

RH850/U2A-EVA Group

Ethernet AVB IF 接続ガイド

要旨

本アプリケーションノートでは、RH850/U2A に搭載される Ethernet の PHY-LSI への接続処理および Ethernet AVB(ETNB)モジュールの初期設定について説明します。

本資料およびプログラムは、RH850/U2A 搭載機能の理解促進を意図するものであり、量産設計を対象とするものではありません。

また、最新のマニュアル、正誤表、テクニカルアップデートや、開発環境の更新を反映しておりません。該当機能を使用される場合には、本プログラムは参考として扱い、最新のドキュメントや開発環境にて、お客様の責任において行ってください。

対象デバイス

- RH850/U2A-EVA Group
 - RH850/U2A16 BGA516

対象統合開発環境

CS+(ルネサスエレクトロニクス社製)

バージョン : V8.03.00

デバイスファイル : R7F702300.DVF, V1.20

評価ボード

Main Board : Y-RH850-X1X-MB-T1-V1

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

参考文献

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

本アプリケーションノートは上記のマニュアルを参照し作成しております。

またデバイスの機能詳細及び電気的特性に関してはユーザーズマニュアル ハードウェア編に記載します。

目次

1. Ethernet AVB 概要.....	3
2. Ethernet AVB サンプルプログラム.....	7
3. Ethernet AVB 初期設定手順	9
4. その他の設定.....	30
5. 付録.....	31

1. Ethernet AVB 概要

1.1 Ethernet AVB(ETNB)モジュール概要

RH850/U2A は以下のユニット数の Ethernet AVB(ETNB)モジュールを搭載しています。

ETNB は 1 ユニットあたり 1 チャンネルの I/F を持っています。なお、ETNB0 は最大で 100Base-TX まで対応が可能なファストイーサネット用、ETNB1 は最大 1000Base-T まで対応可能なギガビットイーサネット用となります。

図 1-1 に ETNB ユニットのチャンネル構成を示します。

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.

図 1-1 ETNB ユニットのチャンネル構成

ETNB モジュールは以下の機能ユニットで構成されています。

- ・ MAC コントローラ (E-MAC)
- ・ DMA 転送コントローラ (AVB-DMAC)

図 1-2 に ETNB0 のブロック図、図 1-3 に ETNB1 のブロック図を示します。

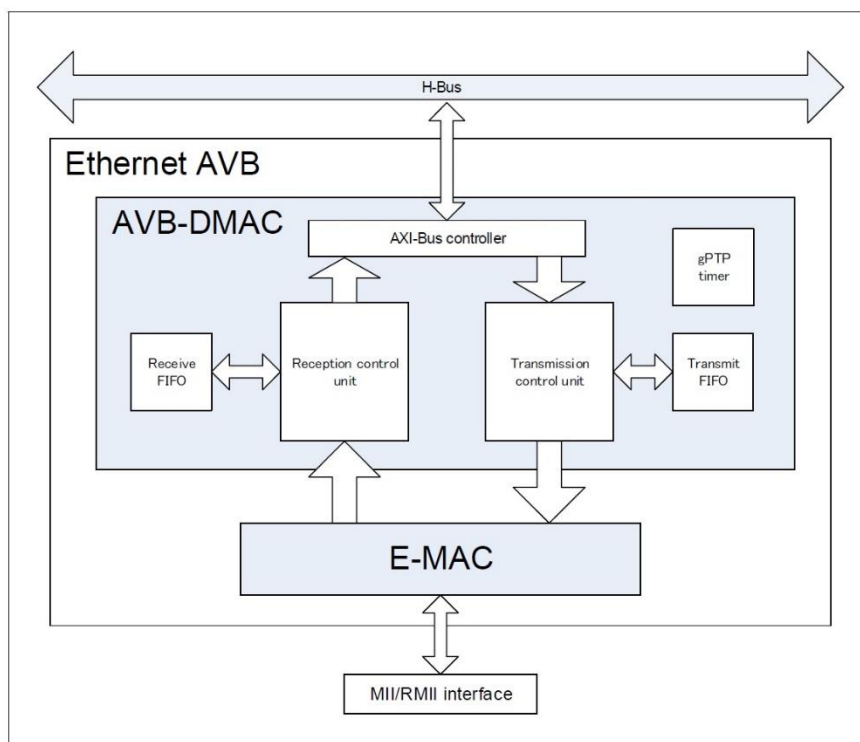


図 1-2 ETNB0 のブロック図

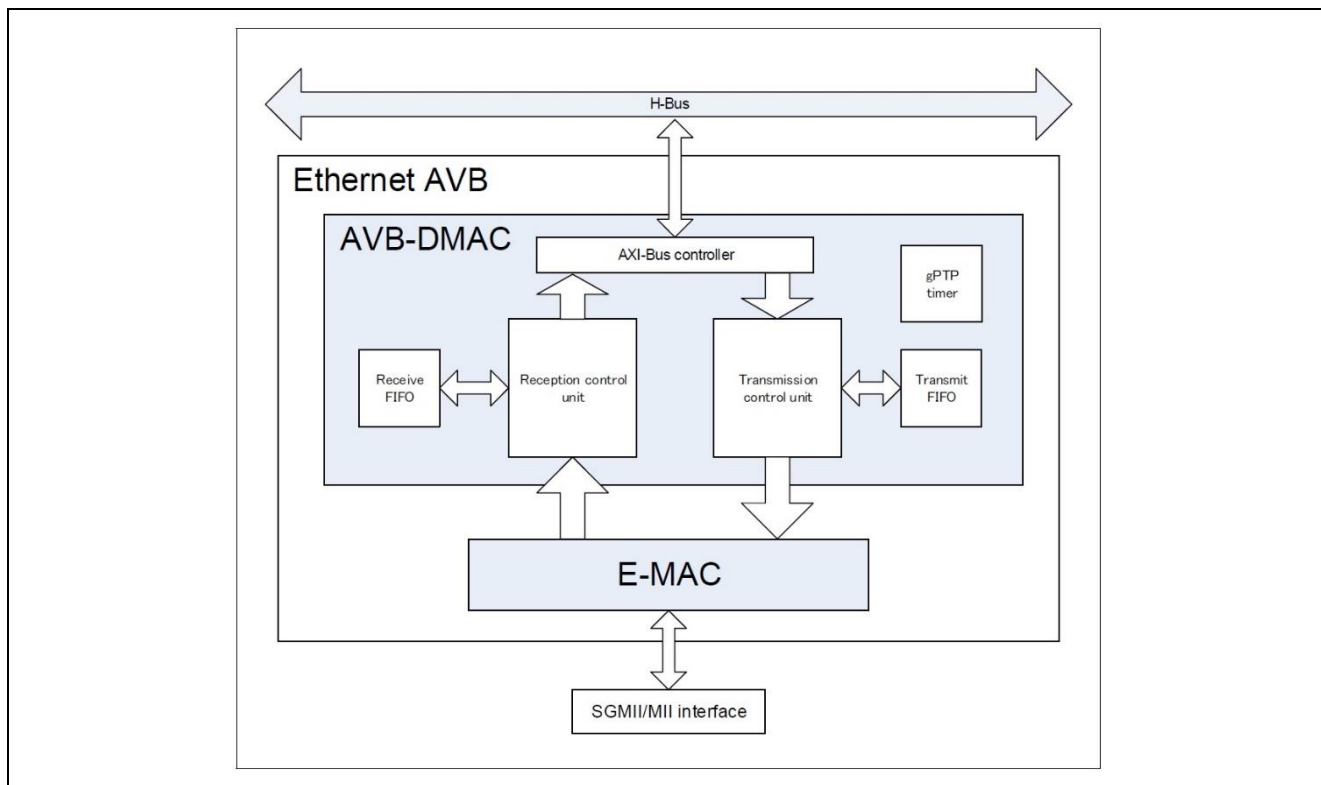


図 1-3 ETNB1 のブロック図

1.2 E-MAC の概要

E-MAC の構成を図 1-4 に示します。E-MAC は、外部接続される PHY-LSI との I/F フォーマットとして MII/RMII/SGMII をサポートします（ETNB0 は MII/RMII を、ETNB1 は MII/SGMII をサポート）。

E-MAC は、送信 FIFO に書き込まれたデータからイーサネットフレームを構築し、MII/RMII/SGMII へ送信します。

また、MII/RMII/SGMII から受信したイーサネットフレームの CRC チェックを実行し、受信 FIFO に書き込みます。

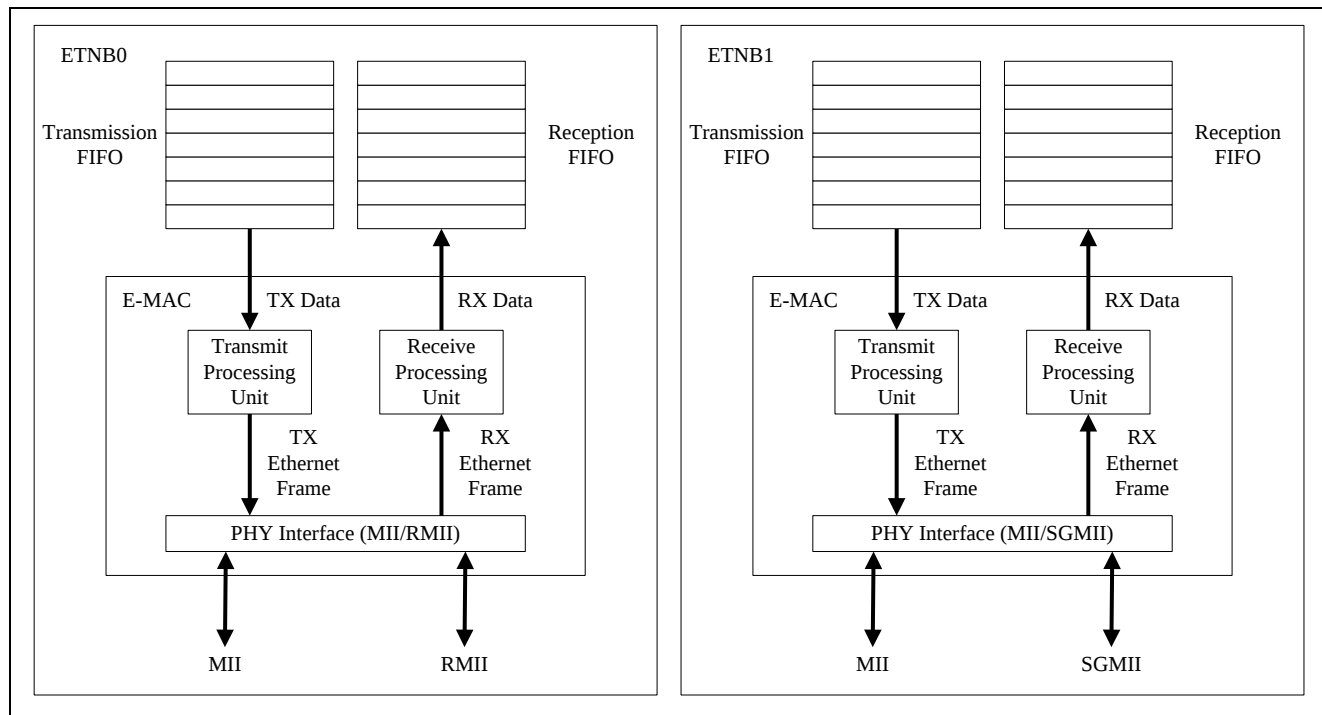


図 1-4 E-MAC の構成

1.3 AVB-DMAC の概要

AVB-DMAC の構成を図 1-5 に示します。AVB-DMAC は、ダイレクトメモリアクセス (DMA) 機能を使用して、URAM^(注1) 上の送信／受信用のイーサネットフレームデータの格納エリアと、送信／受信用の FIFO バッファとの間でイーサネットフレームデータの転送を行います。ソフトウェアから直接、FIFO バッファのデータを読み出したり、書き込んだりすることはできません。

AVB-DMAC が DMA 転送を実施するためには、ディスクリプタと呼ばれる送信および受信データの格納エリアのアドレスを含む情報が必要になります。AVB-DMAC は、ディスクリプタに書き込まれた情報に従って、送信データの格納エリアから送信するデータの読み出し、または受信したデータを受信データ格納エリアへの書き込みを行います。このディスクリプタは、URAM 上に配置する必要があります。

ディスクリプタを複数個並べ、ディスクリプタをリスト化することにより、複数のイーサネットフレームデータの送信／受信を連続的に行うことも可能です。

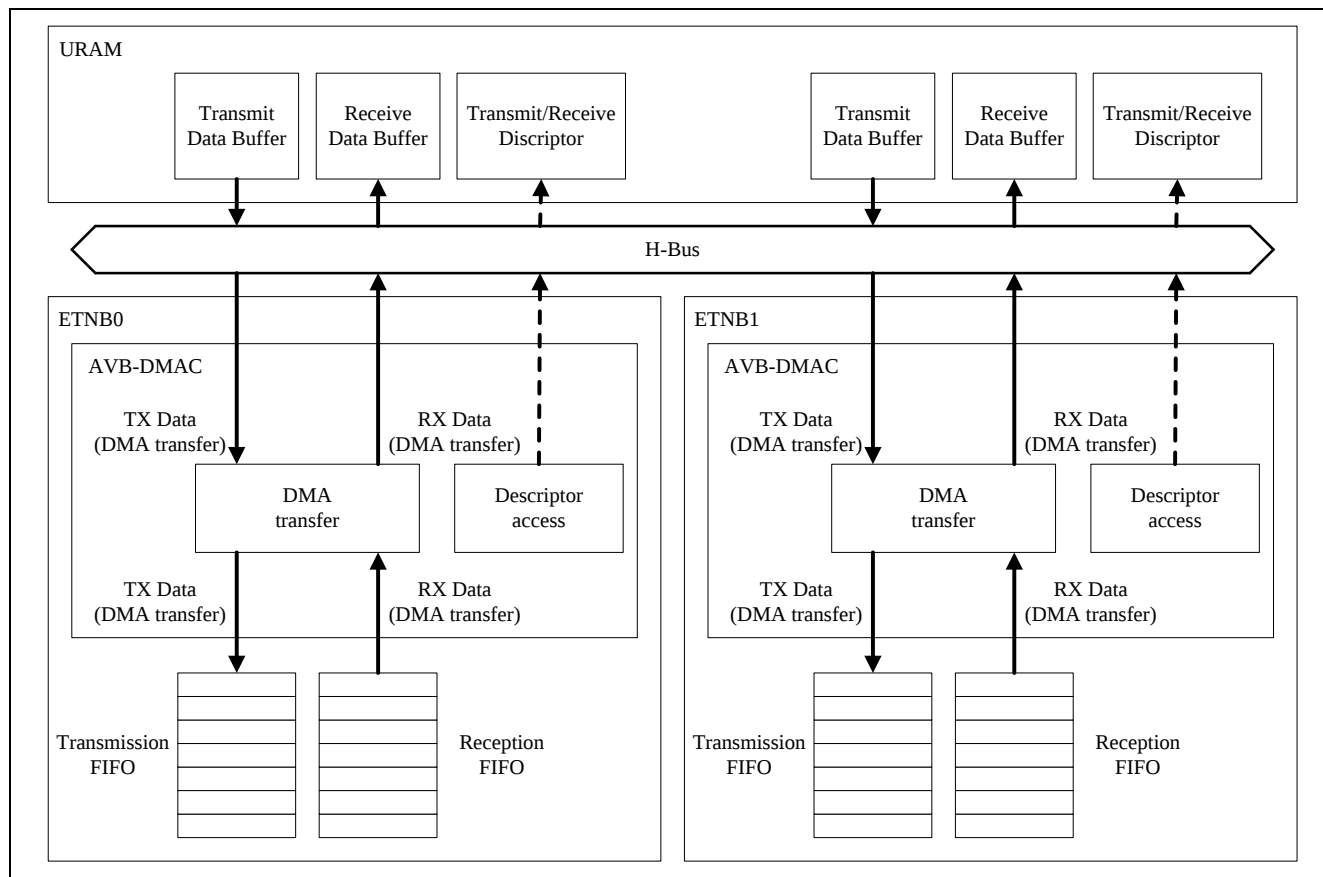


図 1-5 AVB-DMAC の構成

(注 1) URAM (ユーザー RAM) は、LRAM (ローカル RAM) または CRAM (クラスタ RAM) に該当します。

2. Ethernet AVB サンプルプログラム

2.1 サンプルプログラム概要

図 2-1 に OSI 参照モデルと本サンプルプログラムがカバーする範囲を示します。本サンプルプログラムは、ETNB モジュールの初期設定から PHY-LSI の初期設定までをサポートしています。

対向デバイスと通信する場合は TCP/IP 等の上位プロトコルをお客様側で構築していただく必要があります。

OSI参照モデル			プロトコル例	
第7層	アプリケーション層		HTTP, FTP	お客様側で構築いただく範囲
第6層	プレゼンテーション層			
第5層	セッション層			
第4層	トランスポート層		TCP, UDP	
第3層	ネットワーク層		IP	
第2層	データリンク層	LLC	Ethernet	本サンプルプログラムがサポートする範囲
		MAC		
第1層	物理層			

図 2-1 OSI 参照モデルと本サンプルプログラムがカバーする範囲

2.2 サンプルプログラムの動作環境

図 2-2 に、サンプルプログラムの動作環境を示します。対向デバイスとの通信は行いませんが、リンク状態の確認のため評価ボードとホスト PC を LAN ケーブルで接続しています。2.1 節に記載の通り対向デバイスとの通信は上位レイヤのプロトコルを構築する必要があります。

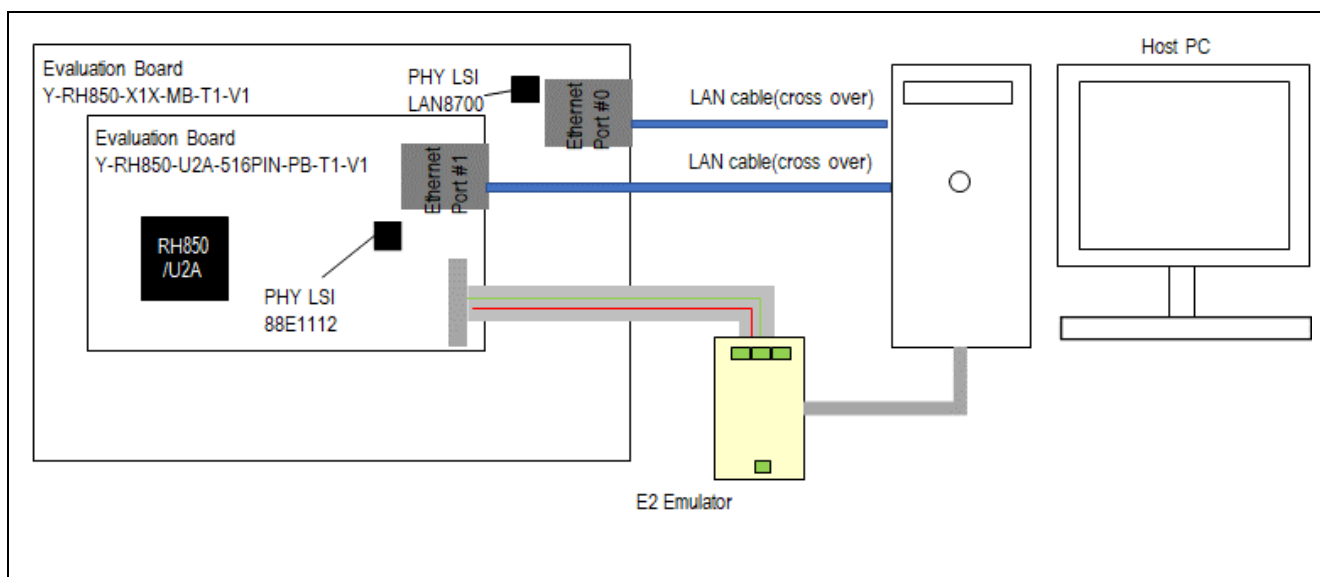


図 2-2 サンプルプログラムの動作環境

2.3 Ethernet AVB 接続例

本節では、MII(100Base-TX),SGMII(1000Base-TX)通信の接続時の動作について説明します。

なお、U2A と PHY-LSI 間で正常に接続が行えた場合、図 2-3 の例のように Windows10 の「イーサネットの状態」画面にて速度が表示されます。

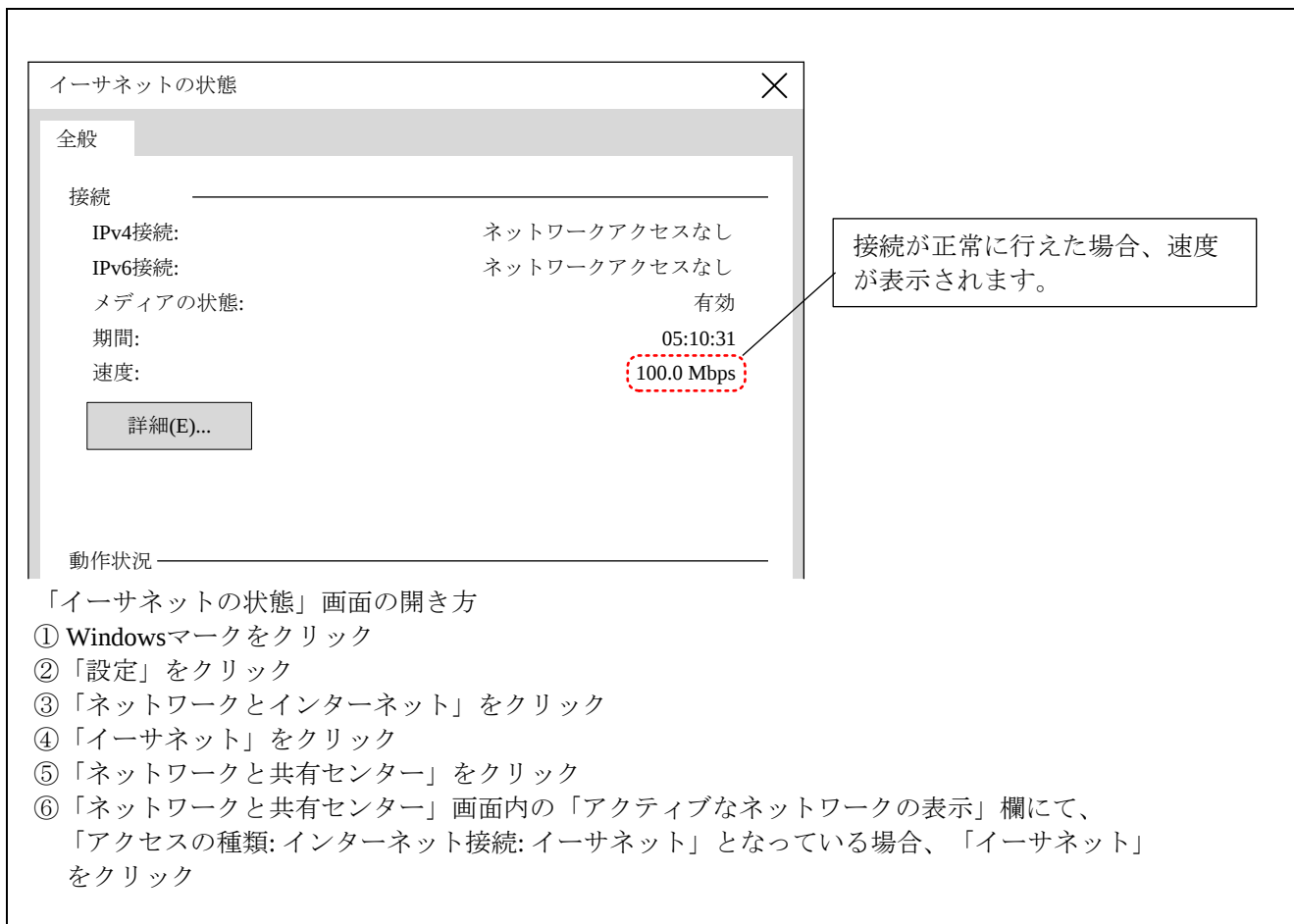


図 2-3 接続が成功した場合の「イーサネットの状態」画面の表示例

3. Ethernet AVB 初期設定手順

Ethernet AVB(ETNB)を使用するためには E-MAC、AVB-DMAC、ディスクリプタの初期設定及び設定が必要となります。

図 3-1 にサンプルコードを例に初期設定手順について説明します。

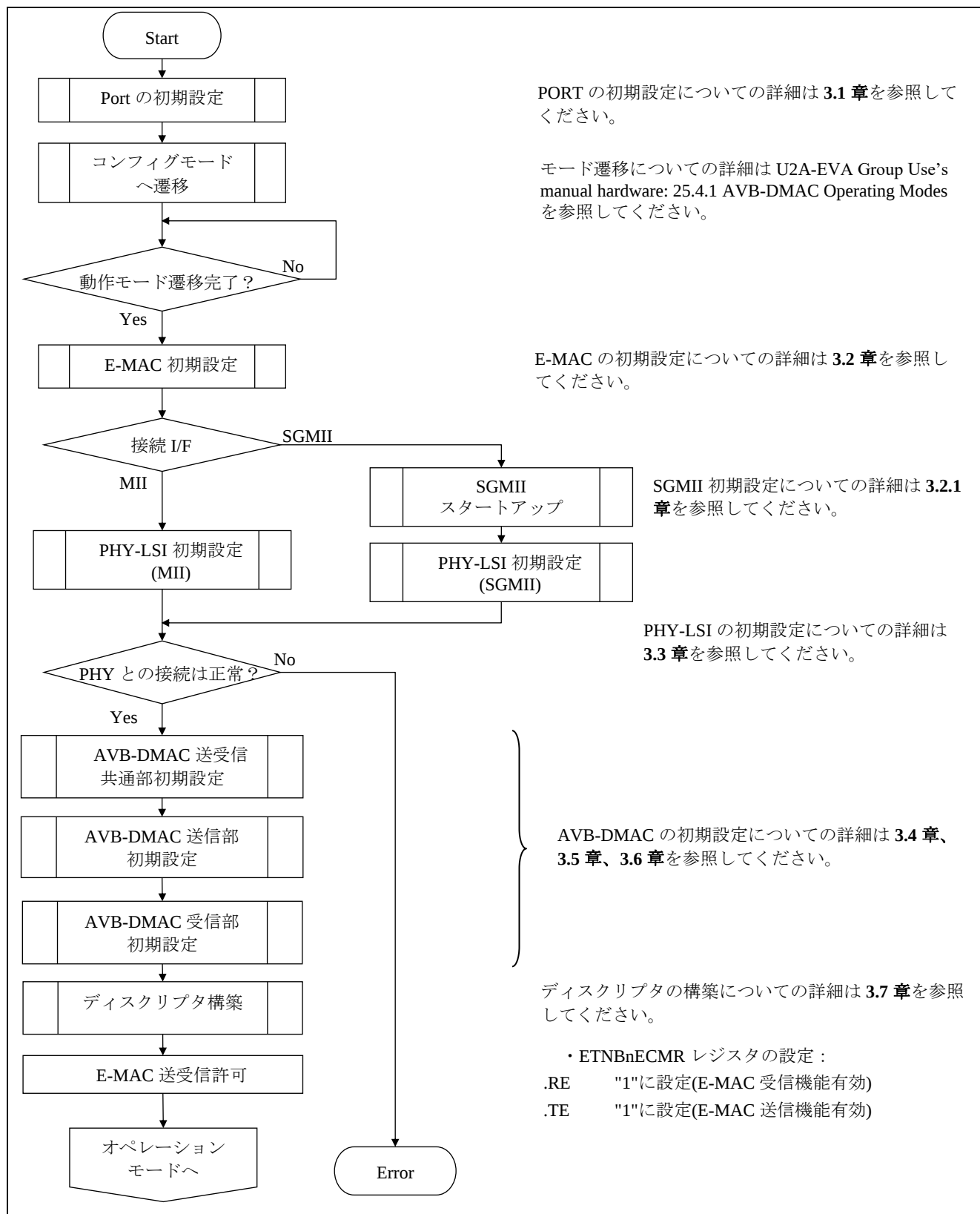


図 3-1 Ethernet AVB 初期設定手順例

3.1 PORT 初期設定例

3.1.1 ETNB0(MII)の場合

MII モードにて接続する場合の各信号用の PORT の初期設定を行います。本サンプルプログラムでは、評価ボード(Main Board :Y-RH850-X1X-MB-T1-V1)付属の Microchip Technology 製 LAN8700 を使用する場合を想定しています。

図 3-2 に MII モードの場合に使用する各信号 PORT について示します。

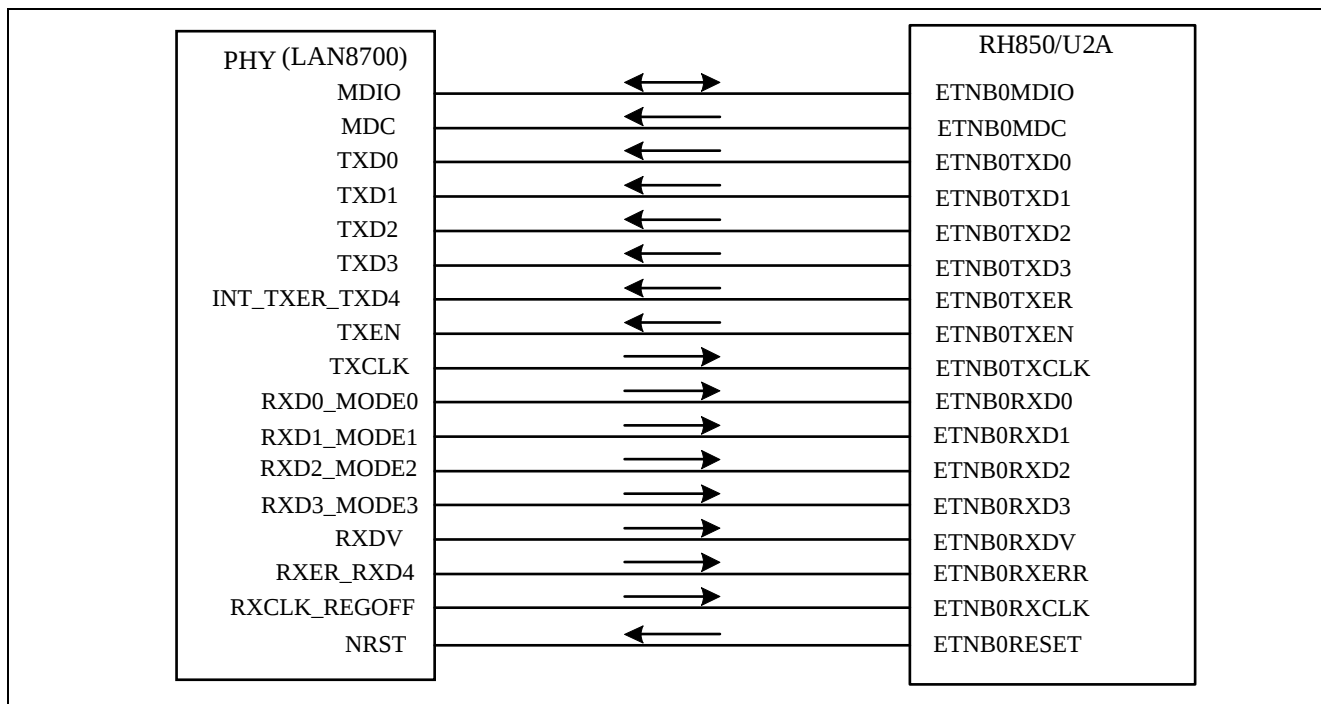


図 3-2 MII モード時の信号接続例

表 3-1、表 3-2、表 3-3 に各 PORT の初期設定例を示します。

表 3-1 PORT 初期設定例[MII] (1/3)

機能(信号)名	PORT 番号	方向	機能概要／設定内容
ETNB0TXCLK	P10_1	入力	送信クロック：100Mbps の場合 25MHz、10Mbps の場合 2.5MHz にて PHY により生成されます。
			入力モード(PM10.PM10_1=1) ソフトウェア I/O 制御(PIPC10.PIPC10_1=0) 兼用モード(PMC10.PMC10_1=1) 第 2 兼用(PFCAE10.PFCAE10_1=0, PFCE10.PFCE10_1=0, PFC10.PFC10_1=1)
ETNB0TXEN	P20_14	出力	送信可：TXD0～4 に有効な送信データがあることを示します。
			出力モード(PM20.PM20_14=0) ソフトウェア I/O 制御(PIPC20.PIPC20_14=0) 兼用モード(PMC20.PMC20_14=1) 第 2 兼用(PFCAE20.PFCAE20_14=0, PFCE20.PFCE20_14=0, PFC20.PFC20_14=1)
ETNB0TXD0	P20_9	出力	送信データビット 0：Ethernet PHY が受け入れる 4 本の送信データビットのうちのビット 0。
			出力モード(PM20.PM20_9=0) ソフトウェア I/O 制御(PIPC20.PIPC20_9=0) 兼用モード(PMC20.PMC20_9=1) 第 2 兼用(PFCAE20.PFCAE20_9=0, PFCE20.PFCE20_9=0, PFC20.PFC20_9=1) ポート出力バッファ特性 HIGH 指定(PDSC20.PDSC20_9=1)

表 3-2 PORT 初期設定例[MII] (2/3)

機能(信号)名	PORT 番号	方向	機能概要／設定内容
ETNB0TXD1	P20_10	出力	送信データビット 1 : Ethernet PHY が受け入れる 4 本の送信データビットのうちのビット 1。 出力モード(PM20.PM20_10=0) ソフトウェア I/O 制御(PIPC20.PIPC20_10=0) 兼用モード(PMC20.PMC20_10=1) 第 2 兼用(PFCAE20.PFCAE20_10=0, PFCE20.PFCE20_10=0, PFC20.PFC20_10=1) ポート出力バッファ特性 HIGH 指定(PDSC20.PDSC20_10=1)
ETNB0TXD2	P20_12	出力	送信データビット 2 : Ethernet PHY が受け入れる 4 本の送信データビットのうちのビット 2。 出力モード(PM20.PM20_12=0) ソフトウェア I/O 制御(PIPC20.PIPC20_12=0) 兼用モード(PMC20.PMC20_12=1) 第 2 兼用(PFCAE20.PFCAE20_12=0, PFCE20.PFCE20_12=0, PFC20.PFC20_12=1) ポート出力バッファ特性 HIGH 指定(PDSC20.PDSC20_12=1)
ETNB0TXD3	P20_13	出力	送信データビット 3 : Ethernet PHY が受け入れる 4 本の送信データビットのうちのビット 3。 出力モード(PM20.PM20_13=0) ソフトウェア I/O 制御(PIPC20.PIPC20_13=0) 兼用モード(PMC20.PMC20_13=1) 第 2 兼用(PFCAE20.PFCAE20_13=0, PFCE20.PFCE20_13=0, PFC20.PFC20_13=1) ポート出力バッファ特性 HIGH 指定(PDSC20.PDSC20_13=1)
ETNB0TXER	P20_8	出力	送信エラー : 10Base-T の場合は無視されます。 出力モード(PM20.PM20_8=0) ソフトウェア I/O 制御(PIPC20.PIPC20_8=0) 兼用モード(PMC20.PMC20_8=1) 第 2 兼用(PFCAE20.PFCAE20_8=0, PFCE20.PFCE20_8=0, PFC20.PFC20_8=1)
ETNB0RXCLK	P10_2	入力	受信クロック : 100Mbps の場合 25MHz、10Mbps の場合 2.5MHz にて PHY により生成されます。 入力モード(PM10.PM10_2=1) ソフトウェア I/O 制御(PIPC10.PIPC10_2=0) 兼用モード(PMC10.PMC10_2=1) 第 2 兼用(PFCAE10.PFCAE10_2=0, PFCE10.PFCE10_2=0, PFC10.PFC10_2=1)
ETNB0RXDV	P10_7	入力	受信データ有効 : デコードされた受信データが RXD0~3 にあることを示します。 入力モード(PM10.PM10_7=1) ソフトウェア I/O 制御(PIPC10.PIPC10_7=0) 兼用モード(PMC10.PMC10_7=1) 第 2 兼用(PFCAE10.PFCAE10_7=0, PFCE10.PFCE10_7=0, PFC10.PFC10_7=1)
ETNB0RXD0	P10_3	入力	受信データビット 0 : Ethernet PHY より受け入れられる 4 本の受信データビットのうちのビット 0。 入力モード(PM10.PM10_3=1) ソフトウェア I/O 制御(PIPC10.PIPC10_3=0) 兼用モード(PMC10.PMC10_3=1) 第 2 兼用(PFCAE10.PFCAE10_3=0, PFCE10.PFCE10_3=0, PFC10.PFC10_3=1)

表 3-3 PORT 初期設定例[MII] (3/3)

機能(信号)名	PORT 番号	方向	機能概要/設定内容
ETNB0RXD1	P10_4	入力	受信データビット 1 : Ethernet PHY より受け入れられる 4 本の受信データビットのうちのビット 1。
			入力モード(PM10.PM10_4=1) ソフトウェア I/O 制御(PIPC10.PIPC10_4=0) 兼用モード(PMC10.PMC10_4=1) 第 2 兼用(PFCAE10.PFCAE10_4=0, PFCE10.PFCE10_4=0, PFC10.PFC10_4=1)
ETNB0RXD2	P10_5	入力	受信データビット 2 : Ethernet PHY より受け入れられる 4 本の受信データビットのうちのビット 2。
			入力モード(PM10.PM10_5=1) ソフトウェア I/O 制御(PIPC10.PIPC10_5=0) 兼用モード(PMC10.PMC10_5=1) 第 2 兼用(PFCAE10.PFCAE10_5=0, PFCE10.PFCE10_5=0, PFC10.PFC10_5=1)
ETNB0RXD3	P10_6	入力	受信データビット 3 : Ethernet PHY より受け入れられる 4 本の受信データビットのうちのビット 3。
			入力モード(PM10.PM10_6=1) ソフトウェア I/O 制御(PIPC10.PIPC10_6=0) 兼用モード(PMC10.PMC10_6=1) 第 2 兼用(PFCAE10.PFCAE10_6=0, PFCE10.PFCE10_6=0, PFC10.PFC10_6=1)
ETNB0RXER	P10_0	入力	受信エラー : Ethernet PHY から現在送信されているフレームにエラーが検出されたことを示します。
			入力モード(PM10.PM10_0=1) ソフトウェア I/O 制御(PIPC10.PIPC10_0=0) 兼用モード(PMC10.PMC10_0=1) 第 2 兼用(PFCAE10.PFCAE10_0=0, PFCE10.PFCE10_0=0, PFC10.PFC10_0=1)
ETNB0MDC	P20_6	入力	マネジメントクロック : シリアル I/F 用のクロックです。
			出力モード(PM20.PM20_6=0) ソフトウェア I/O 制御(PIPC20.PIPC20_6=0) 兼用モード(PMC20.PMC20_6=1) 第 2 兼用(PFCAE20.PFCAE20_6=0, PFCE20.PFCE20_6=0, PFC20.PFC20_6=1)
ETNB0MDIO	P20_3	入出力	マネジメントデータ入出力 : シリアル I/F のデータの入出力。
			直接 I/O 制御(PIPC20.PIPC20_3=1) 兼用モード(PMC20.PMC20_3=1) 第 2 兼用(PFCAE20.PFCAE20_3=0, PFCE20.PFCE20_3=0, PFC20.PFC20_3=1)
ETNB0RESET	P20_0	出力	外部リセット : 外部リセット用の端子です。
			出力モード(PM20.PM20_0=0) ポートモード(PMC20.PMC20_14=0)

3.1.2 ETNB1(SGMII)の場合

SGMII モードにて接続する場合の各信号用の PORT の初期設定を行います。本サンプルプログラムでは、評価ボード(Piggy Back Board : Y-RH850-U2A-516PIN-PB-T1-V1)付属の Marvell Semiconductor 製 88E1112 を使用する場合を想定しています。図 3-3 に SGMII の場合に使用する各信号 PORT について示します。

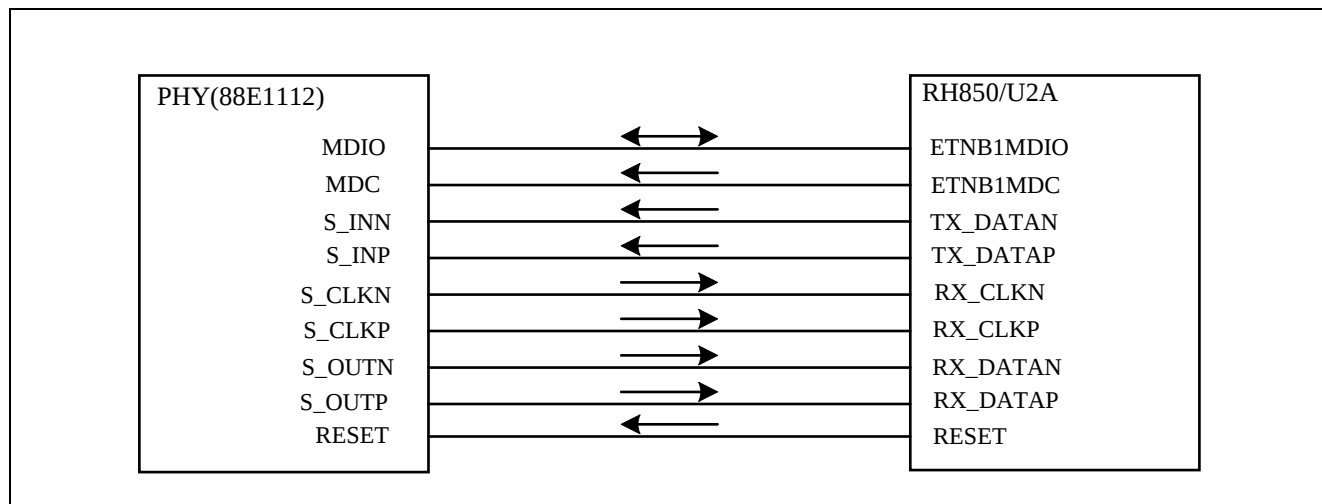


図 3-3 SGMII モード時の信号接続例

表 3-4 に各 PORT の初期設定例を示します。

表 3-4 PORT 設定例[SGMII]

機能(信号)名	PORT 番号	方向	機能概要／設定内容
TX_DATAN	-----	出力	SGMII 送信シリアルデータ出力(Negative) 専用端子のため、PORT 設定は不要。(注2)
TX_DATAP	-----	出力	SGMII 送信シリアルデータ出力(Positive) 専用端子のため、PORT 設定は不要。(注2)
RX_CLKN	-----	入力	専用端子のため、PORT 設定は不要。
RX_CLKP	-----	入力	専用端子のため、PORT 設定は不要。
RX_DATAN	-----	入力	SGMII 送信シリアルデータ入力(Negative) 専用端子のため、PORT 設定は不要。(注2)
RX_DATAP	-----	入力	SGMII 送信シリアルデータ入力(Positive) 専用端子のため、PORT 設定は不要。(注2)
ETNB1MDC	P3_6	出力	マネジメントクロック 出力モード(PM3.PM3_6=0) ソフトウェア I/O 制御(PIPC3.PIPC3_6=0) 兼用モード(PMC3.PMC3_6=1) 第 3 兼用(PFCAE6.PFCAE3_6=0, PFCE6.PFCE3_6=1, PFC6.PFC3_6=0)
ETNB1MDIO	P3_7	入出力	マネジメントデータ入出力 出力モード(PM3.PM3_7=0) ソフトウェア I/O 制御(PIPC3.PIPC3_7=1) 兼用モード(PMC3.PMC3_7=1) 第 3 兼用(PFCAE3.PFCAE3_7=0, PFCE3.PFCE3_7=1, PFC3.PFC3_7=0)
RESET	P9_0	出力	外部 SW リセット 出力モード(PM9.PM9_0=0) ポートモード(PMC9.PMC9_0=0)

(注2) RX_DATAN、RX_DATAP、TX_DATAN および TX_DATAP の極性設定については、OPBT14 で定義されます。詳細は U2A-EVA Group User's Manual Hardware : 51.12.20 "OPBT 14"を参照してください。

3.2 E-MAC の初期設定例

図 3-4 にサンプルプログラムを例とした、E-MAC 初期設定のフローを示します。

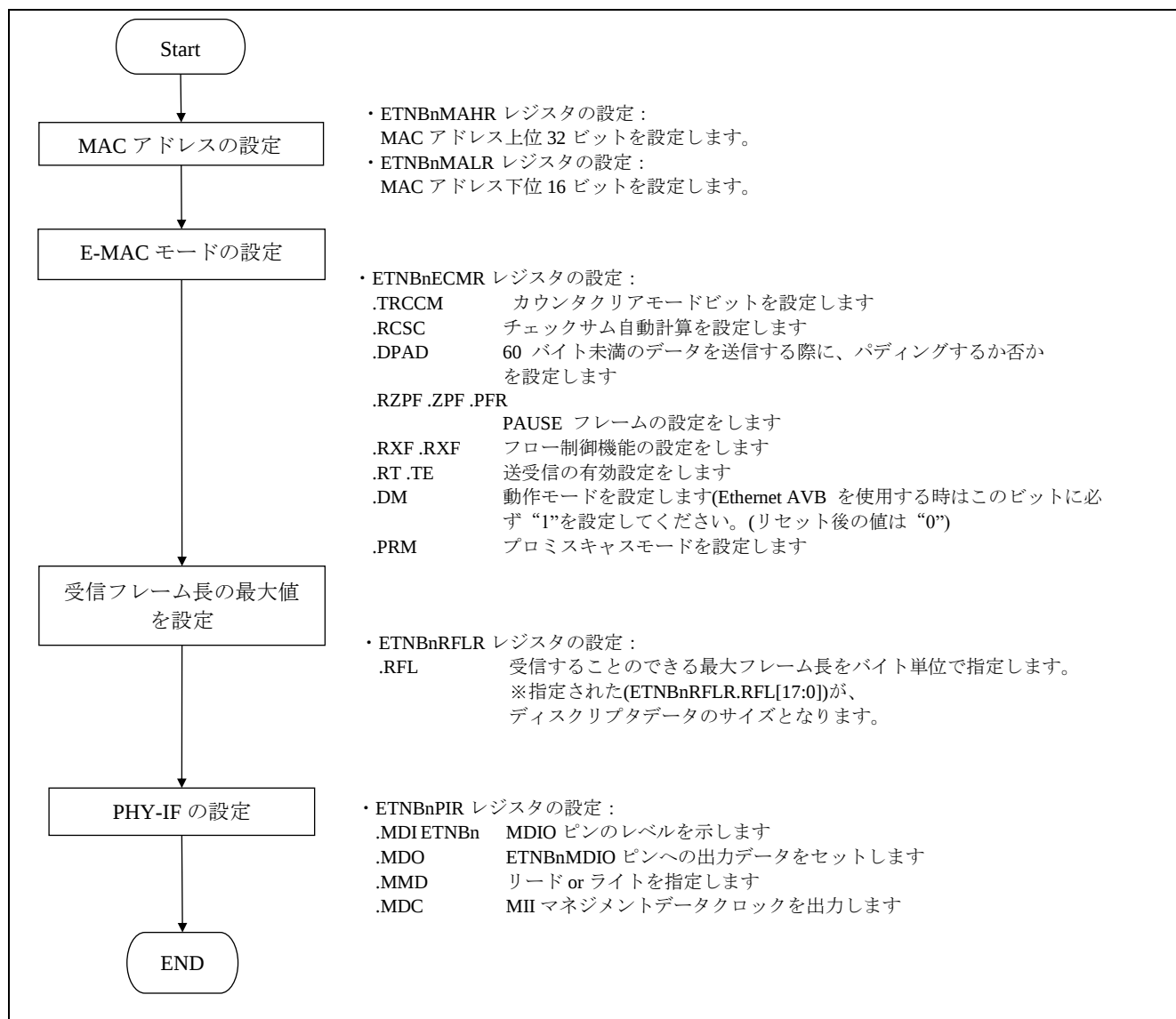


図 3-4 E-MAC の初期設定フロー例

表 3-5 に E-MAC 初期設定時の各機能のレジスタ設定例を示します。

表 3-5 E-MAC 初期設定時のレジスタ設定例

レジスタ名	設定値	機能
ETNBnMAHR	0x00800F00	MAC アドレスの上位 32 ビット(00:80:0F:00)
ETNBnMALR	0x00000001	MAC アドレスの下位 16 ビット(00:01)
ETNBnECMR	0x00200003	カウンタクリアモード : 書き込み時に 0 クリアを選択
		チェックサム計算 : チェックサム自動計算を選択
		TIME=0 のポーズフレーム受信無効
		TIME=0 のポーズフレームへの応答、送信無効
		データパディング : データパディング無効を選択
		モード選択 : 全二重モード選択
		プロミスキャスモード : プロミスキャスモード操作を選択
ETNBnRFLR.	0x000005F9	受信フレームデータ長(1,529byte)
ETNBnECSIPR	0x00000000	E-MAC 割り込み禁止

3.2.1 SGMII スタートアップ

図 3-5 にサンプルプログラムを例とした、SGMII のスタートアップ例を示します。

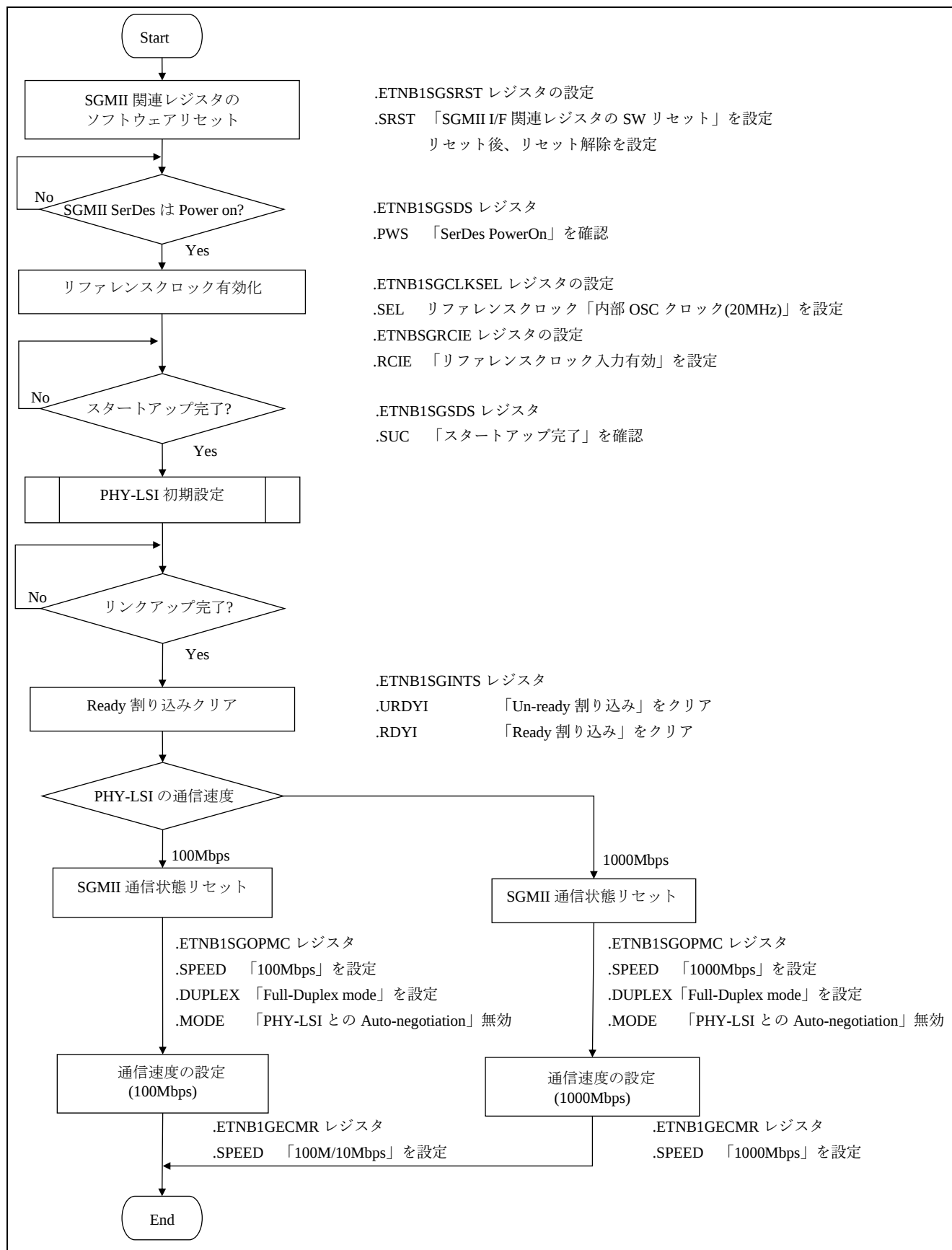


図 3-5 SGMII のスタートアップフロー例

表 3-6 に SGMII のスタートアップ時の各機能のレジスタ設定例を示します。

表 3-6 SGMII スタートアップ時のレジスタ設定例[SGMII]

レジスタ名	設定値	機能
ETNB1SGSRST	0x00000000	リセット解除
ETNB1SGCLKSEL ^(注3)	0x01	内部クロック (20MHz)
ETNB1SGRCIE	0x01	リファレンスクロック入力有効
ETNB1SGINTS	0x00000000	Un-ready 割り込み無し(状態をクリア)
		Ready 割り込み無し(状態をクリア)
ETNB1SGOPMC	0x0000000B (1Gbps 通信時)	通信速度=1000Mbps
		Full-Duplex モード
		PHY-LSI 間 Auto-Negotiation : バイパスモード
	0x00000007 (100Mbps 通信時)	通信速度=100Mbps
		Full-Duplex モード
		PHY-LSI 間 Auto-Negotiation : バイパスモード
ETNB1GECMR	0x00000001 (1Gbps 通信時)	通信速度設定=1000Mbps
	0x00000000 (100Mbps 通信時)	通信速度設定=100Mbps
ETNB1ECMR	0x00000002	モード選択 : 全二重モード選択

(注3) SGMII の基準クロックに MOSC を使用する場合、20MHz の発振子を使用する必要があります。

3.3 PHY-LSI の初期設定例

Ethernet PHY のレジスタへのアクセスは、レジスタ ETNBnPIR を介して実施されます。ETNBnPIR は IEEE802.3u にて指定された MII フレーム形式に準拠するシリアル I/F として使用されます。

図 3-6、図 3-7 にデータのフレーム構造とタイミングについて示します。MII マネジメントフレームフォーマットは、U2A-EVA Group User's Manual Hardware : 「25.4.13.2 MII Management Frame Format」を参照ください。

PHY レジスタへのアクセスは、ETNBnPIR にて行われますが、アクセスは 1 ビット単位 of データ書き込み、1 ビット単位 of データ読み取り、バスリリース、および独立したバスリリースの組み合わせによって実装されます。

PHY レジスタへのアクセスタイミングについての詳細は、U2A-EVA Group User's Manual Hardware : 25.4.13.3 MII Register Access Procedure を参照ください。(タイミングは PHY-LSI の種類によって異なります)。

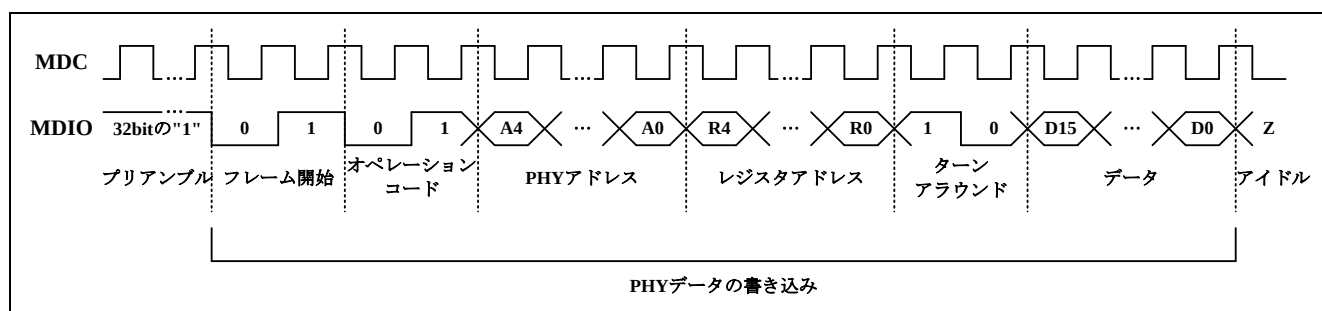


図 3-6 MDIO のタイミングとフレーム構造（書き込みサイクル）

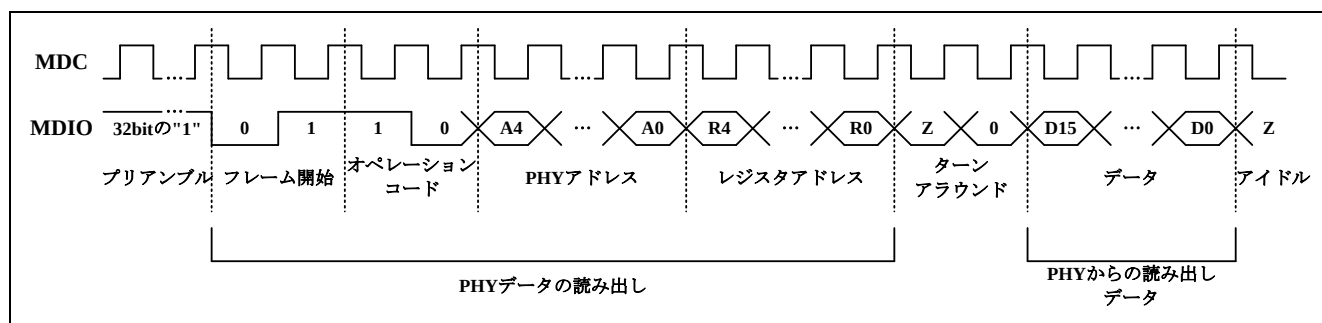


図 3-7 MDIO のタイミングとフレーム構造（読み出しサイクル）

3.3.1 ETNB0(MII)の場合

PHY-LSI は、使用する PHY に合わせて初期設定してください。本サンプルプログラムでは、評価ボード (Main Board : Y-RH850-X1X-MB-T1-V1) 付属の Microchip Technology 製 LAN8700 を使用する場合は想定しています。図 3-8 に PHY-LSI の初期設定のシーケンス例を示します。

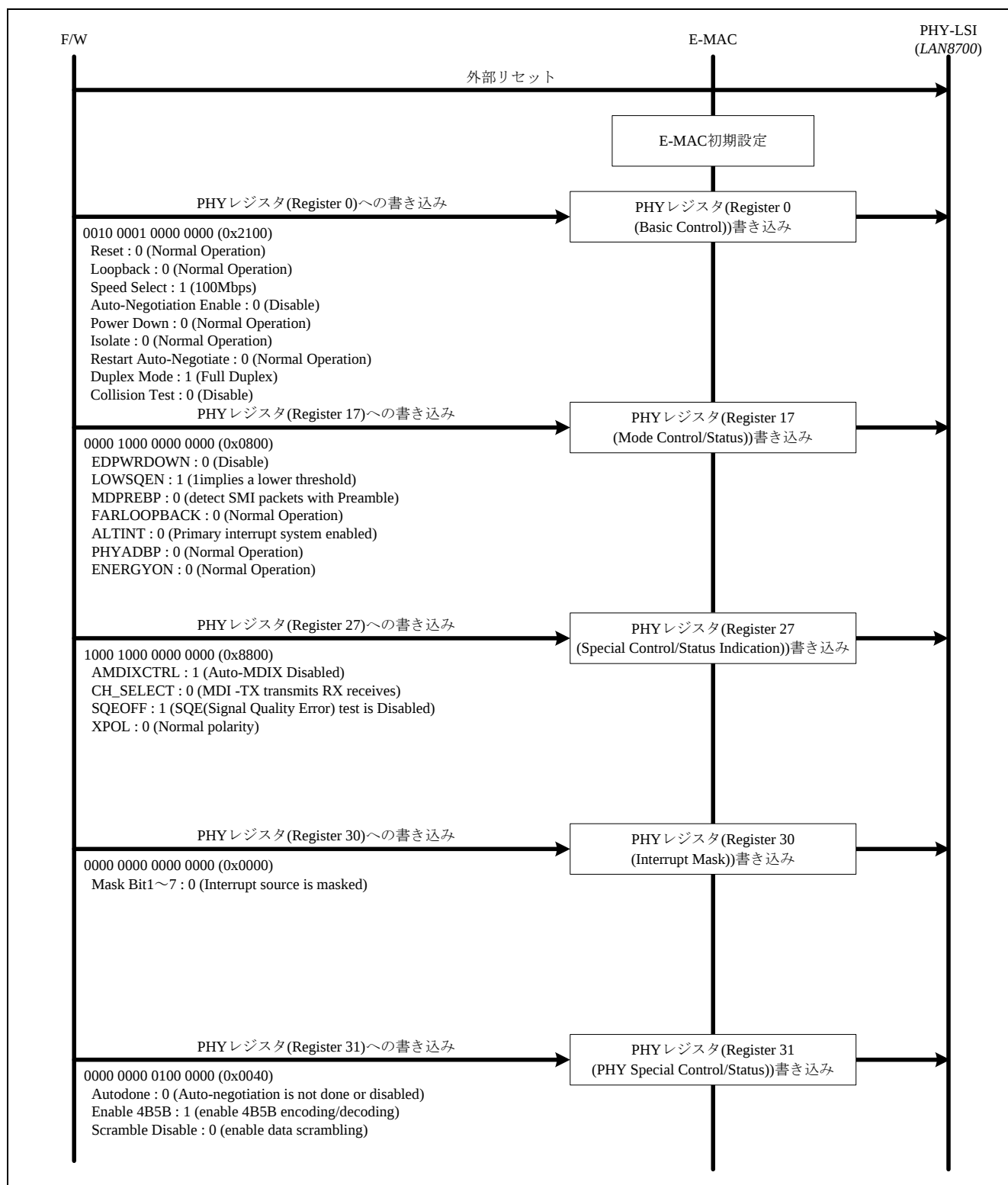


図 3-8 PHY-LSI(LAN8700 MII)の初期設定シーケンス例

3.3.2 ETNB1(SGMII)の場合

PHY-LSI は、使用する PHY に合わせて初期設定してください。本サンプルプログラムでは、評価ボード (Piggy Back Board : Y-RH850-U2A-516PIN-PB-T1-V1) 付属の Marvell Semiconductor 製 88E1112 を使用する場合は想定しています。図 3-9 に PHY-LSI の初期設定例のシーケンス例を示します。

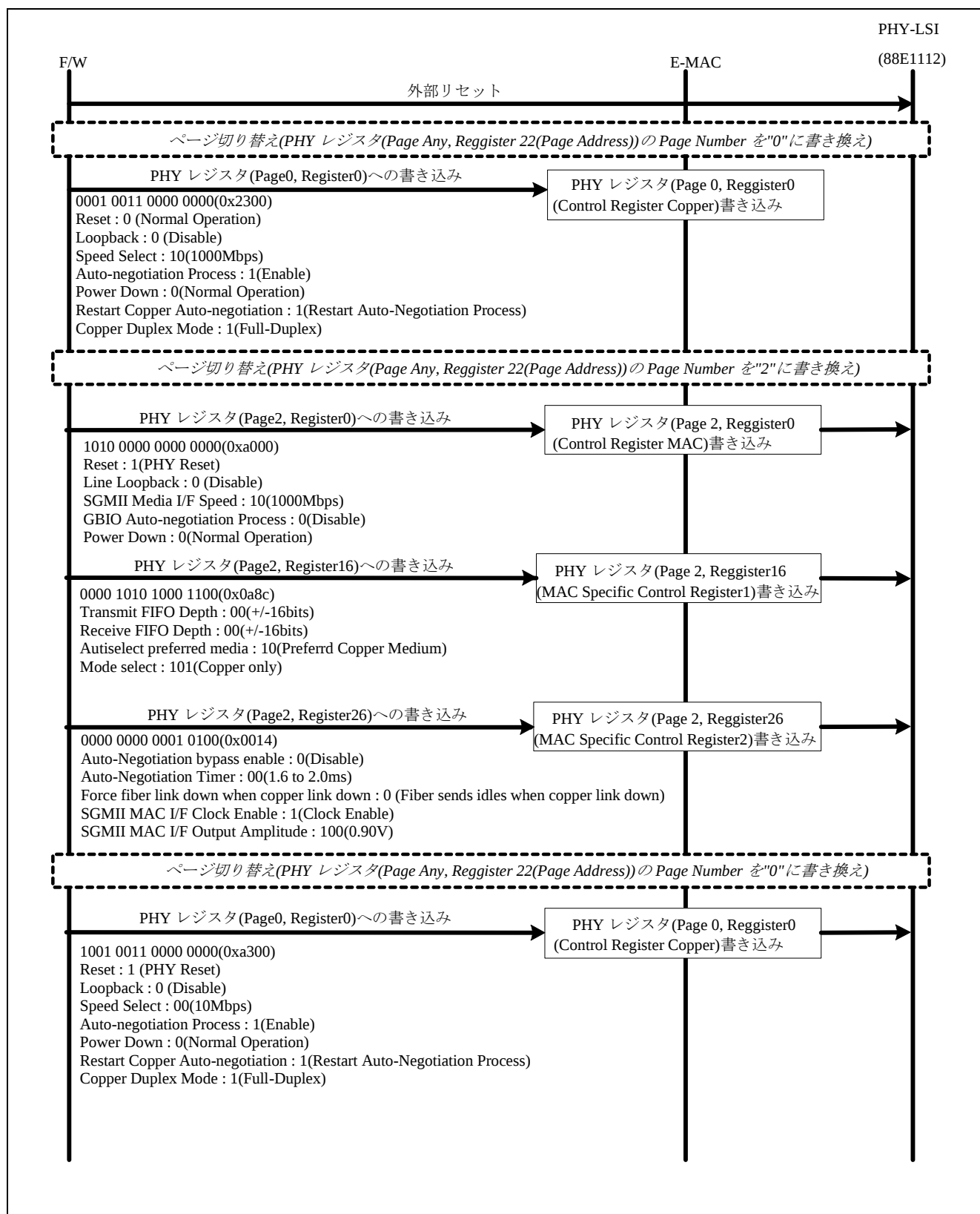


図 3-9 PHY-LSI(88E1112 SGMII)の初期設定例シーケンス例

3.4 AVB-DMAC 送受信共通部の初期設定例

図 3-10 にサンプルプログラムの AVB-DMAC の送受信共通部の初期設定のフロー例を示します。

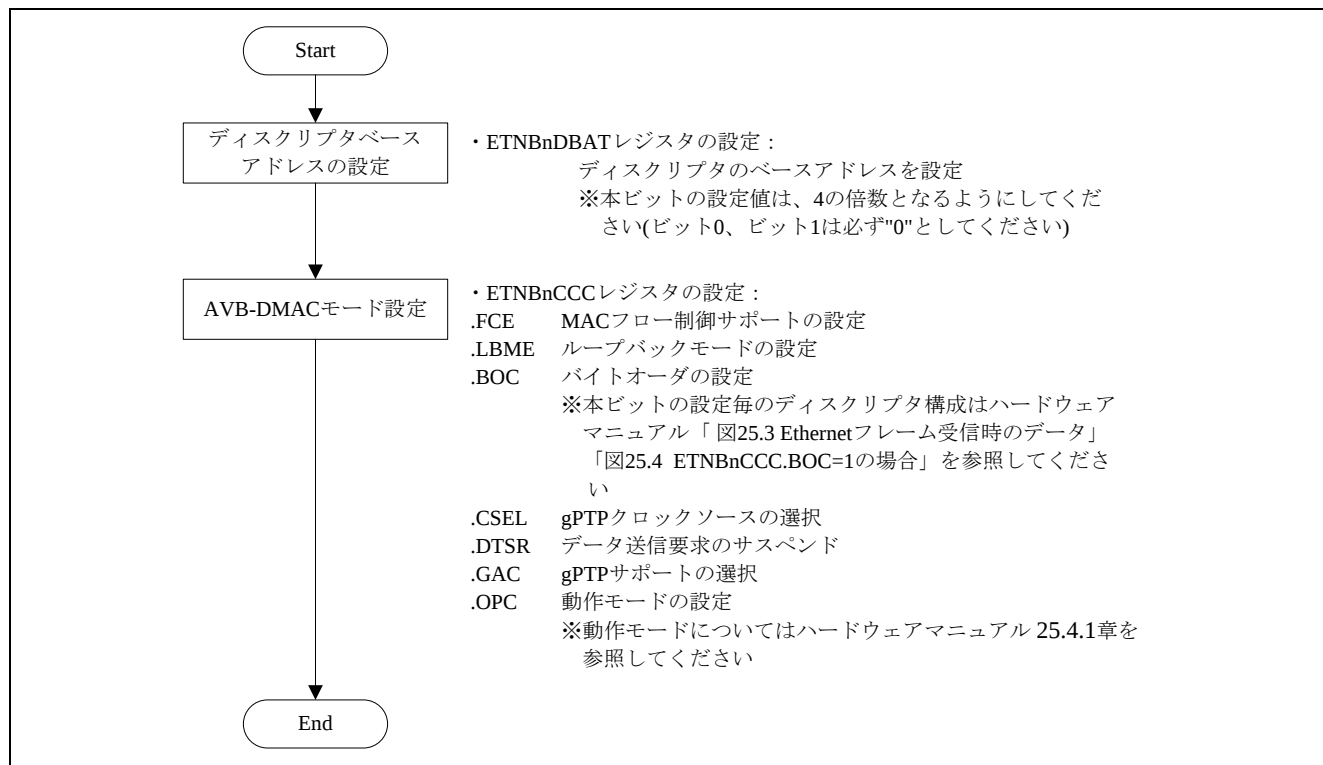


図 3-10 AVB-DMAC 送受信共通部の初期設定フロー例

表 3-7 に AVB-DMAC 送受信共通部の初期設定時の各機能のレジスタ設定例を示します。

表 3-7 AVB-DMAC 送受信共通部の初期設定時のレジスタ設定例

レジスタ名	設定値	機能
ETNBnDBAT	_ETNB_TEST_DBAT_TOP	ディスクリプタベースアドレス(URAM アドレス)を設定
ETNBnCCC	0x01000001	LOOPBACK モード有効に設定
		バイトオーダーをビッグエンディアンに設定
		コンフィグレーションモードに設定

3.5 AVB-DMAC 送信部の初期設定例

図 3-11 にサンプルプログラムの AVB-DMAC の送信部の初期設定のフロー例を示します。

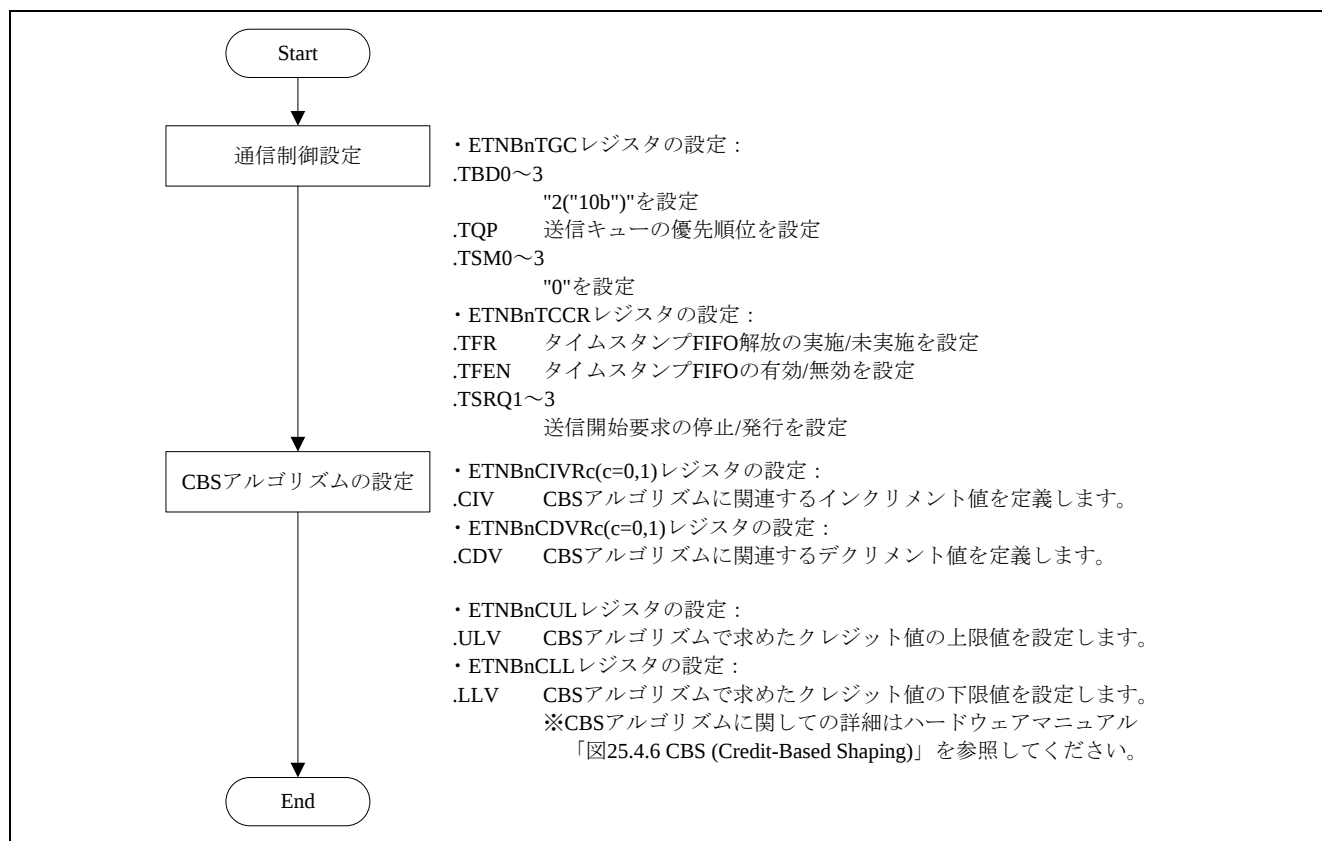


図 3-11 AVB-DMAC 送信部の初期設定フロー例

表 3-8 に AVB-DMAC 送信部の初期設定時のレジスタ設定例を示します。

表 3-8 AVB-DMAC 送信部の初期設定時のレジスタ設定例

レジスタ名	設定値	機能
ETNBnTGC	0x00222200	送信キュー優先度を非 AVB モードに
ETNBnTCCR	0x00000000	タイムスタンプ FIFO 動作無し
ETNBnCIVRc(c=0, 1)	0x00000001	CBS インクリメント値を 1
ETNBnCDVRc (c=0, 1)	0xFFFFFFFF	CBS デクリメント値を-1
ETNBnCULc (c=0, 1)	0x7FFFFFFF	CBS 上限値を設定
ETNBnCLLc (c=0, 1)	0x80000001	CBS 下限値を設定

3.6 AVB-DMAC 受信部の初期設定例

図 3-12 にサンプルプログラムの AVB-DMAC の受信部の初期設定のフロー例を示します。

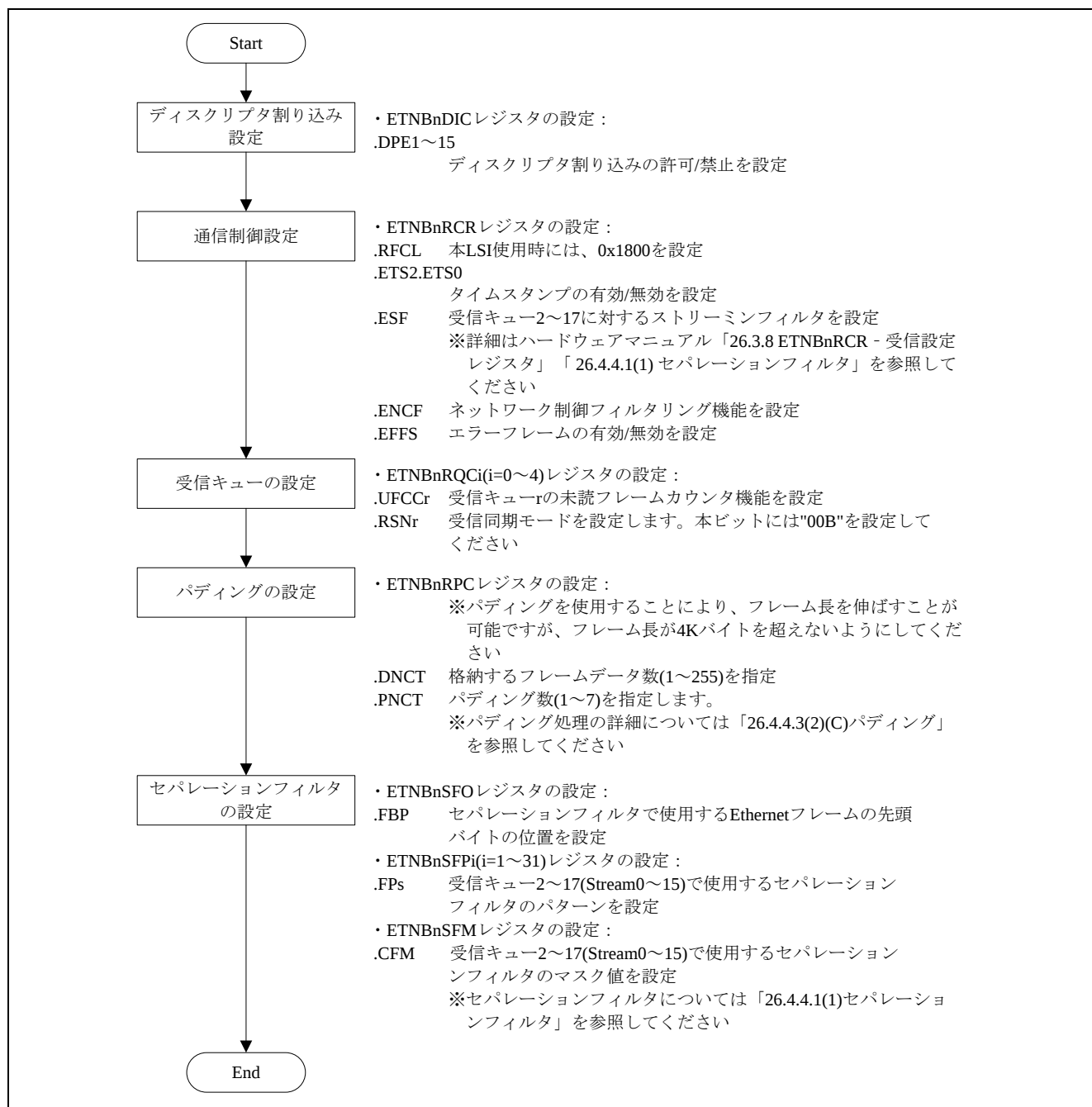


図 3-12 AVB-DMAC 受信部の初期設定フロー例

表 3-9 に AVB-DMAC 受信部の初期設定時のレジスタ設定例を示します。

表 3-9 AVB-DMAC 受信部の初期設定時のレジスタ設定例

レジスタ名	設定値	機能
ETNBnDIC	0x000000F0	ディスクリプタ割り込み 7~4 を有効
ETNBnRCR	0x18000005	受信 FIFO 警告レベル(0x1800 (推奨値))
		ストリーミングフィルタ設定(01b)
		エラーフレーム有効
ETNBnRQCi(i=0~4)	0x00000000	受信キューの設定 (未設定)
ETNBnRPC	0x00BC0400	格納フレームデータ数(188)
		パディング数(4)
ETNBnSFO	0x00000000	セパレーションフィルタオフセット値 (未設定)
ETNBnSFPI(i=0)	0xBFFFFFFF	受信キュー2(Stream0)のセパレーションフィルタパターン
ETNBnSFPI(i=1)	0x0000BBBB	受信キュー2(Stream0)のセパレーションフィルタパターン
ETNBnSFPI(i=2~31)	0x00000000	受信キュー3~17(Stream1~15)のセパレーションフィルタパターン (未設定)
ETNBnSFM0	0xFFFFFFFF	セパレーションフィルタマスク値
ETNBnSFM1	0x0000FFFF	セパレーションフィルタマスク値

3.7 ディスクリプタの構築例

図 3-13 にサンプルプログラムの受信ディスクリプタの構築例を、図 3-14、図 3-15 にサンプルプログラムを示します。本サンプルプログラムでは、4つの受信キュー(4~7)を制御するディスクリプタを構築します。全ての受信キューに対して、データタイプ(DT)が FEMPT、データサイズ(DS)が 1516 バイトのディスクリプタを用意します。

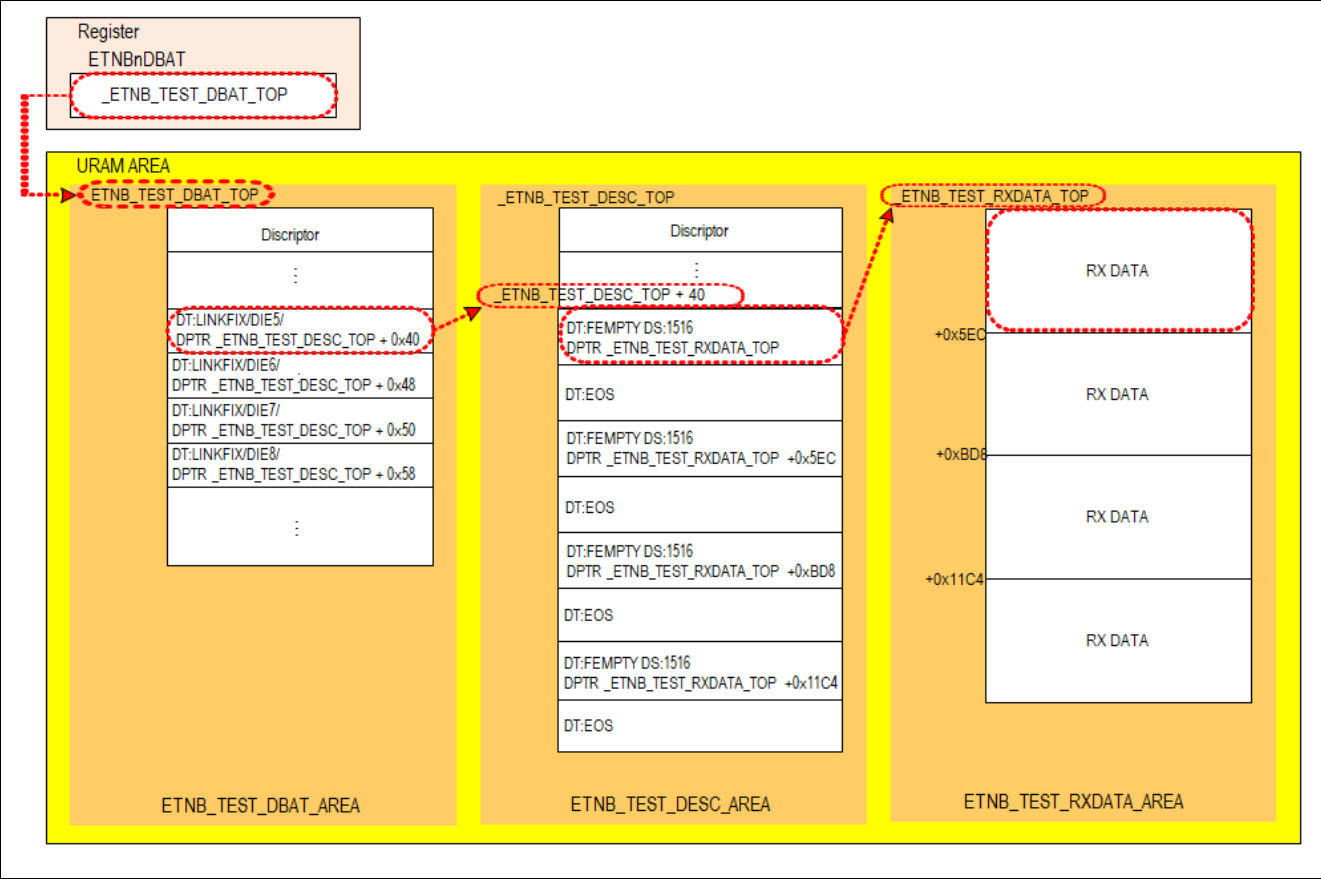


図 3-13 サンプルプログラムの受信ディスクリプタの構築例

```

/* 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;

(中略)

/* Descriptor type enumeration */
typedef enum { /* DT:Rx:Tx: Description */
    enDT0 = 0, /* 0:--:--: Reserved. */
    enDT1, /* 1:--:--: Reserved. */
    enLEEMPTY, /* 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;

(中略)

```

図 3-14 サンプルプログラムの受信ディスクリプタ構築処理例(1/2)

```

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

        (中略)

UL      ETNB_TEST_RXDATA_TOP;

        (中略)

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

        (中略)

    UB      *pttRxDataPtr;
    UL      u4tQueNo;
    UL      u4tDataIdx;

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

        (中略)

    /*---- Rx descriptor configuration ----*/
    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 < 0x600UL; u4tDataIdx++ ) /* Set default value in Rx data
area. */
        {
            pttRxDataPtr[u4tDataIdx] = 0xFFU;
        }

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

    return;
}

```

図 3-15 サンプルプログラムの受信ディスクリプタ構築処理例(2/2)

図 3-16 にサンプルプログラムの送信ディスクリプタの構築例を、図 3-17、図 3-18 にサンプルプログラムを示します。本サンプルプログラムでは、4 つの送信キュー(0~3)を制御するディスクリプタを構築します。全ての送信キューに対して、データタイプ(DT)が FSINGLE、データサイズ(DS)が 1516 バイトのディスクリプタを用意します。

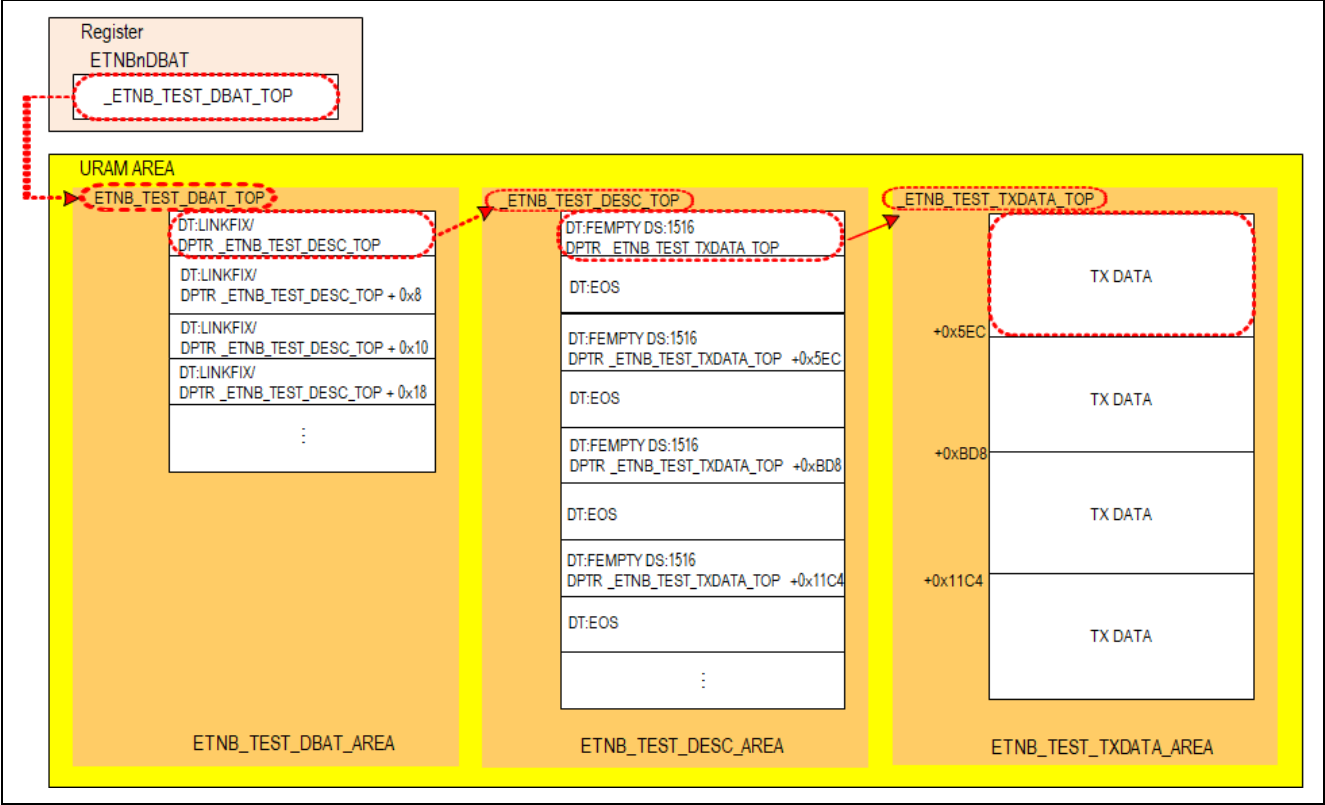


図 3-16 サンプルプログラムの送信ディスクリプタの構築例

```

/* 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;

(中略)

/* Descriptor type enumeration */
typedef enum { /* DT:Rx:Tx: Description */
    enDT0 = 0, /* 0:--:--: Reserved. */
    enDT1, /* 1:--:--: Reserved. */
    enLEEMPTY, /* 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;

(中略)

```

図 3-17 サンプルプログラムの送信ディスクリプタ構築処理例 (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;

(中略)

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

(中略)

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. */
}

(中略)

return;
}

```

図 3-18 サンプルプログラムの送信ディスクリプタ構築処理 (2/2)

4. その他の設定

4.1 オプションバイトについて

ETNB0 にて MII モードを使用する場合、オプションバイト 14 (OPBT14) のモード選択ビット (ETN_RMII_SEL) の設定 (ETNB0 の MII/RMII の選択、**MI**I を設定) を行う必要があります。

同様に、ETNB1 にて SGMII モードを使用する場合、オプションバイト 14 (OPBT14) のモード選択ビット (ETN_SGMII_SEL) の設定 (ETNB1 の MII/SGMII の選択、**SG**MII を設定) を行う必要があります。

4.2 モジュールスタンバイレジスタについて

ETNB0 および ETNB1 の機能を有効にするためには、モジュールスタンバイレジスタ (MSR_ETNB) の設定 (機能へのクロック供給有無の設定) を行う必要があります。このレジスタの初期値は ETNB0、ETNB1 とともに無効 (ETNB0、ETNB1 とともにクロック供給無し) が設定されます。

5. 付録

5.1 モジュール一覧

以下にサンプルプログラムのモジュール一覧を示します。

表 5-1 にモジュール一覧を示します。

表 5-1 モジュール一覧

モジュール名	ラベル名	機能
メインルーチン	main_pm0	各種設定、アプリケーションの起動を行います。
PORT 初期設定	R_PORT_Init	PORT の初期設定を行います。
PHY-LSI リセット	R_ETNB_ResetPhy	Ethernet PHY-LSI のリセットを行います。
E-MAC 初期設定	R_ETNB_InitEmacRegs	E-MAC の初期設定を行います。
PHY-LSI 初期設定	R_ETNB_InitPhyRegs	PHY-LSI の初期設定を行います。
AVB-DMAC 共通部初期設定	R_ETNB_InitDmacRegs	AVB-DMAC 共通部の初期設定を行います。
AVB-DMAC 送信部初期設定	R_ETNB_InitRxRegs	AVB-DMAC 送信処理部の各種設定を行います。
AVB-DMAC 受信部初期設定	R_ETNB_InitTxRegs	AVB-DMAC 受信処理部の各種設定を行います。
ディスクリプタ構築	loc_CreateDescriptor	ディスクリプタの構築を行います。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.0	2021.1.29	-	初版

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策
—— CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレイやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。
2. 電源投入時の処置
—— 電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。
3. 電源オフ時における入力信号
—— 当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。
4. 未使用端子の処理
—— 未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。
5. クロックについて
—— リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後、リセットを解除してください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。
6. 入力端子の印加波形
—— 入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。
7. リザーブアドレス（予約領域）のアクセス禁止
—— リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。
8. 製品間の相違について
—— 型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ幅射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
 2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
 4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
 5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
 6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
 7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
 8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
 9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
 10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
 11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
 12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

<http://www.renesas.com/contact>

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。