

Open-source  
SpaceWire-RMAP Target IP Core  
User Manual  
Ver.0.30

Document No. 230968003  
Date 10 Dec, 2013  
Shimafuji Electric Inc.  
Printed in Japan

## 改 訂 履 歴

版 数	日 付	内 容	備 考
Ver.0.10	2013/12/10	第一版	
Ver.0.20	2013/12/14	14.2. クロックに、説明を追加。 12. ファイル構成に、文字コードの注記を追加。	
Ver.0.30	2014/02/28	3. ブロック図の【図 1：RMAP Target IPブロック図】更新。 【図 2：RMAP Target IPを組み込んだシステム例】追加。 9.6. RMAP Command Decode Informationの表内の rmapErrorCodeの欄に説明追加。 9.8. Indication Signals 追加。	

# 目次

1. 概要.....	1
2. License と免責事項 .....	1
3. ブロック図.....	2
4. モジュールの機能.....	4
4.1. RMAP Target IP.....	4
4.2. SpaceWireCODECIP .....	4
4.3. RMAPDecoder.....	4
4.4. RMAPDMAController.....	4
5. 内部バス.....	5
5.1. バス幅の設定.....	5
5.2. データアラインメント .....	6
5.2.1. バス幅 8bit .....	6
5.2.2. バス幅 16bit .....	6
5.2.3. バス幅 32bit .....	6
6. RMAPTargetIP モジュールのインスタンス .....	7
7. RMAP データ長とアドレス.....	8
7.1. 命令ごとの取扱い可能な最大データ長 .....	8
7.2. アクティブバイトレーン .....	8
7.2.1. バス幅 8bit .....	8
7.2.2. バス幅 16bit .....	8
7.2.3. バス幅 32bit .....	9
8. データフロー .....	10
9. インターフェース信号 .....	12
9.1. Generic .....	12
9.2. SpaceWire Control Signal.....	12
9.3. SpaceWire Link Status .....	13
9.4. SpaceWire Error Status .....	14
9.5. SpaceWire Statistical Information.....	14
9.6. RMAP Command Decode Information.....	15
9.7. BUS Signals .....	16
9.8. Indication Signals.....	16
10. 内部バス信号動作タイミング .....	17
10.1. Write 動作.....	17
10.2. Read 動作.....	18
10.3. トランザクション許可取得タイミング .....	19
11. パッケージ.....	21
11.1. Constant 宣言.....	21
11.2. ステートマシン宣言 .....	21
12. ファイル構成.....	22
13. パフォーマンス.....	22
14. タイミング制約.....	23

14.1.	ucf ファイル.....	23
14.2.	クロック .....	24

# 1. 概要

本マニュアルは Open-source SpaceWire-RMAP Target IP Core に付いての説明を目的としたものです。

SpaceWireは、宇宙機（衛星）内で搭載コンポーネント間のデータ通信を行うための通信I/Fおよび通信プロトコルの仕様です。

RMAP（リモート・メモリ・アクセス・プロトコル）は、リモートの SpaceWire ノードのメモリに対して、読出しや書込みをサポートすることを目的としたプロトコルです。

SpaceWire ネットワークのコンフィギュレーションや SpaceWire ノードの制御、SpaceWire ノード間のデータ転送に使用できます。

SpaceWire-RMAP Target IP は、Xilinx 及び Altera の FPGA をターゲットに、RMAP Target 機能を実装することを目的とした VHDL 形式の IP コアです。

SpaceWire ネットワークから受信した RMAP パケットを WISHBONE ライクなバスを通して、メモリに対して Write、Read、Read-Modify-Write を行います。

バス幅は、generic 又はパッケージファイルの設定により、32bit、16bit、8bit の 3 種類が選択可能です。

## ➤ 主要仕様

- RMAP 仕様 : ECSS-E-ST-50-11C に適合
- SpaceWire 転送レート : 最大 100Mbps
- 内部バス : WISHBONE ライクなバス
- バス幅 : 8bit、16bit、32bit
- バス動作クロック : 50MHz ~ 100MHz
- バスのエンディアン : ビッグエンディアン

# 2. Licenseと免責事項

本ソフトウェアおよび関連文書のファイル（以下「ソフトウェア」）の複製を取得するすべての人に対し、ソフトウェアを無制限に扱うことを無償で許可します。これには、ソフトウェアの複製を使用、複写、変更、結合、掲載、頒布、サブライセンス、および/または販売する権利、およびソフトウェアを提供する相手に同じことを許可する権利も無制限に含まれます。

上記の著作権表示および本許諾表示を、ソフトウェアのすべての複製または重要な部分に記載するものとします。

ソフトウェアは「現状のまま」で、明示であるか暗黙であるかを問わず、何らの保証もなく提供されます。ここでいう保証とは、商品性、特定の目的への適合性、および権利非侵害についての保証も含みますが、それに限定されるものではありません。 作者または著作権者は、契約行為、不法行為、またはそれ以外であろうと、ソフトウェアに起因または関連し、あるいはソフトウェアの使用またはその他の扱いによって生じる一切の請求、損害、その他の義務について何らの責任も負わないものとします。

### 3. ブロック図

RMAP Target IP ブロック図を下記に示します。

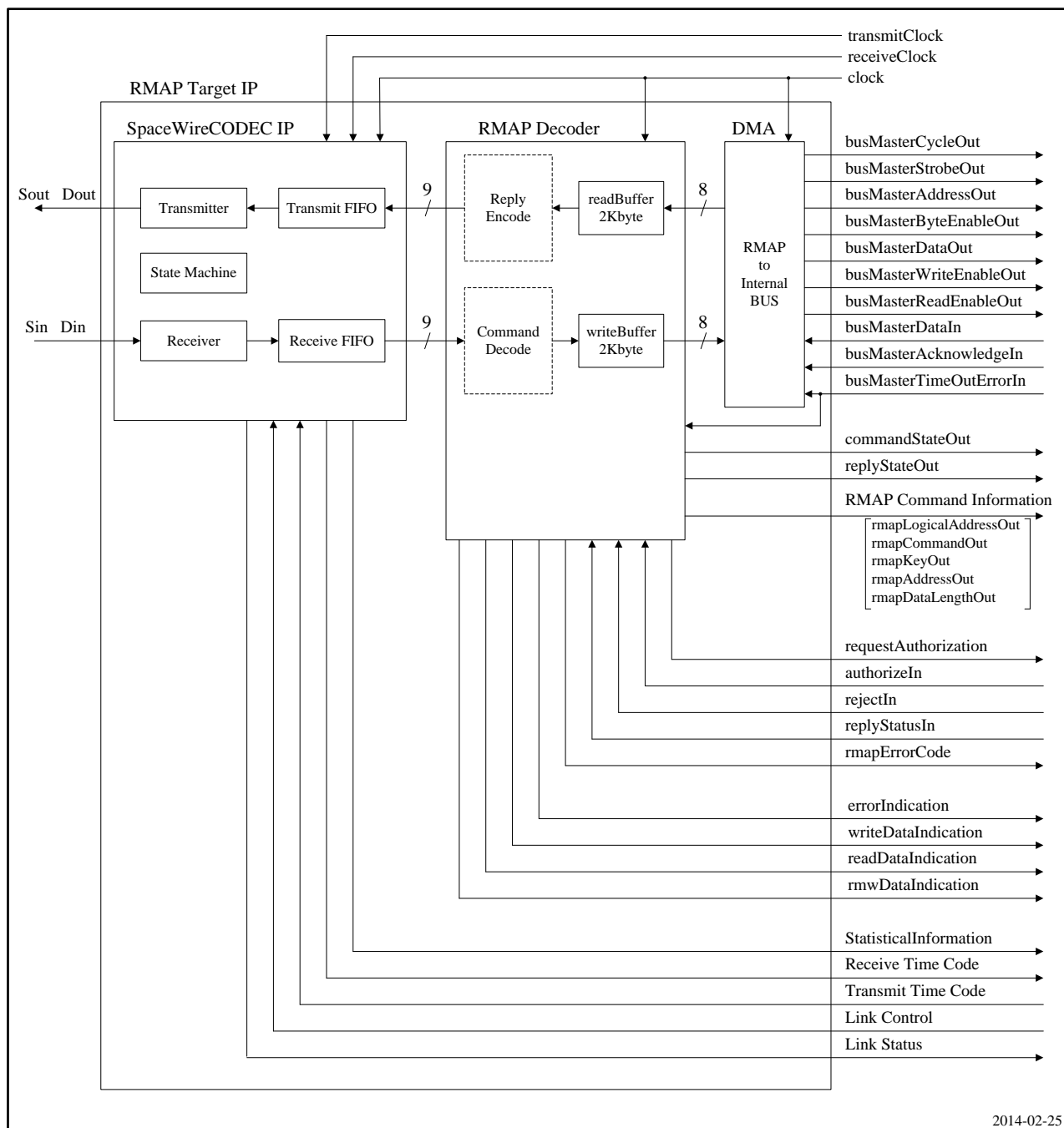
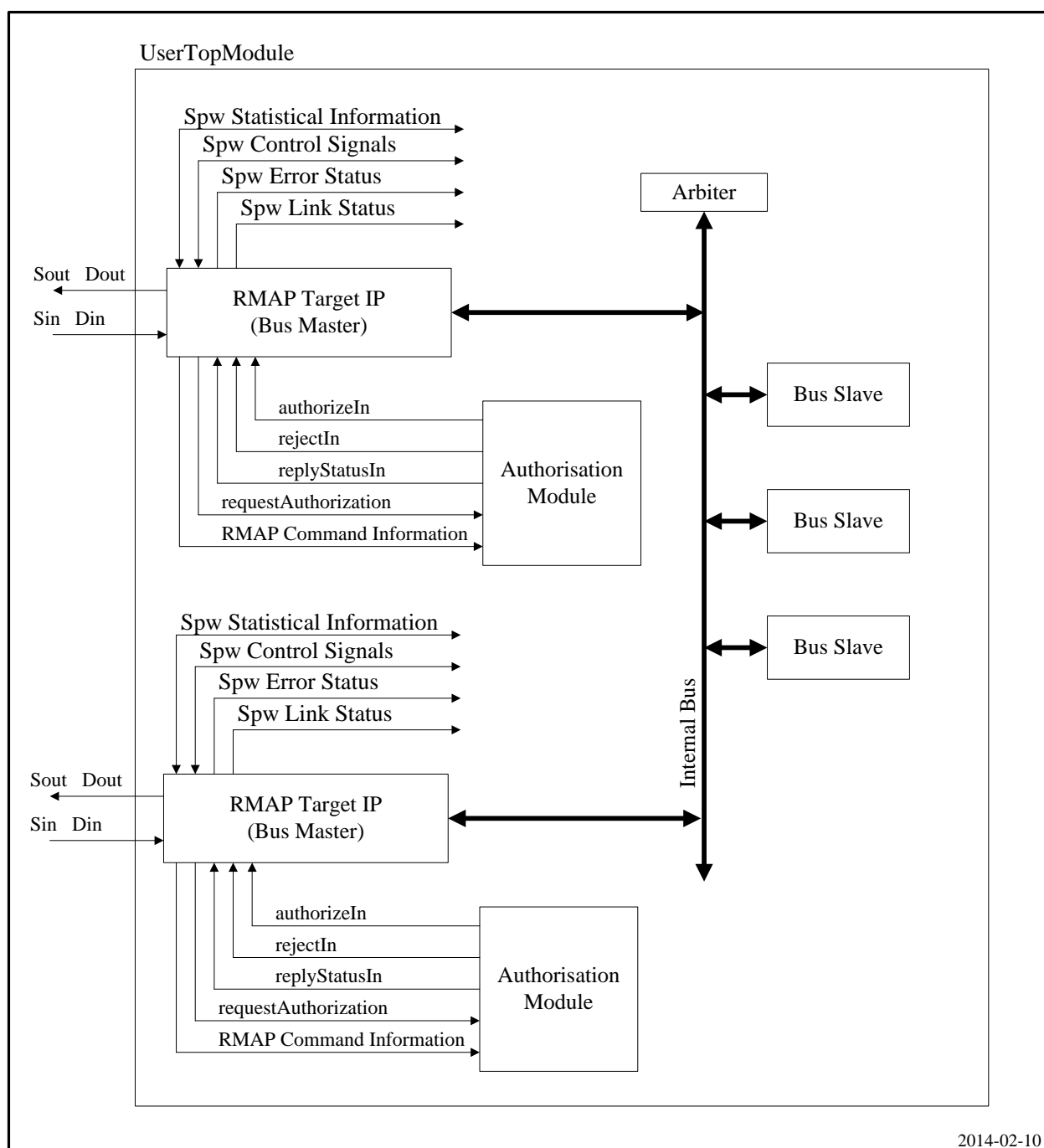


図 1 : RMAP Target IP ブロック図



2014-02-10

図 2 : RMAP Target IP を組み込んだシステム例

## 4. モジュールの機能

### 4.1. RMAP Target IP

RMAP Target IP のトップモジュールです。

SpaceWireCODECIP、RMAPDecoder、RMAPDMAController の各モジュールをインスタンスしています。

### 4.2. SpaceWireCODECIP

SpaceWire 信号の送受信モジュールです。

- SpaceWire ネットワークから受信した Data/Strobe 信号を 9bit パラレルデータに変換して、ReceiveFIFO に格納します。
- TransmittFIFO に格納された 9bit パラレルデータを Data/Strobe 信号に変換して、SpaceWire ネットワークへ送信します。

詳細は Open-source SpaceWire CODEC IP Core User Manual を参照してください。

### 4.3. RMAPDecoder

RMAP の Command パケットの解析と、Reply パケットの生成を行うモジュールです。

- ReceiveFIFO から受信データを読み出して、Command パケットの解析を行います。
- Command パケット解析の際には、ユーザー側に RMAP パケットの指定する logicalAddress、key、command、address、dataLength を渡し、この RMAP パケットを用いたメモリへのアクセス許可/不許可応答を待ちます。

### 4.4. RMAPDMAController

RMAPDecoder で解析された RMAP のアドレスとデータを内部バスに出力し、メモリや I/O に対して Read/Write アクセスをするモジュールです。

- RMAPDecoder から Write 命令を受けると、writeBuffer に書込まれているデータを内部バスを通して、メモリへ書き込みます。
- RMAPDecoder から Read 命令を受けると、内部バスを通してメモリからデータを読み出して、readBuffer に書き込みます。



## 5. 内部バス

内部バスは RMAPTargetIP とメモリ間のデータ転送に利用されます。

『図 1 : RMAP Target IP ブロック図』の DMAController から出力される【busMaster\*\*\*\*】という信号が内部バス信号です。

### 5.1. バス幅の設定

本 RMAP Target IP は RMAPTargetIPPackage.vhdl の cBusWidth の設定値を変更する事で、DMA モジュールから出力されるバス幅が変わります。

cBusWidth = 32 : 32bit バス  
cBusWidth = 16 : 16bit バス  
cBusWidth = 8 : 8bit バス

下記の表に各バス幅を設定した際に、変化する信号の bit 幅を示します。

入出力信号名	バス幅 8bit 設定時 有効データ範囲	バス幅 16bit 設定時 有効データ範囲	バス幅 32bit 設定時 有効データ範囲
busMasterDataOut	(7 downto 0)	(15 downto 0)	(31 downto 0)
busMasterDataOut	(7 downto 0)	(15 downto 0)	(31 downto 0)
busMasterByteEnableOut	(0 downto 0)	(1 downto 0)	(3 downto 0)

## 5.2. データアラインメント

### 5.2.1. バス幅 8bit

Address offset ⇒		--
		[bit7 : bit0]
Address	0x0000	0
	0x0001	1
	0x0002	2
	0x0003	3

### 5.2.2. バス幅 16bit

Address offset ⇒		[A0] = 0	[A0] = 1
		[bit15 : bit8]	bit7 : bit0
Address	0x0000	0	1
	0x0002	2	3
	0x0004	4	5
	0x0006	6	7

※ビッグエンディアン

### 5.2.3. バス幅 32bit

Address offset ⇒		[A1:A0] = 0,0	[A1:A0] = 0,1	[A1:A0] = 1,0	[A1:A0] = 1,1
		[bit31 : bit24]	[bit23 : bit16]	[bit15 : bit8]	[bit7 : bit0]
Address	0x0000	0	1	2	3
	0x0004	4	5	6	7
	0x0008	8	9	A	B
	0x000C	C	D	E	F

※ビッグエンディアン

## 6. RMAPTargetIPモジュールのインスタンス

- 内部バスの bit 幅の設定【cBusWidth】、及び Command パケットデコード用、Reply パケットエンコード用のステートマシンのタイプ宣言が含まれている、【RMAPTargetIPPackage】をライブラリに宣言しないとインスタンス出来ないの注意してください。
- 同様に、SpaceWire CODEC IP から出力される統計情報を格納する配列のタイプ宣言を行っている、【SpaceWireCODECIPPackage】もライブラリ宣言して下さい。

RMAP Target IP をインスタンスする際の記述例。

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

library work;
use work.SpaceWireCODECIPPackage.all;
use work.RMAPTargetIPPackage.all;

--
signal busMasterByteEnableOut : std_logic_vector ((cBusWidth/8) - 1 downto 0);
signal busMasterDataIn       : std_logic_vector (cBusWidth - 1      downto 0);
signal busMasterDataOut      : std_logic_vector (cBusWidth - 1      downto 0);
signal statisticalInformationClear : std_logic;
signal statisticalInformation   : bit32X8Array;

begin

M0 : RMAPTargetIP
generic map ( gBusWidth => cBusWidth )
port map (
    clock    => clock,
    reset    => reset,

    --
    busMasterByteEnableOut => busMasterByteEnableOut,
    busMasterDataIn       => busMasterDataIn      ,
    busMasterDataOut      => busMasterDataOut

    --
    statisticalInformationClear => statisticalInformationClear,
    statisticalInformation     => statisticalInformation
);
```

## 7. RMAPデータ長とアドレス

### 7.1. 命令ごとの取扱い可能な最大データ長

RMAP Target IP の書き込みバッファサイズは 2048byte です。

したがって、書き込み前照合コマンド命令（Verify Data Before Write）に対しては、2048byte 以上のデータを取り扱う事は出来ません。

以下にコマンド別の、受付可能な最大データ長の表を示します。

RMAP コマンド	Read/Write	書き込み前 照合の有無	Reply	アドレス	取扱い可能な 最大データ長
“0010”	Read	無し	有り	固定	16Mbyte
“0011”	Read	無し	有り	インクリメント	16Mbyte
“0111”	RMW	無し	有り	インクリメント	4byte
“1000”	Write	無し	無し	固定	16Mbyte
“1001”	Write	無し	無し	インクリメント	16M byte
“1010”	Write	無し	有り	固定	16M byte
“1011”	Write	無し	有り	インクリメント	16M byte
“1100”	Write	有り	無し	固定	2048byte
“1101”	Write	有り	無し	インクリメント	2048byte
“1110”	Write	有り	有り	固定	2048byte
“1111”	Write	有り	有り	インクリメント	2048byte

### 7.2. アクティブバイトレーン

#### 7.2.1. バス幅 8bit

バス幅が 8bit の場合は、アクセス開始アドレスとデータ長の組み合わせに関係なく、メモリへのアクセスが可能です。

#### 7.2.2. バス幅 16bit

下記の表に、バス幅 16bit 設定時の、バスのアクティブバイトレーンを示します。

データ長	アドレスオフセット [A0]	データ [15:8]	データ [7:0]
1 byte	0	✓	-
1 byte	1	-	✓
2 byte	0	✓	✓
2 byte	1	不許可	
(2 x N) byte	0	✓	✓
(2 x N) byte	1	不許可	
(2 x N) + 1 byte	0	不許可	
(2 x N) + 1 byte	1	不許可	

不許可の場合、リプライ有り設定ならば、エラーリプライをイニシエータに返します。

### 7.2.3. バス幅 32bit

下記の表に、バス幅 32bit 設定時の、バスのアクティブバイトレーンを示します。

データ長	アドレスオフセット [A1:A0]	データ [31:24]	データ [23:16]	データ [15:8]	データ [7:0]
1 byte	0	✓	-	-	-
1 byte	1	-	✓	-	-
1 byte	2	-	-	✓	-
1 byte	3	-	-	-	✓
2byte	0	✓	✓	-	-
2byte	1	不許可			
2byte	2	-	-	✓	✓
2byte	3	不許可			
3byte	0	不許可			
3byte	1				
3byte	2				
3byte	3				
4byte	0	✓	✓	✓	✓
4byte	1	不許可			
4byte	2	不許可			
4byte	3	不許可			
(4 x N) byte	0	✓	✓	✓	✓
(4 x N) byte	1	不許可			
(4 x N) byte	2	不許可			
(4 x N) byte	3	不許可			
(4 x N) + 1 byte	0~3	不許可			
(4 x N) + 2 byte	0~3	不許可			
(4 x N) + 3 byte	0~3	不許可			

不許可の場合、リプライ有り設定ならば、エラーリプライをイニシエータに返します。

## 8. データフロー

### Command 処理

- ① SpaceWire CODEC IP が SpaceWire データを受信すると ReceiveFIFO にデータパケットが格納される。
- ② RMAP Decoder モジュールが ReceiveFIFO から 9bit (Data Control Flag + データ) 単位でデータを抜き出して、RMAP Command のデコードを行う。
- ③ ヘッダーCRC まで RMAP Command のデコードが正常に進むと、LogicalAddress、Command、Key、Address、Data Length の情報とトランザクション要求信号 (requestAuthorization) をユーザー側へ送る。
- ④ ユーザー側からトランザクション承認信号 (authorizeIn) が返されるまで、PMAP パケットのデコードは一時停止する。
- ⑤ ユーザー側からトランザクション承認信号 (authorizeIn) が返されると、Write の場合はデータ部分のデコードが開始される。  
RMAP パケットで指定されたアドレスやデータ長で、メモリへアクセスすることをユーザー側が許可しない場合は、ユーザー側はトランザクション拒否信号 (rejectIn) を返す。  
トランザクション拒否の場合、ステータスコード (replyStatusIn) は、0x00 以外のエラーコードを出力する。
- ⑥ Write コマンドの場合、  
RMAP データのデコード終了後、データは writeBuffer を通して DMA モジュールへ渡され内部バスを通してメモリに Write される。  
リプライ要求があれば、ステータス情報をリプライする。

Read コマンドの場合、指定されたアドレスのメモリやレジスタから、DMA モジュールがバスを通してデータを読み出す。

読み出したデータは、readBuffer を通して RMAP Decoder モジュールへ渡され Reply パケットが生成される。

### Reply 処理

- ⑦ RMAP Decoder モジュールで、Reply 先アドレス、ステータス情報、Reply データなどから、Reply パケットが生成されて SpaceWire CODEC IP の TransmitFIFO へ書き込まれる。  
TransmitFIFO へ書き込まれたデータは SpaceWire CODEC IP の Transmitter でパラレルからシリアルに変換されて送信される。

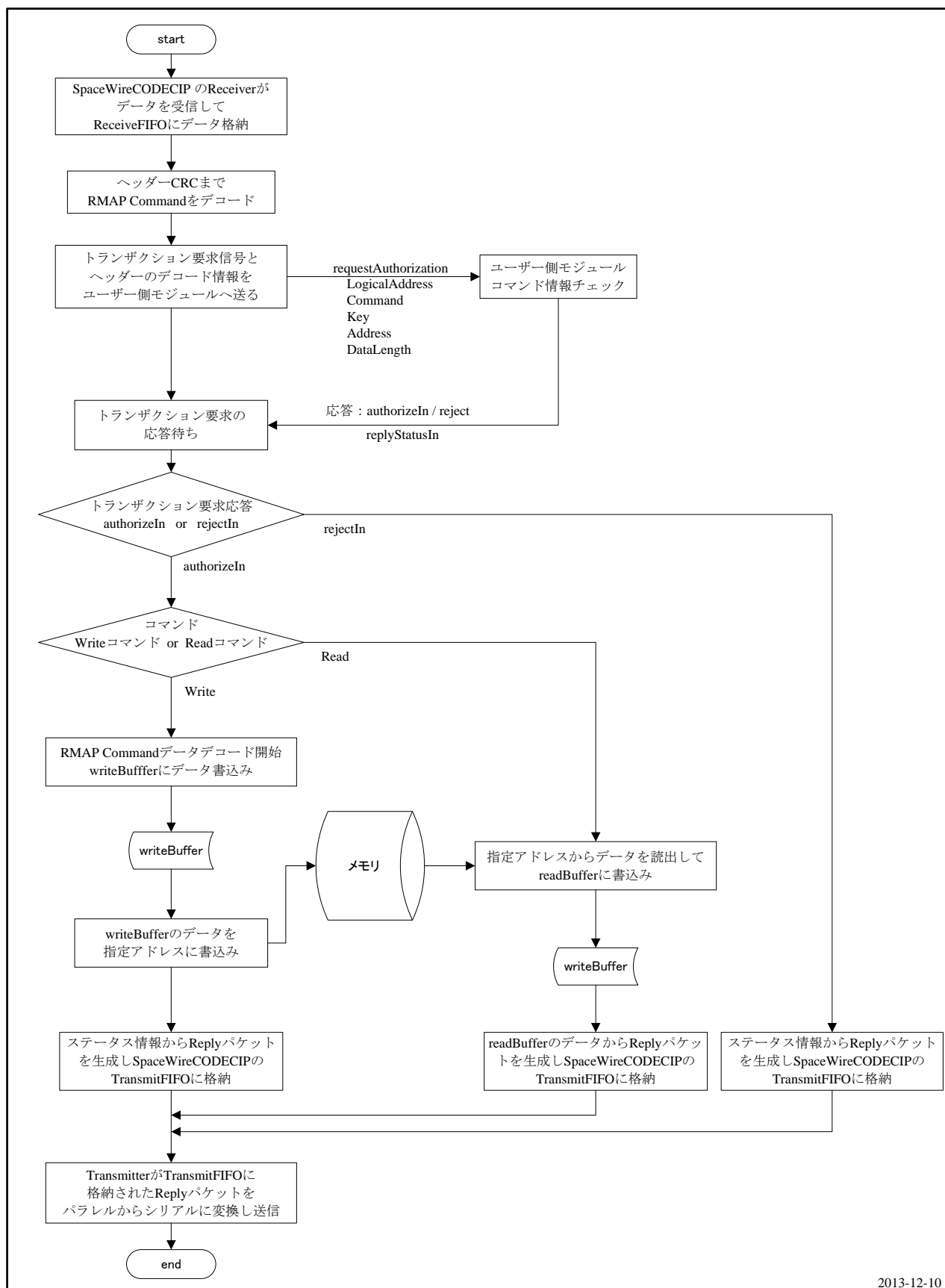


図 3 : RMAP デコード処理フローチャート

## 9. インターフェース信号

SpaceWire RMAP Target IP は各 Clock の立ち上がりエッジで動作します。

入力信号は“H”にする事で、その機能が有効になります。

出力信号は“H”の時に、その機能が有効になっています。

SpaceWire RMAP Target IP インターフェース信号を以下に示します。

### 9.1. Generic

信号名	I/O	説明
gBusWidth (integer range 8 to 32)	I	バスのバス幅を設定します。 32 = バス幅 32bit 16 = バス幅 16bit 8 = バス幅 8bit

### 9.2. SpaceWire Control Signal

信号名	I/O	説明
clock	I	システムクロック RMAP Decoder、RMAP DMA Controllerの動作クロックです。 DMAから入出力されるバスの動作クロックとなります。
transmitClock	I	SpaceWire 送信クロック
receiveClock	I	SpaceWire 受信クロック
reset	I	リセット
spaceWireDataIn	I	SpaceWire CODEC IPに対するSpaceWireData入力
spaceWireStrobeIn	I	SpaceWire CODEC IPに対するSpaceWireStrobe入力
spaceWireDataOut	O	SpaceWire CODEC IPのSpaceWireData出力
spaceWireStrobeOut	O	SpaceWire CODEC IPのSpaceWireStrobe出力
tickIn	I	タイムコード送信
timeIn[5:0]	I	送信タイムコード
controlFlagsIn[1:0]	I	送信タイムコード制御フラグ
tickOut	O	受信タイムコード
timeOut[5:0]	O	受信タイムコード
controlFlagsOut[1:0]	O	受信タイムコード制御フラグ
linkStart	I	“H”を入力する事で LinkStart を有効にします。 リンクアップシーケンスは自動的に Ready から Started に遷移します。 “L”を入力する事で LinkStart を無効にします。
linkDisable	I	“H”を入力する事でリンクを Disable にします。 “L”を入力する事で Disable を解除します。 LinkStart 及び AutoStart は LinkDisable が“L”の時のみ有効になります。
autoStart	I	“H”を入力する事で AutoStart を有効にします。 NULL を受信する事でリンクが始動するようになります。 “L”を入力する事で AutoStart を無効にします。
transmitClockDivideValue [5:0]	I	リンクアップ後の転送レート設定値 $\text{転送レート} = \frac{\text{TransmitClock}[MHz]}{(\text{TransmitClockDivideValue} + 1)}$



### 9.3. SpaceWire Link Status

信号名	I/O	説明
linkStatus[15:0]	O	リンクステータス情報
linkStatus[0]		EnableTransmit リンクアップシーケンスが <b>Started</b> に遷移すると“H”になります。 リンクダウン又はリンクアップシーケンス中のエラーで“L”になります。
linkStatus[1]		EnableReceive リンクアップシーケンスが <b>ErrorWait</b> に遷移すると“H”になります。 リンクダウン又はリンクアップシーケンス中のエラーで“L”になります。
linkStatus[2]		SendNULLs リンクアップシーケンスが <b>Started</b> に遷移すると“H”になります。 リンクダウン又はリンクアップシーケンス中のエラーで“L”になります。
linkStatus[3]		SendFCTs リンクアップシーケンスが <b>Connecting</b> に遷移すると“H”になります。 リンクダウン又はリンクアップシーケンス中のエラーで“L”になります。
linkStatus[4]		SendNChar リンクアップシーケンスが <b>Run</b> に遷移すると“H”になります。 リンクダウン又はリンクアップシーケンス中のエラーで“L”になります。
linkStatus[5]		SendTimeCode リンクアップシーケンスが <b>Run</b> に遷移すると“H”になります。 リンクダウン又はリンクアップシーケンス中のエラーで“L”になります。
linkStatus[6]		Reseve “0”固定
linkStatus[7]		SpaceWireReset リンクアップシーケンスが <b>ErrorReset</b> に遷移すると“H”になります。 リンクアップシーケンスが <b>ErrorWait</b> に遷移すると“L”になります。
linkStatus[8]		リンク確立信号 リンクアップシーケンスが <b>Run</b> に遷移すると“H”になります。 リンクダウンすると“L”になります。
linkStatus[9]		Transmitter が N-Char を送信すると 1 クロックだけ“H”になります。
linkStatus[10]		Receiver が N-Char を受信すると 1 クロックだけ“H”になります。
linkStatus[11]		Receiver が NULL を受信すると 1 クロックだけ“H”になります。
linkStatus[12]		Receiver が FCT を受信すると 1 クロックだけ“H”になります。
linkStatus[13]		Receiver が EOP を受信すると 1 クロックだけ“H”になります。
linkStatus[14]		Receiver が EEP を受信すると 1 クロックだけ“H”になります。
linkStatus[15]		Reseve “0”固定

## 9.4. SpaceWire Error Status

信号名	I/O	説明
errorStatus[7:0]	O	リンクエラー情報
errorStatus[0]		CharacterSequenceError リンクアップシーケンス中に予期しない FCT や N-Char を受信すると“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[1]		CreditError CreditCount が 56 以上になった時又は OutstandingCount が 0 の時に N-Char を受信すると“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[2]		RxErr EnableReceive が“H”の時に ParityError 又は DisconnectError 又は ESCError を検知すると“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[3]		Reseve “0”固定
errorStatus[4]		Parity Error EnableReceive が“H”の時にパリティエラーを検知すると“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[5]		DisconnectError EnableReceive が“H”の時にデータ及びストロブ信号が 850ns 間変化がないと“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[6]		ESCError EnableReceive が“H”の時に ESC キャラクターの後に受信した制御キャラクターが FCT 以外の時に“H”になります。 リンクアップシーケンスが ErrorReset に遷移すると“L”になります。
errorStatus[7]		Reseve “0”固定

## 9.5. SpaceWire Statistical Information

信号名	I/O	説明
statisticalInformationClear	I	“H”を入力する事で統計情報がクリアされます。 “L”を入力する事でカウントを開始します。
statisticalInformation 8*[31:0]	O	統計情報
statisticalInformation 0 [31:0]		Transmitter が EOP を送信した積算回数
statisticalInformation 1 [31:0]		Receiver が EOP を受信した積算回数
statisticalInformation 2 [31:0]		Transmitter が EEP を送信した積算回数
statisticalInformation 3 [31:0]		Receiver が EEP を受信した積算回数
statisticalInformation 4 [31:0]		Transmitter が N-Char を送信した積算回数
statisticalInformation 5 [31:0]		Receiver が N-Char を受信した積算回数
statisticalInformation 6 [31:0]		リンクアップシーケンスがリンクアップした積算回数
statisticalInformation 7 [31:0]		リンクアップシーケンスがリンクダウンした積算回数

## 9.6. RMAP Command Decode Information

以下の信号は、ユーザー側のモジュールが RMAP Target IP の動作状況（デコードした RMAP Command 情報やエラー情報）を確認する為の信号です。

信号名	I/O	説明
commandStateOut	O	RMAPDecoder モジュールの、Command パケットデコード用ステートマシンの状態を出力します。
replyStateOut	O	RMAPDecoder モジュールの、Reply パケットエンコード用ステートマシンの状態を出力します。
rmapLogicalAddressOut [7:0]	O	RMAP の Command パケットをデコードして得られた、ターゲット論理アドレス。 ユーザー側でのコマンドチェック用。
rmapCommandOut [7:0]	O	RMAP の Command パケットをデコードして得られた、RAMP コマンド。 ユーザー側でのコマンドチェック用。
rmapKeyOut [7:0]	O	RMAP の Command パケットをデコードして得られた、RAMP Key。 ユーザー側でのコマンドチェック用。
rmapAddressOut [31:0]	O	RMAP の Command パケットをデコードして得られた、RMAP アドレス。 ユーザー側でのコマンドチェック用。
rmapDataLengthOut [23:0]	O	RMAP の Command パケットをデコードして得られた、データ長。 ユーザー側でのコマンドチェック用。
requestAuthorization	O	トランザクション要求信号 ヘッダーCRC までデコード終了後、上記 5 つの RMAP コマンドの設定でメモリにアクセスしてよいか、ユーザーに許可を求めます。
authorizeIn	I	トランザクション承認信号 ユーザー側で RMAP コマンド情報を確認後、メモリへのアクセスを許可する場合、”H”を出力します。 ユーザー側で RMAP コマンド情報の確認が必要ない場合は、常に”H”にしてください。
rejectIn	I	トランザクション拒否信号 ユーザー側で RMAP コマンド情報を確認後、メモリへのアクセスを拒絶する場合、”H”を出力します。 ”H”の場合、RMAP Command でリプライが要求されていれば、イニシエータにエラーリプライを返します。
replyStatusIn [7:0]	I	requestAuthorization を受けたユーザー側で RMAP コマンドを解析した結果のステータスコード。 メモリへのアクセスを許可する場合は 0x00。 メモリへのアクセスを拒絶する場合はユーザーが任意のエラーコードを設定してください。
rmapErrorCode [7:0]	O	RMAPDecoder で最終的に採用されたステータスコード。 エラーが無い場合は 0x00。 エラーが有る場合は仕様書に準拠したエラーコード又は、ユーザー側から送られてきたエラーコード（replyStatusIn）の値が入ります。  Header CRC エラーはユーザー側には一般エラー（0x01）として通知されます。

## 9.7. BUS Signals

信号名	I/O	説明
busMasterCycleOut	O	サイクル信号
busMasterStrobeOut	O	ストロブ信号
busMasterAddressOut [31:0]	O	アドレス信号
busMasterByteEnableOut [3:0] or [1:0] or [0:0]	O	読み書きデータバイトイネーブル
busMasterDataIn [31:0] or [15:0] or [7:0]	I	メモリからの読み込みデータ
busMasterDataOut [31:0] or [15:0] or [7:0]	O	メモリへの書き込みデータ
busMasterWriteEnableOut	O	Write/Read選択 1 = Write 0 = Read
busMasterReadEnableOut	O	Read/Write選択 1 = Read 0 = Write busMasterWriteEnableOutと機能が被るのでOpenでよい。
busMasterAcknowledgeIn	I	バス動作完了
busMasterTimeOutErrorIn	I	バス動作タイムアウト 長時間Ackが返ってこない場合や、メモリへのアクセスに失敗した場合、ユーザー側から任意のタイミングで入力する。 エラーコード0x01が出力される。 1 = タイムアウト発生 0 = タイムアウト未発生

※ サイクル信号とストロブ信号は同じタイミングで、出力されているので、どちらか片方を見て制御すればよい。

## 9.8. Indication Signals

信号名	I/O	説明
errorIndication	O	エラー通知信号 Write,Read,RMWの何れかのコマンド実行時に、メモリへのアクセス許可【authorizeIn = '1'】を受けた後、メモリアccessに失敗すると、clockに同期したワンショット信号を出力。
writeDataIndication	O	データ書き込み通知 Writeコマンド動作時にメモリへのWriteが終了したら、clockに同期したワンショット信号を出力。
readDataIndication	O	データ読出し通知 Readコマンド動作時にメモリからのReadが終了したら、clockに同期したワンショット信号を出力。
rmwDataIndication	O	リード・モディファイ・ライト通知 RMWコマンド動作時にメモリの読み書きが終了したら、clockに同期したワンショット信号を出力。

## 10. 内部バス信号動作タイミング

RMAP Target IP の内部バス信号のタイミングチャートを示します。

### 10.1. Write 動作

バス幅 32bit の RMAP Target IP が、ユーザー側のメモリやレジスタに対して Write 動作を行う際のタイミングチャートを下記に示します。

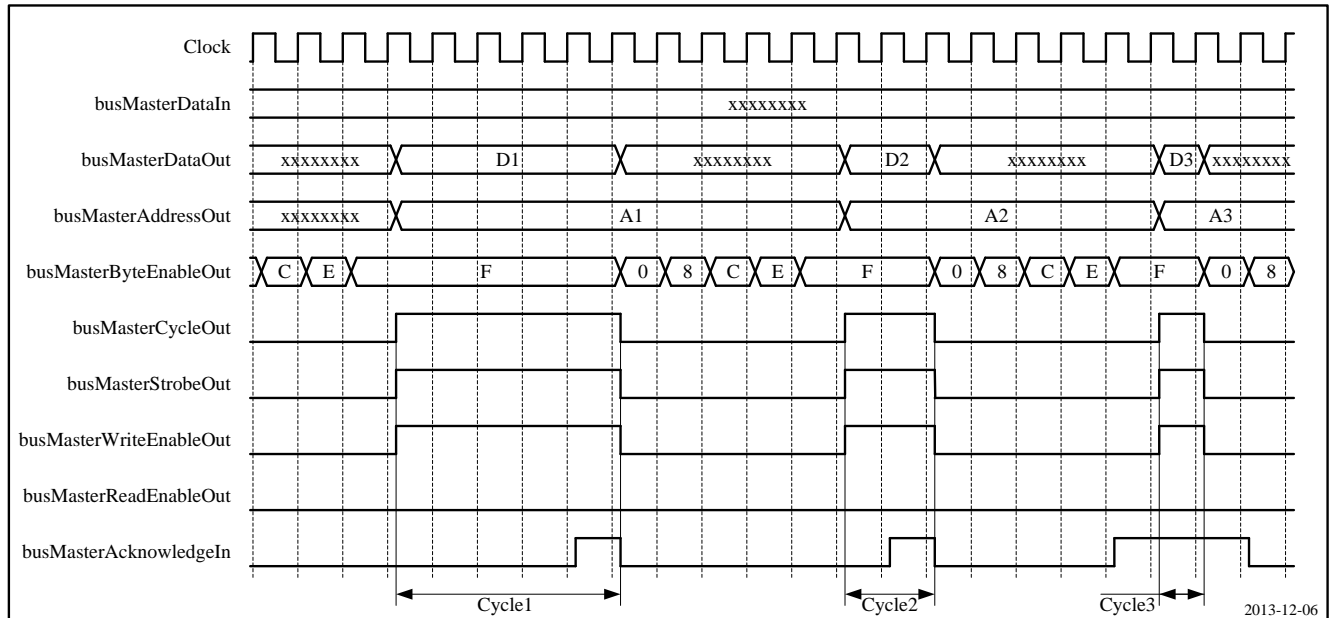


図 4 : バス幅 32bit 設定の RMAPTargetIP による Write アクセス タイミングチャート

- ① busMasterDataOut、busMasterAddressOut、busMasterByteEnableOut、busMasterWriteEnableOut が有効値になると共に、busMasterStrobeOut が”H”になる。
- ② ユーザー側で busMasterDataOut の値を busMasterAddressOut で指定されるアドレスヘデータを写込む。
- ③ ユーザー側で busMasterAcknowledgeIn を”H”にして、データ書込が終了したことを RMAP Target IP に伝える。
- ④ busMasterStrobeOut が”L”になる。

図 4 の busMasterAcknowledgeIn の各タイミング Cycle1,2,3 は下記の状況での busMasterAcknowledgeIn の応答を表しています。

Cycle1 : 4 ウェイトサイクル  
 Cycle2 : 1 ウェイトサイクル  
 Cycle3 : ノーウェイト

※ バースト転送には対応していません。

## 10.2. Read 動作

バス幅 32bit の RMAP Target IP が、ユーザー側のメモリやレジスタに対して Read 動作を行う際のタイミングチャートを下記に示します。

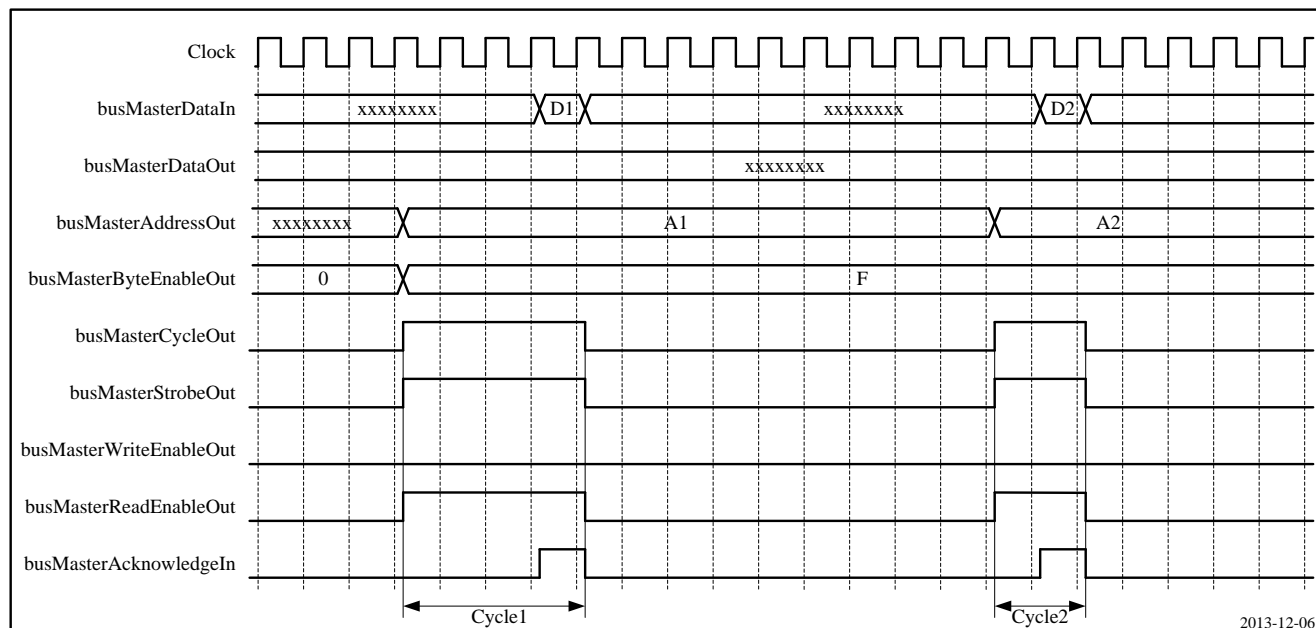


図 5：バス幅 32bit 設定の RMAPTargetIP による Read アクセス タイミングチャート

- ① busMasterAddressOut、busMasterByteEnableOut、busMasterReadEnableOut が有効値になると共に、busMasterStrobeOut が”H”になる。
- ② ユーザー側が busMasterAddressOut で指定されたアドレスからデータを読み出して、busMasterDataIn へ出力する。
- ③ ユーザー側で busMasterAcknowledgeIn を”H”にして、データ読出しが終了したことを RMAP Target IP に伝える。
- ④ busMasterStrobeOut が”L”になる。

図 5の busMasterAcknowledgeIn の各タイミング Cycle1,2 は下記の状況での busMasterAcknowledgeIn の応答を表しています。

Cycle1： 3 ウェイトサイクル

Cycle2： 1 ウェイトサイクル

※ バースト転送には対応していません。

### 10.3. トランザクション許可取得タイミング

RMAP Target IP がユーザー側にメモリへのアクセス許可を取る際に入出力される RMAP Command 情報のタイミングチャートを以下に示します。

※ 図中の各信号は clock に同期した出力です。

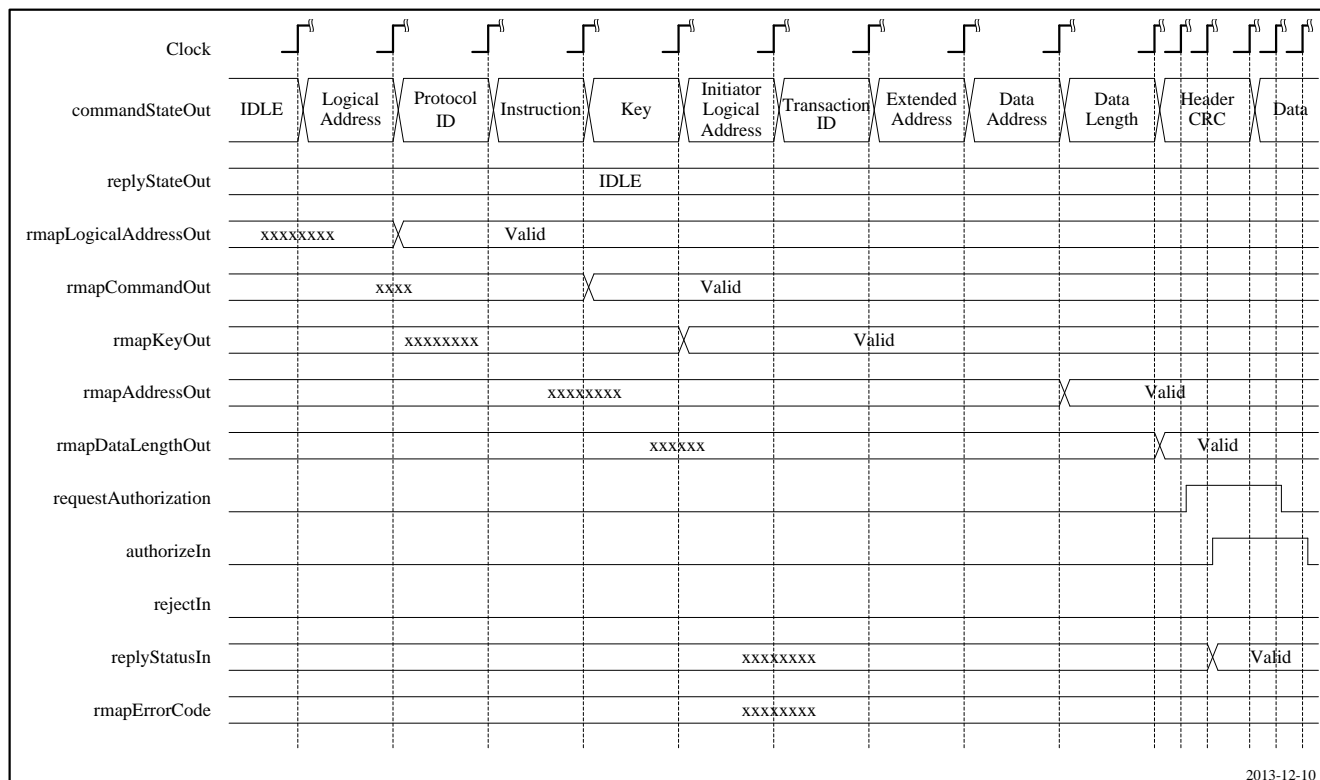


図 6：メモリアクセス許可取得 タイミングチャート 1

- ① RMAPCommand の LogicalAddress フィールドのデコードが終了すると、rmapLogicalAddressOut 信号が有効になります。rmapKeyOut 信号や InitiatorLogicalAddress 信号も同様に対応する RMAPCommand フィールドのデコードが終了すると、それぞれ値が有効になります。
- ② 受信した HeaderCRC の確認まで正常にデコードが進むと、RMAP Target IP はユーザーモジュールに対してメモリアクセスの許可を取る為に、トランザクション要求信号 (requestAuthorization) を”H”にします。  
この時点で、RMAPCommand のデコードは一時停止します。
- ③ ユーザー側モジュールでは、RMAP Target IP から送られてきた RMAPCommand 情報を確認して、アドレスやデータ長の組み合わせに問題が無ければ、トランザクション承認信号 (authorizeIn) を”H”にして下さい。そうでなければトランザクション拒否信号 (rejectIn) を”H”にして下さい。  
トランザクション承認信号 (authorizeIn) が”H”になると RMAPCommand のデコードが再開します。
- ④ トランザクション要求信号 (requestAuthorization) は、commandStateOut が commandStateData、commandStateDataEOP、又は commandStateWaitEOP に遷移したら”L”に落ちます。  
トランザクション要求信号 (requestAuthorization) に連動させて、トランザクション承認信号 (authorizeIn) 又はトランザクション拒否信号 (rejectIn) を、”L”に落としてください。
- ⑤ commandStateOut が IDLE に到達すると、rmapErrorCode の値が有効になります。

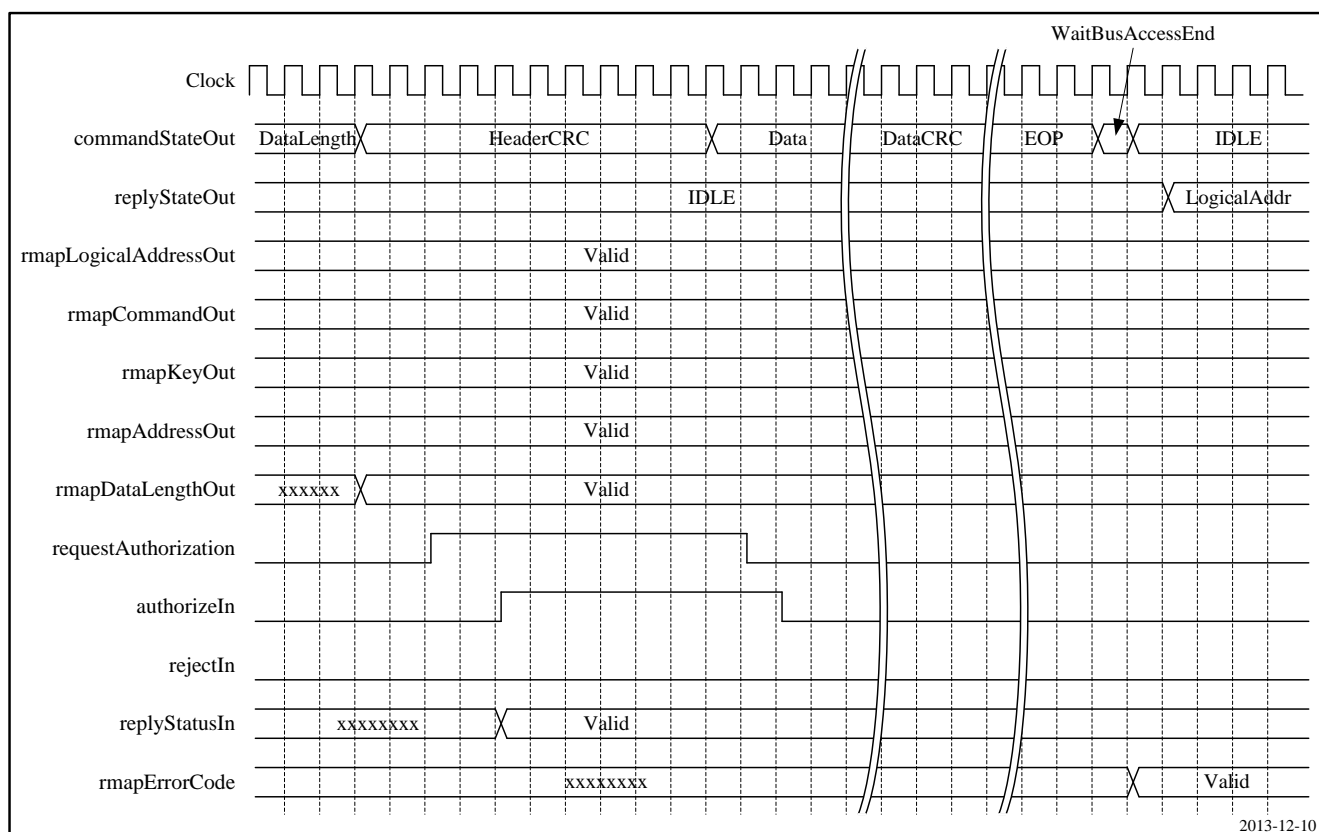


図 7：メモリアクセス許可取得 タイミングチャート 2



## 11. パッケージ

RMAPTargetIPPackage 内で宣言されている宣言を以下に示します。

### 11.1. Constant 宣言

Constant 名	type	Bit 幅	説明
cBusWidth	integer	8 to 32	AHB のバス幅を設定します。 RMAP Target IP (AHB版)はバス幅 32bit固定なので、32を設定します。 通常gBusWidthにアサインして使用します。
cRMAPCRCRevision	std_logic	1	CRC リビジョン 0 = リビジョン E 1 = リビジョン F デフォルト値は 1 に設定されています。
cUseDevice	integer	0 to 1	使用デバイスの設定 0 = Altera 1 = Xilinx デフォルト値は 1 に設定されています

### 11.2. ステートマシン宣言

		説明
StateMachine 名	commandStateMachine	現在デコード中の RMAPCommand フィールドを表すステートマシンです。
State 名	commandStateIdle	デコード停止中
	commandStateTargetLogicalAddress	【ターゲット論理アドレス】
	commandStateProtocolID	【プロトコル ID】
	commandStateInstruction	【インストラクション】
	commandStateKey	【キー】
	commandStateReplyAddress	【リプライアドレス】
	commandStateInitiatorLogicalAddress	【イニシエータ論理アドレス】
	commandStateTransactionID	【トランザクション ID】
	commandStateExtendedAddress	【拡張アドレス】
	commandStateAddress	【アドレス】
	commandStateDataLength	【データ長】
	commandStateHeaderCRC	【ヘッダーCRC】
	commandStateData	【データ】
	commandStateDataCRC	【データ CRC】
	commandStateDataEOP	【データ】
	commandStateWaitEOP	【EOP】
	commandStateWaitBusAccessEnd	バスのアクセスが終了するまで本ステートで、待機します。

## 12. ファイル構成

RMAP Target IP ファイル構成を以下に示します。

ファイル名	説明
RMAPTargetIP.vhdl	RMAP Target IP トップモジュール
RMAPTargetIPDecoder.vhdl	RMAP デコード モジュール
RMAPTargetIPDMAController.vhdl	RMAP パケット <--> バス信号変換モジュール
RMAPTargetIPCRCRomXilinx.vhdl	XilinxIP 用 CRCROM ラッパーファイル
RMAPTargetIPFIFO8x2kXilinx.vhdl	XilinxIP 用 2048byte FIFO バッファラッパーファイル
RMAPTargetIPCRCRomAltera.vhdl	AlteraIP 用 CRCROM ラッパーファイル
RMAPTargetIPFIFO8x2kAltera.vhdl	AlteraIP 用 2048byte FIFO バッファラッパーファイル
RMAPTargetIPPackage.vhdl	バス幅、CRC リビジョン、使用 FPGA 選択用 constant 宣言パッケージ
crcRomXilinx.xco	CRC ROM
crcData.coe	CRC ROM データ
FIFO8x2kXilinx.xco	バッファFIFO
crcRomAltera.vhd	CRC ROM
crcData.hex	CRC ROM データ
FIFO8x2kAltera.vhd	バッファFIFO

※ 上記ソースの文字コードは UTF-8 です。

※ プロジェクトを作成する際に以下の点に注意してください。

- Xilinx のプロジェクトに、Altera 用のファイルを Add しないで下さい。
- Altera のプロジェクトに、Xilinx 用のファイルを Add しないで下さい。

## 13. パフォーマンス

SpaceWire Router IP のリソースの使用率を以下に示します。

- デバイス Xilinx 製 Spartan-6 XC6SLX100T
- 合成ツール ISE 14.6

リソースの使用率は合成オプションやデバイスによって異なります。

モジュール名	Flip-Flops	LUT	Slices	分散 RAM (LUT 内)	Block RAM	
					16BWER	8BWER
RMAPTargetIP	2071	2084	1186	36	2	2
SpaceWireCODECIP	699	722	363	36	0	0
RMAPTargetIPDecoder	751	891	449	0	2	2
RMAPDMAController	548	753	432	0	0	0

## 14. タイミング制約

### 14.1. ucf ファイル

タイミング制約は、合成回路の正確なタイミングを確保するために必要とされます。  
全てのクロック乗せ換えにはタイミング制約が必要です。

SpaceWire 通信レート 100Mbps、内部バス動作速度 50MHz 時のタイミング制約を以下に示します。

```
clock = 50MHz
```

```
transmitClock = 100MHz
```

```
receiveClock = 166.6MHz
```

```
NET "clock"          TNM_NET = "SYS_CLK";
```

```
NET "transmitClock"  TNM_NET = "TX_CLK";
```

```
NET "receiveClock"TNM_NET = "RX_CLK";
```

```
TIMESPEC TS_SYS_CLK_to_TX_CLK = FROM "SYS_CLK" TO "TX_CLK" TIG;
```

```
TIMESPEC TS_TX_CLK_to_SYS_CLK = FROM "TX_CLK" TO "SYS_CLK" TIG;
```

```
TIMESPEC TS_TX_CLK_to_RX_CLK  = FROM "TX_CLK" TO "RX_CLK" TIG;
```

```
TIMESPEC TS_RX_CLK_to_TX_CLK  = FROM "RX_CLK" TO "TX_CLK" TIG;
```

```
TIMESPEC TS_SYS_CLK_to_RX_CLK = FROM "SYS_CLK" TO "RX_CLK" TIG;
```

```
TIMESPEC TS_RX_CLK_to_SYS_CLK = FROM "RX_CLK" TO "SYS_CLK" TIG;
```

```
TIMESPEC "TS_clk"      = PERIOD "SYS_CLK" 20 ns HIGH 50 %;
```

```
TIMESPEC "TS_txclk"    = PERIOD "TX_CLK " 10 ns HIGH 50 %;
```

```
TIMESPEC "TS_rxclk"    = PERIOD "RX_CLK " 6 ns  HIGH 50 %;
```

## 14.2. クロック

RMAP Target IP には Clock、TransmitClock、ReceiveClock を供給する必要があります。

clock : バスの動作クロック  
transmitClock : SpaceWire 送信クロック  
receiveClock : SpaceWire 受信クロック

SpaceWire の通信レート 100Mbps 時の推奨クロックを以下に示します。

Clock	50-100MHz
TransmitClock	100.0MHz
ReceiveClock	166.6MHz

入力クロックの条件

- ReceiveClock は受信レートの 1.5 倍程度の周波数を入力してください。
- TransmitClock を 100MHz 以下で使用する時は ReceiveClock も同じ比率で下げてください。  
詳細は Open-source SpaceWire CODEC IP Core User Manual を参照してください。

例：TransmitClock を 50MHz で使用する時。

Clock	50.0MHz
TransmitClock	50.0MHz
ReceiveClock	83.3MHz