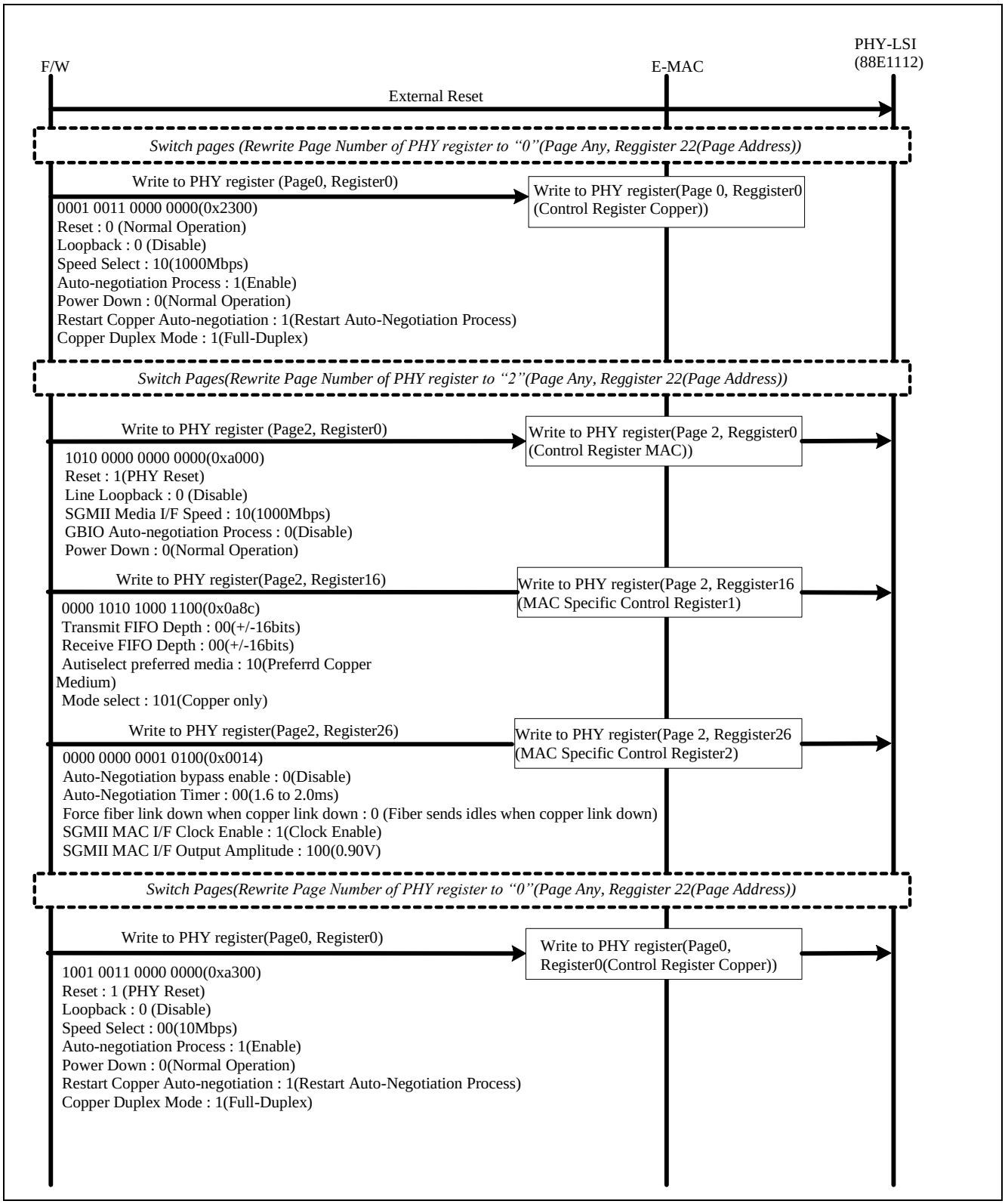

Figure 3-9 Sequence of PHY-LSI (88E1112 SGMII) Initialization Example

3.4 Example of AVB-DMAC Registers Initialization

Figure **3-10** shows the initialization flow of the AVB-DMAC registers of the sample code.

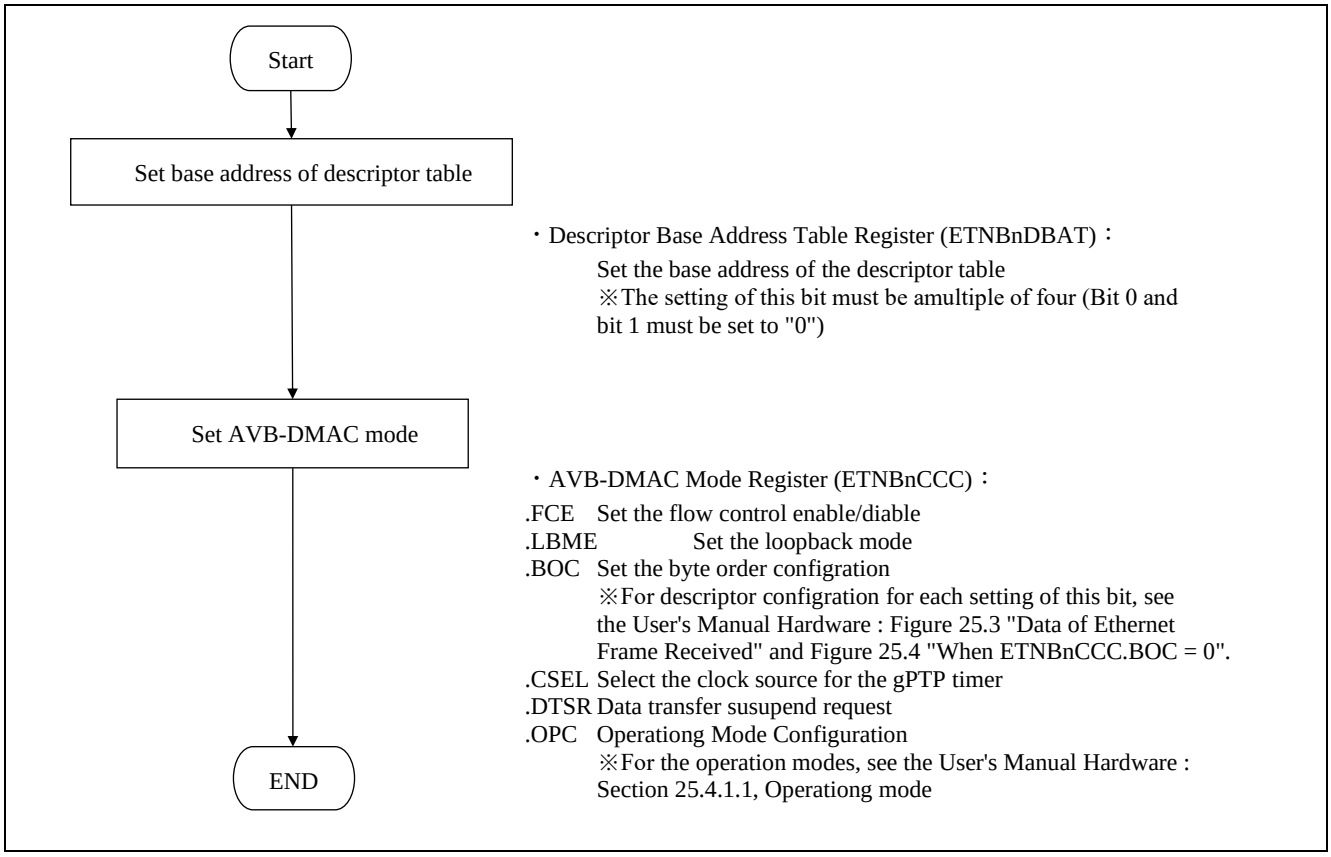

Figure 3-10 Example of AVB-DMAC Registers Initialization Flow

Table **3-7** lists examples of the register settings for the AVB-DMAC registers initialization.

Table 3-7 Example of AVB-DMAC Register Settings during Initialization

Register Name	Setting Value	Function
ETNBnDBAT	_ETNB_TEST_DBAT_TOP	Set the descriptor base address (URAM address)
ETNBnCCC	0x01000001	Loopback Mode enable
		The byte order is set to Big Endian.
		Set configuration mode

3.5 Example of Initialization of AVB-DMAC Transmission registers

Figure **3-11** shows the initialization flow of the AVB-DMAC transmission registers of the sample code.

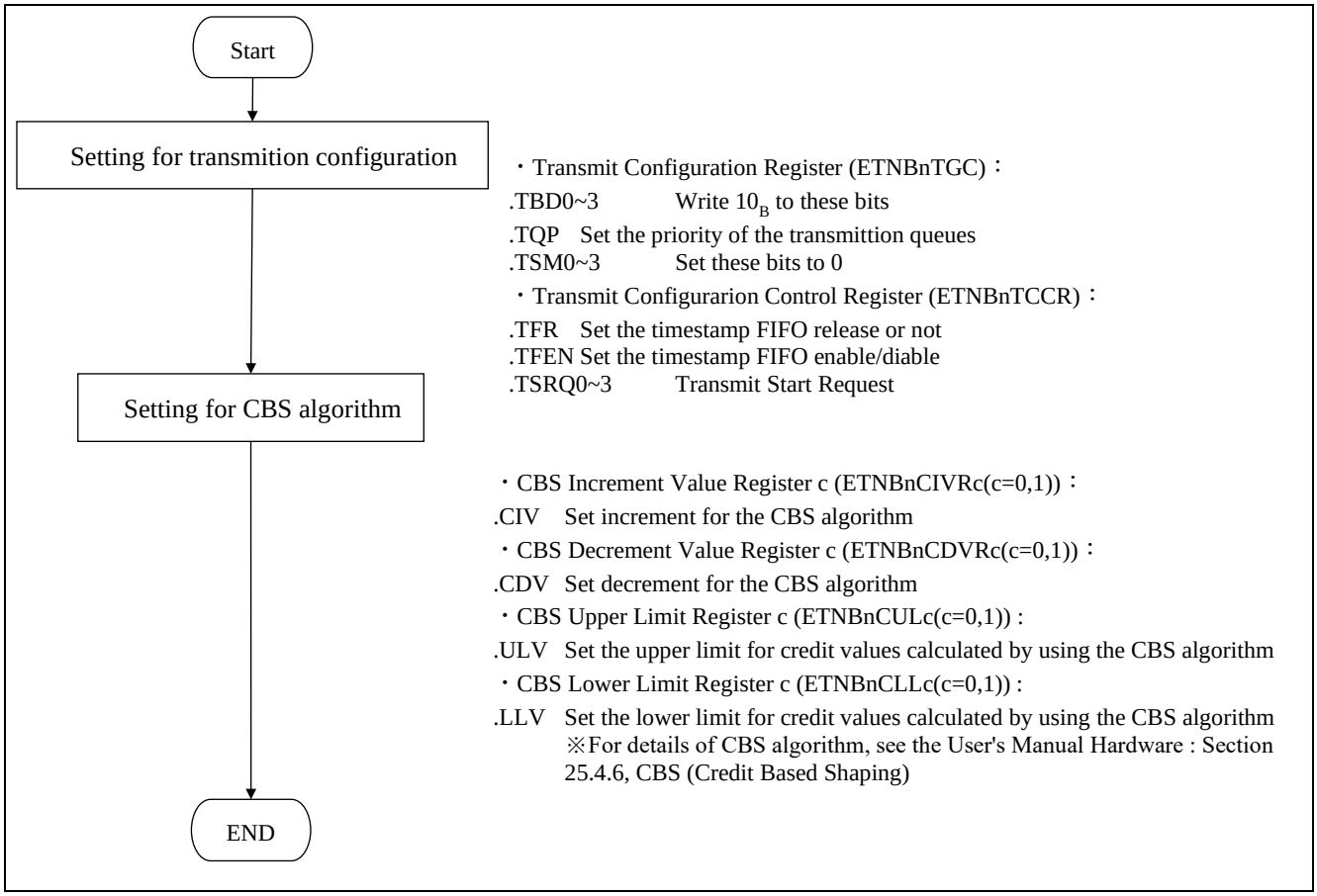

Figure 3-11 Example of Initialization Flow of AVB-DMAC transmission registers

Table **3-8** lists examples of the register setting for AVB-DMAC transmission registers initialization.

Table 3-8 Examples of AVB-DMAC Transmission Register Settings during Initialization

Register Name	Setting Value	Function
ETNBnTGC	0x00222200	Transmit Queue Priority is set to non-AVB mode.
ETNBnTCCR	0x00000000	No operation of Time Stamp FIFO
ETNBnCIVRc (c=0, 1)	0x00000001	The CBS increment value is set to 1.
ETNBnCDVRc (c=0, 1)	0xFFFFFFFF	The CBS decrement value is set to -1.
ETNBnCULc (c=0, 1)	0x7FFFFFFF	Set the maximum value of the CBS
ETNBnCLLc (c=0, 1)	0x800000001	Set the minimum value of the CBS

3.6 Example of AVB-DMAC Reception Registers Initialization

Figure **3-12** shows the initialization flow of the AVB-DMAC reception registers of the sample code.

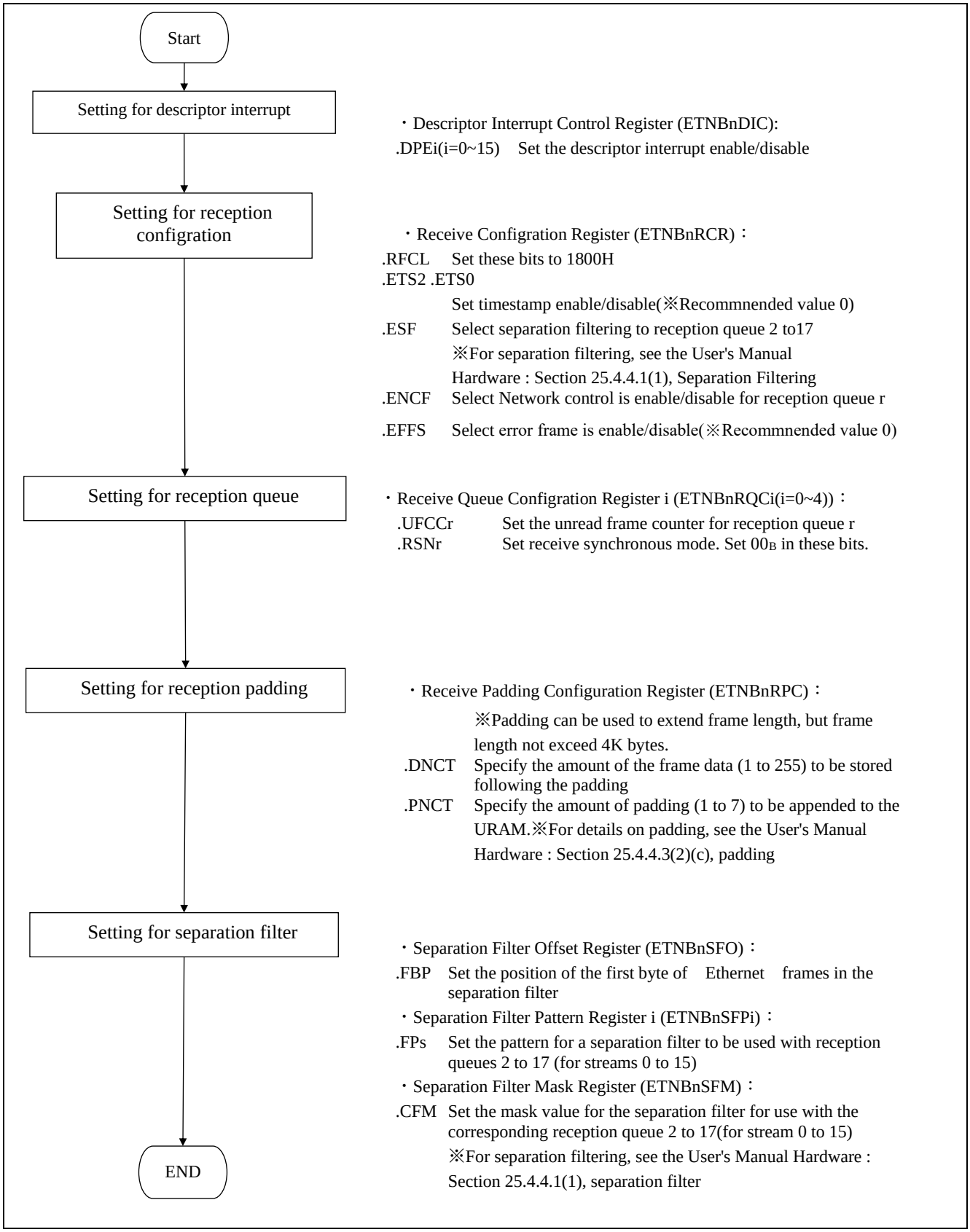

Figure 3-12 Example of Initialization Flow of AVB-DMAC reception registers

Table **3-9** lists examples of the settings for the AVB-DMAC reception registers initialization.

Table 3-9 Example of AVB-DMAC Reception Register Settings during Initialization

Register Name	Setting Value	Function
ETNBnDIC	0x000000F0	Enable Descriptor interrupt 7 to 4
ETNBnRCR	0x18000005	Set Receive FIFO Warning Level to 0x1800 (Recommended value)
		Streaming filter setting (01b)
		Error frame enable
ETNBnRQCi (i=0～4)	0x00000000	Receive queue setting (Unspecified)
ETNBnRPC	0x00BC0400	Amount of stored frame data (188)
		Amount of Padding (4)
ETNBnSFO	0x00000000	Separation filter offset value (Unspecified)
ETNBnSFPi (i=0)	0xBBBBBBBB	Separation filter pattern of the receive queue 2 (Stream0)
ETNBnSFPi (i=1)	0x0000BBBB	Separation filter pattern of the receive queue 2 (Stream0)
ETNBnSFPi (i=2～31)	0x00000000	Separation filter pattern of the receive queue 3 to 17 (Stream 1 to15) (Unspecified)
ETNBnSFM0	0xFFFFFFFF	Separation filter mask value
ETNBnSFM1	0x0000FFFF	Separation filter mask value

3.7 Descriptor Configuration Examples

Figure **3-13** shows the configuration example of the reception descriptors of the sample code. Figure **3-14** and Figure **3-15** show the sample code. Regarding this sample code, the descriptors that control the four reception queues (4 to 7) are constructed. The descriptors, whose data type (DT) is FEMPT and data size (DS) is 1516 bytes, will be prepared in all reception queues.

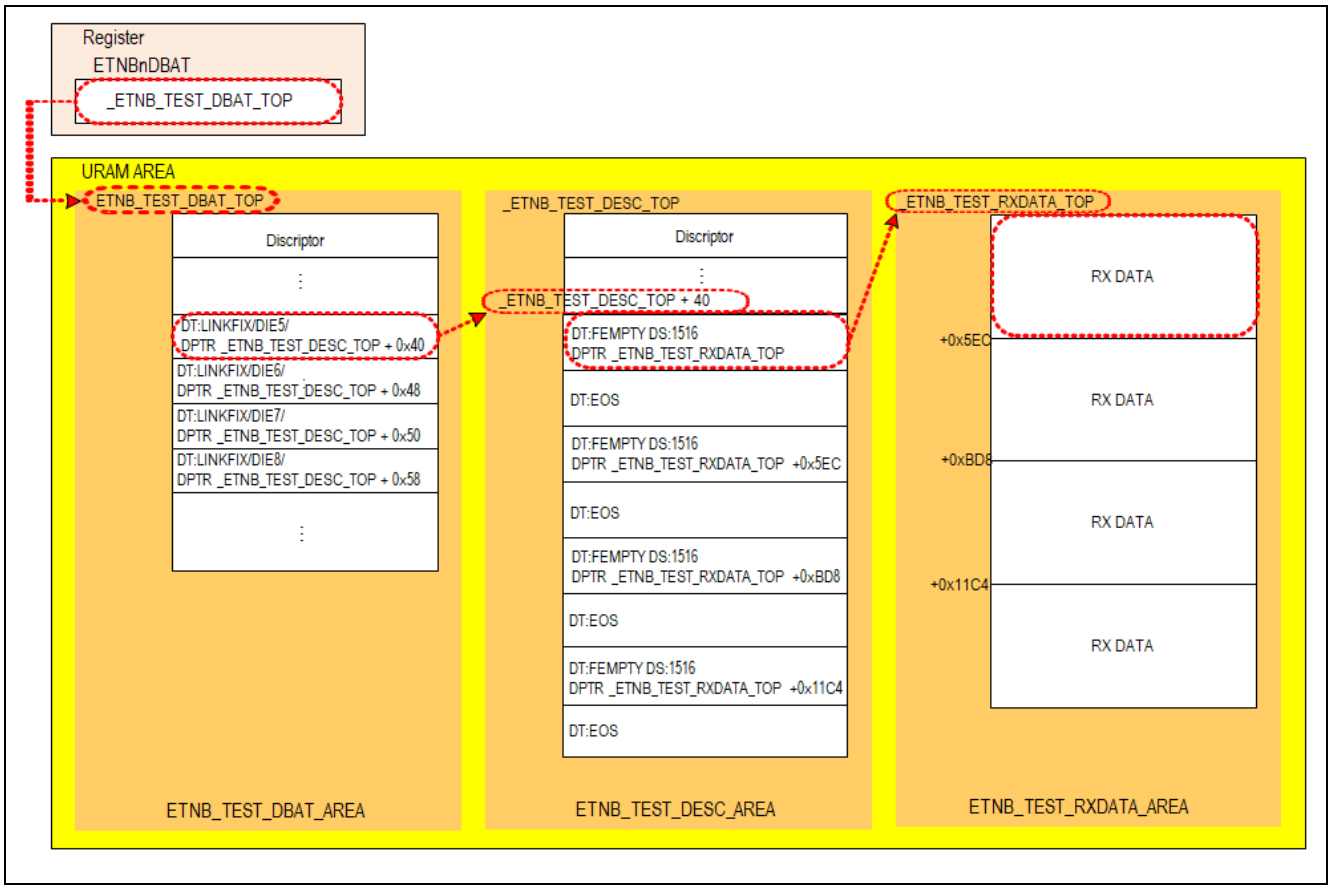

Figure 3-13 Configuration Example of Reception Descriptor of Sample Code

```
/* Normal descriptor for Rx (64 bit length) */
typedef struct
{
  UL            bDs : 12;      /* Data size [byte] .                                          */
  const UL      bTr :  1;      /* Truncated status bit.                                       */
  const UL      bEi :  1;      /* Error status bit.                                           */
  UL            bPs :  2;      /* Padding selection bit.                                      */
  UL            bMsc:  8;      /* MAC status code bit.                                        */
  UL            bDie:  4;      /* Descriptor interrupt number. If 0, interrupt does not occur. */
  UL            bDt :  4;      /* Descriptor type.                                            */
  UL            u4Dptr;        /* Descriptor data pointer (data is 32 bit alignment).         */
} EthDrv_stRxDescriptor;

        (Omission)

/* Descriptor type enumeration */
typedef enum {  /* DT:Rx:Tx: Description                                                     */
  enDT0 = 0,    /*  0:--:--: Reserved.                                                      */
  enDT1,        /*  1:--:--: Reserved.                                                      */
  enLEMPTY,     /*  2:SW:SW: After LINK descriptor has been done by AVB-DMAC.               */
  enEEMPTY,     /*  3:SW:SW: After EOS descriptor has been done by AVB-DMAC.                */
  enFMID,       /*  4:SW:HW: Data descriptor (middle area).                                 */
  enFSTART,     /*  5:SW:HW: Data descriptor (start area).                                  */
  enFEND,       /*  6:SW:HW: Data descriptor (end area).                                    */
  enFSINGLE,    /*  7:SW:HW: Data descriptor (single data).                                 */
  enLINK,       /*  8:HW:HW: Point next chain descriptor (changed by AVB-DMAC).             */
  enLINKFIX,    /*  9:SW:SW: Point next chain descriptor (not changed).                     */
  enEOS,        /* 10:HW:HW: Point next chain descriptor (not changed).                     */
  enDT11,       /* 11:--:--: Reserved.                                                      */
  enFEMPTY,     /* 12:HW:SW: There is no valid frame data.                                  */
  enFEMPTY_IS,  /* 13:HW:--: There is no valid frame data (DPTR points incremental base).   */
  enFEMPTY_IC,  /* 14:HW:--: There is no valid frame data (data is stored incremental area). */
  enFEMPTY_ND,  /* 15:HW:--: There is no valid frame data (data is not stored).             */
} EthDrv_enDescDataType;

        (Omission)
```

Figure 3-14 Configuration Processing Example of Reception Descriptor of Sample Code (1/2)

```
/*****************************************************************************
Imported global variables and functions (from other files)
*****************************************************************************/
UL      ETNB_TEST_DBAT_TOP;
UL      ETNB_TEST_DESC_TOP;

        (Omission)

UL      ETNB_TEST_RXDATA_TOP;

        (Omission)

static void loc_CreateDescriptor( void )
{
  UB    *pttBaseDescPtr;
  UB    *pttDataDescPtr;

        (Omission)

  UB    *pttRxDataPtr;
  UL    u4tQueNo;
  UL    u4tDataIdx;

  pttBaseDescPtr = (UB*)&ETNB_TEST_DBAT_TOP;
  pttDataDescPtr = (UB*)&ETNB_TEST_DESC_TOP;

        (Omission)

  /*---- Rx descriptor configration ----*/
  pttRxDataPtr = (UB*)&ETNB_TEST_RXDATA_TOP;    /* Set top address of data area. */

  /* Entry 4-21 is assigned rx queue. # only 4-8 is initialized in this program */
  for( u4tQueNo = (UL)4 ; u4tQueNo < (UL)8 ; u4tQueNo++ )
  {
    /* Descriptor area initialization */
    ((EthDrv_stRxDescriptor*)pttBaseDescPtr)->bDie    = u4tQueNo; /* Descriptor interrupt number.*/
    ((EthDrv_stRxDescriptor*)pttBaseDescPtr)->bDt     = enLINKFIX;
    ((EthDrv_stRxDescriptor*)pttBaseDescPtr)->u4Dptr  = (UL)pttDataDescPtr;
    ((EthDrv_stRxDescriptor*)pttDataDescPtr)->bDs     = 1516U;
    ((EthDrv_stRxDescriptor*)pttDataDescPtr)->bPs     = 0U;     /* No padding. */
    ((EthDrv_stRxDescriptor*)pttDataDescPtr)->bMsc    = 0U;     /* Initialize MAC status code. */
    ((EthDrv_stRxDescriptor*)pttDataDescPtr)->bDie    = 0U;     /* Descriptor interrupt number. */
    ((EthDrv_stRxDescriptor*)pttDataDescPtr)->bDt     = enFEMPTY;
    ((EthDrv_stRxDescriptor*)pttDataDescPtr)->u4Dptr  = (UL)pttRxDataPtr;
    pttDataDescPtr += sizeof(EthDrv_stRxDescriptor);

    ((EthDrv_stRxDescriptor*)pttDataDescPtr)->bDie    = 0U;     /* Descriptor interrupt number. */
    ((EthDrv_stRxDescriptor*)pttDataDescPtr)->bDt     = enEOS;
    pttDataDescPtr += sizeof(EthDrv_stRxDescriptor);

    /* Data area initialization */
    pttBaseDescPtr += sizeof(EthDrv_stRxDescriptor);
    for( u4tDataIdx = 0UL;u4tDataIdx < 0x5ECUL;u4tDataIdx++ )   /* Set default value in Rx data
area. */
    {
      pttRxDataPtr[u4tDataIdx] = 0xFFU;
    }

    pttRxDataPtr += 0x5ECUL;    /* Increment data ptr address. */
  }

  return;
}
```

Figure 3-15 Configuration Processing Example of Reception Descriptor of Sample Code [MII] (2/2)

Figure **3-16** shows the configuration example of the transmission descriptors of the sample code. Figure **3-17** and Figure **3-18** show the sample code. Regarding this sample code, the descriptors that control the four transmission queues (0 to 3) are constructed. The descriptors, whose data type (DT) is FSINGLE and data size (DS) is 1516 bytes, will be prepared for all transmission queues.

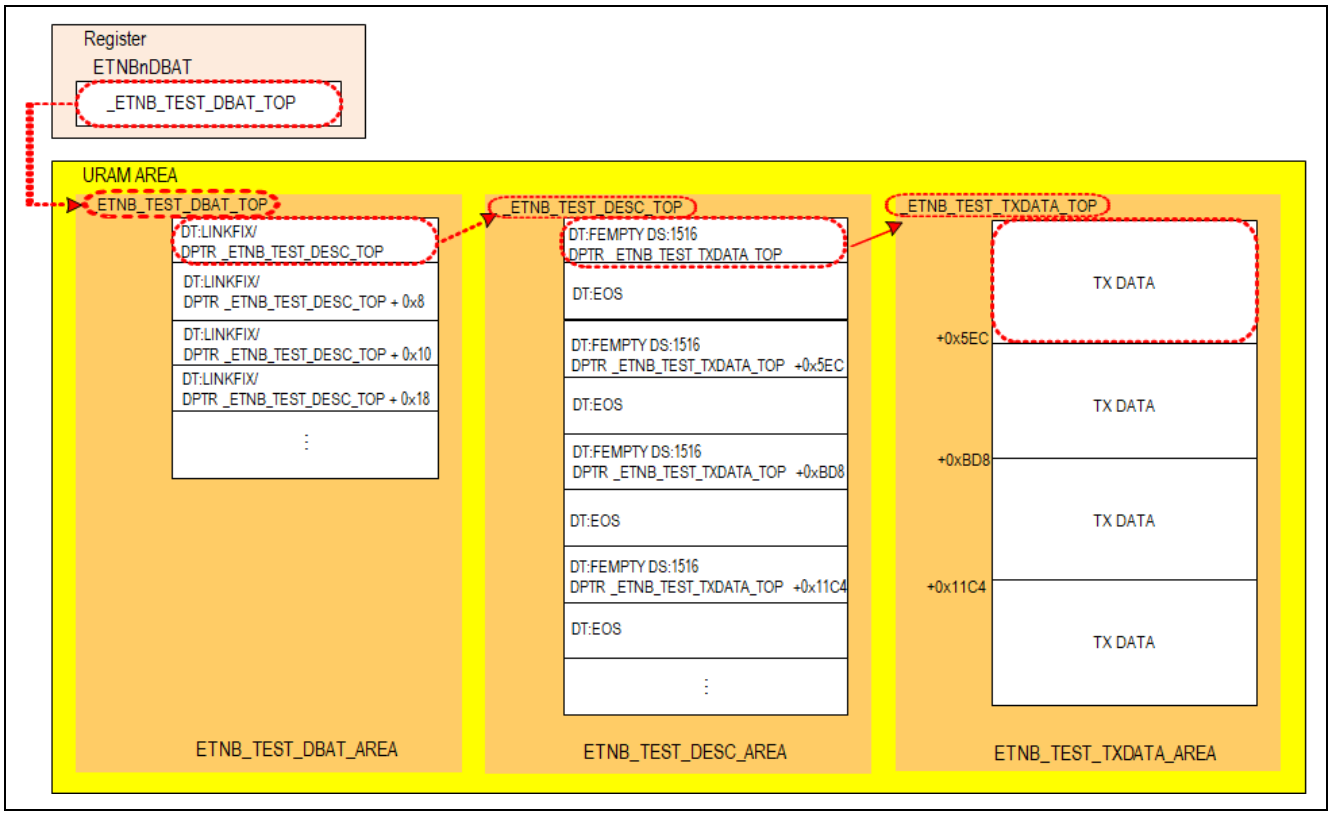

Figure 3-16 Configuration Example of Transmission Descriptor of Sample Code

```
/* Normal descriptor for Tx (64 bit length) */
typedef struct
{
  UL            bDs : 12;      /* Data size [byte] .                                    */
  UL            bTag: 10;      /* Frame tag related to timestamp.                         */
  UL            bTsr:  1;      /* Timestamp storing request bit.                          */
  UL            bMsr:  1;      /* MAC status storing request bit.                         */
  UL            bDie:  4;      /* Descriptor interrupt number. If 0, interrupt does not occur. */
  UL            bDt :  4;      /* Descriptor type.                                      */
  UL            u4Dptr;        /* Descriptor data pointer (data is 32 bit alignment).        */
} EthDrv_stTxDescriptor;

        (Omission)

/* Descriptor type enumeration */
typedef enum {  /* DT:Rx:Tx: Description                                               */
  enDT0 = 0,    /*  0:--:--: Reserved.                                                */
  enDT1,        /*  1:--:--: Reserved.                                                */
  enLEMPTY,     /*  2:SW:SW: After LINK descriptor has been done by AVB-DMAC.               */
  enEEMPTY,     /*  3:SW:SW: After EOS descriptor has been done by AVB-DMAC.                */
  enFMID,       /*  4:SW:HW: Data descriptor (middle area).                           */
  enFSTART,     /*  5:SW:HW: Data descriptor (start area).                            */
  enFEND,       /*  6:SW:HW: Data descriptor (end area).                              */
  enFSINGLE,    /*  7:SW:HW: Data descriptor (single data).                           */
  enLINK,       /*  8:HW:HW: Point next chain descriptor (changed by AVB-DMAC).             */
  enLINKFIX,    /*  9:SW:SW: Point next chain descriptor (not changed).                   */
  enEOS,        /* 10:HW:HW: Point next chain descriptor (not changed).                   */
  enDT11,       /* 11:--:--: Reserved.                                               */
  enFEMPTY,     /* 12:HW:SW: There is no valid frame data.                           */
  enFEMPTY_IS,  /* 13:HW:--: There is no valid frame data (DPTR points incremental base).     */
  enFEMPTY_IC,  /* 14:HW:--: There is no valid frame data (data is stored incremental area).   */
  enFEMPTY_ND,  /* 15:HW:--: There is no valid frame data (data is not stored).             */
} EthDrv_enDescDataType;

        (Omission)
```

Figure 3-17 Example of Configuration Processing of Transmission Descriptor of Sample Code (1/2)

```
/*****************************************************************************
Imported global variables and functions (from other files)
*****************************************************************************/
UL  ETNB_TEST_DBAT_TOP;
UL  ETNB_TEST_DESC_TOP;
UL  ETNB_TEST_TXDATA_TOP;

        (Omission)

static void loc_CreateDescriptor( void )
{
UB *pttBaseDescPtr;
  UB *pttDataDescPtr;
  UB *pttTxDataPtr;

        (Omission)

  UL  u4tQueNo;
  UL  u4tDataIdx;

  pttBaseDescPtr = (UB*)&ETNB_TEST_DBAT_TOP;
  pttDataDescPtr = (UB*)&ETNB_TEST_DESC_TOP;

   /*---- Tx descriptor initialization ----*/
  pttTxDataPtr = (UB*)&ETNB_TEST_TXDATA_TOP;               /* Set top address of data area.   */

  for( u4tQueNo = (UL)0 ; u4tQueNo < (UL)4 ; u4tQueNo++ )     /* Entry 0-3 is assigned tx queue. */
  {
    /* Descriptor area initialization */
    ((EthDrv_stTxDescriptor*)pttBaseDescPtr)->bDie    = 0U;   /* Descriptor interrupt number.    */
    ((EthDrv_stTxDescriptor*)pttBaseDescPtr)->bDt     = enLINKFIX;
    ((EthDrv_stTxDescriptor*)pttBaseDescPtr)->u4Dptr  = (UL)pttDataDescPtr;
    ((EthDrv_stTxDescriptor*)pttDataDescPtr)->bDs     = 1516U;
    ((EthDrv_stTxDescriptor*)pttDataDescPtr)->bTag    = 0U;   /* Frame tag related to timestamp. */
    ((EthDrv_stTxDescriptor*)pttDataDescPtr)->bTsr    = 0U;   /* Timestamp storing request.      */
    ((EthDrv_stTxDescriptor*)pttDataDescPtr)->bMsr    = 0U;   /* MAC status storing request.     */
    ((EthDrv_stTxDescriptor*)pttDataDescPtr)->bDie    = 0U;   /* Descriptor interrupt number.    */
    ((EthDrv_stTxDescriptor*)pttDataDescPtr)->bDt     = enFSINGLE;
    ((EthDrv_stTxDescriptor*)pttDataDescPtr)->u4Dptr  = (UL)pttTxDataPtr;
    pttDataDescPtr += sizeof(EthDrv_stTxDescriptor);
    ((EthDrv_stTxDescriptor*)pttDataDescPtr)->bDie    = 0U;   /* Descriptor interrupt number.    */
    ((EthDrv_stTxDescriptor*)pttDataDescPtr)->bDt     = enEOS;
    pttDataDescPtr += sizeof(EthDrv_stTxDescriptor);

    pttBaseDescPtr += sizeof(EthDrv_stTxDescriptor);  /* Increment descriptor base address. */
    /* Data area initialization */
    if( u4tQueNo == (UL)0 )
    {
      memcpy( pttTxDataPtr, &TxFrameHeader, sizeof(TxFrameHeader) );/* Frame header set.        */
      for( u4tDataIdx = sizeof(TxFrameHeader) ;u4tDataIdx < 0x5ECUL ;u4tDataIdx++ )
      {
        pttTxDataPtr[u4tDataIdx] = (UB)(u4tDataIdx + u4tQueNo);   /* Transmit data set.        */
      }
    }
    else
    {
      for( u4tDataIdx = 0 ;u4tDataIdx < 0x5ECUL ;u4tDataIdx++ )
      {
        pttTxDataPtr[u4tDataIdx] = (UB)(u4tDataIdx + u4tQueNo);    /* Transmit data set.          */
      }
    }
    pttTxDataPtr += 0x5ECUL;                                 /* Increment data ptr address.    */
  }

        (Omission)

  return;
}
```

Figure 3-18 Example of Configuration Processing of Transmission Descriptor of Sample Code (2/2)

4. Other configuration

4.1 Option Bytes (OPBT)

To use MII mode in the ETNB0, the setting of the mode selection bit of the OPBT14 (ETN_RMII_SEL) is required. (MII/RMII selection of the ETNB0)
Likewise, to use SGMII mode in the ETNB1, the setting of the mode selection bit of the OPBT14 (ETN_SGMII_SEL) is required. (MII/SGMII selection of the ETNB1)

4.2 Module Standby Registers

To enable the functions of the ETNB0 and the ETNB1, the setting of the module standby register (MSR_ETNB) is required. (The setting to supply clocks for functions) The initial value of this register is set to Disable in both cases of the ETNB0 and the ETNB1. (Neither the ETNB0 nor the ETNB1 is supplied with clocks.)

5. Appendix

5.1 Module List

The modules of sample code for the ETNB0 and the ETNB1 are listed below.
Table **5-1** lists the modules.

Table 5-1 List of Modules

Module Name	Label Name	Function
Main Routine	main_pm0	Settings, Application activation
PORT Initialization	R_PORT_Init	Port initialization
PHY-LSI Reset	R_ETNB_ResetPhy	Reset Ethernet PHY-LSI
E-MAC Initialization	R_ETNB_InitEmacRegs	E-MAC initialization
PHY-LSI Initialization	R_ETNB_InitPhyRegs	PHY-LSI initialization
AVB-DMAC Initialization	R_ETNB_InitDmacRegs	AVB-DMAC registers initialization
AVB-DMAC Transmission Registers Initialization	R_ETNB_InitRxRegs	AVB-DMAC transmission registers settings
AVB-DMAC Reception Registers Initialization	R_ETNB_InitTxRegs	AVB-DMAC reception registers settings
Descriptor Configuration	loc_CreateDescriptor	Configurate the descriptors

Revision History

Rev.	Date	Description	
		Page	Summary
1.0	2021.1.29	-	Isuued 1st version.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

——

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   - Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

- For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.