




# Arora V BSRAM & SSRAM ユーザーガイド

UG300-1.3.2J, 2024-02-02

## 著作権について(2024)

著作権に関する全ての権利は、**Guangdong Gowin Semiconductor Corporation** に留保されています。

 **GOWIN** 及び **Gowin** は、当社により、中国、米国特許商標庁、及びその他の国において登録されています。商標又はサービスマークとして特定されたその他全ての文字やロゴは、それぞれの権利者に帰属しています。何れの団体及び個人も、当社の書面による許可を得ず、本文書の内容の一部もしくは全部を、いかなる視聴覚的、電子的、機械的、複写、録音等の手段によりもしくは形式により、伝搬又は複製をしてはなりません。

## 免責事項

当社は、**GOWINSEMI Terms and Conditions of Sale**(GOWINSEMI取引条件)に規定されている内容を除き、(明示的か又は黙示的に拘わらず)いかなる保証もせず、また、知的財産権や材料の使用によりあなたのハードウェア、ソフトウェア、データ、又は財産が被った損害についても責任を負いません。当社は、事前の通知なく、いつでも本文書の内容を変更することができます。本文書を参照する何れの団体及び個人も、最新の文書やエラッタ(不具合情報)については、当社に問い合わせる必要があります。

## バージョン履歴

日付	バージョン	説明
2023/04/20	1.0J	初版。
2023/05/25	1.1J	<ul style="list-style-type: none"><li>● シングルポート・モードおよびデュアルポート・モードでの <b>Read-before-Write</b> のサポートを削除。</li><li>● デュアルポート・モード、セミ・デュアルポート・モード、および <b>ECC</b> 機能付きのセミ・デュアルポート・モードの注記を追加。</li></ul>
2023/06/30	1.2J	<b>138K</b> デバイスの、シングルポート・モードおよびデュアルポート・モードでの <b>Read-before-Write</b> のサポートを削除。
2023/09/08	1.3J	図 5-1 RAM16S1 モードのタイミング図および図 5-5 RAM16SDP1 モードのタイミング図を更新。
2023/12/12	1.3.1J	<b>Read-before-Write</b> モードの説明( <b>Arora V 138K</b> デバイスが <b>Read-before-Write</b> モードをサポートしません)を追加。
2024/02/02	1.3.2J	<ul style="list-style-type: none"><li>● 表 5-1 <b>SSRAM</b> のモードに注記を追加、一部の <b>SSRAM</b> プリミティブをサポートしないデバイスに関する情報を追加。</li><li>● <b>Read-before-Write</b> モードの説明(<b>Arora V 25K</b> デバイスがデュアルポート・モードでの <b>Read-before-Write</b> をサポートしません)を追加。</li></ul>

# 目次

目次.....	i
図一覧.....	iii
表一覧.....	v
<b>1 本マニュアルについて .....</b>	<b>1</b>
1.1 マニュアルの内容.....	1
1.2 関連ドキュメント.....	1
1.3 用語、略語.....	2
1.4 テクニカル・サポートとフィードバック.....	2
<b>2 概要.....</b>	<b>3</b>
2.1 BSRAM の特性.....	3
2.2 BSRAM の構成モード.....	4
<b>3 BSRAM プリミティブ.....</b>	<b>7</b>
3.1 デュアルポート・モード.....	7
3.2 シングルポート・モード.....	19
3.3 セミ・デュアルポート・モード.....	24
3.4 ECC 機能付きのセミ・デュアルポート・モード.....	31
3.5 ROM モード.....	39
<b>4 BSRAM 出力のリセット.....</b>	<b>45</b>
<b>5 SSRAM プリミティブ.....</b>	<b>48</b>
5.1 RAM16S1.....	48
5.2 RAM16S2.....	51
5.3 RAM16S4.....	53
5.4 RAM16SDP1.....	56
5.5 RAM16SDP2.....	58
5.6 RAM16SDP4.....	61
5.7 ROM16.....	63
<b>6 IP の呼び出し.....</b>	<b>66</b>
6.1 デュアルポート・モードの BSRAM.....	66

---

6.2 ECC 機能付きのセミ・デュアルポート・モードの BSRAM.....	69
6.3 シングルポート・モードの SSRAM .....	71
<b>7 初期化ファイル .....</b>	<b>74</b>
7.1 2 進数形式(Bin File).....	74
7.2 16 進数形式(Hex File) .....	74
7.3 アドレス付き 16 進数(Address-Hex File).....	75

## 図一覧

図 3-1 DPB/DPX9B Normal 書き込みモードのタイミング図 (Bypass 読み出しモード) .....	8
図 3-2 DPB/DPX9B Normal 書き込みモードのタイミング図 (Pipeline 読み出しモード) .....	9
図 3-3 DPB/DPX9B Write-through 書き込みモードのタイミング図 (Bypass 読み出しモード) .....	10
図 3-4 DPB/DPX9B Write-through 書き込みモードのタイミング図 (Pipeline 読み出しモード) .....	11
図 3-5 DPB/DPX9B Read-before-Write 書き込みモードのタイミング図 (Bypass 読み出しモード) .....	12
図 3-6 DPB/DPX9B Read-before-Write 書き込みモードのタイミング図(Pipeline 読み出しモード) .....	12
図 3-7 DPB/DPX9B のポート図 .....	14
図 3-8 SP/SPX9 のポート図 .....	21
図 3-9 セミ・デュアルポート BSRAM の normal 書き込みモードのタイミング図 (Bypass 読み出しモード).....	25
図 3-10 セミ・デュアルポート BSRAM の normal 書き込みモードのタイミング図 (Pipeline 読み出しモード).....	26
図 3-11 SDPB/SDPX9B のポート図.....	27
図 3-12 SDP36KE のポート図 .....	33
図 3-13 ROM のタイミング図(Bypass モード).....	40
図 3-14 ROM のタイミング図 (Pipeline モード) .....	40
図 3-15 pROM/pROMX9 のポート図 .....	41
図 4-1 出力リセットのブロック図 .....	45
図 4-2 同期リセットのタイミング図(Pipeline モード) .....	46
図 4-3 同期リセットのタイミング図(Bypass モード) .....	46
図 4-4 非同期リセットのタイミング図(Pipeline モード).....	46
図 4-5 非同期リセットのタイミング図(Bypass モード).....	47
図 5-1 RAM16S1 モードのタイミング図.....	49
図 5-2 RAM16S1 のポート図 .....	49
図 5-3 RAM16S2 のポート図 .....	51
図 5-4 RAM16S4 のポート図 .....	54
図 5-5 RAM16SDP1 モードのタイミング図.....	56
図 5-6 RAM16SDP1 のポート図.....	56
図 5-7 RAM16SDP2 のポート図.....	59
図 5-8 RAM16SDP4 のポート図.....	61

図 5-9 ROM16 モードのタイミング図 .....	63
図 5-10 ROM16 のポート図 .....	64
図 6-1 DPB IP の構成ウィンドウ .....	67
図 6-2 SDP36KE IP の構成ウィンドウ .....	70
図 6-3 RAM16S IP の構成ウィンドウ .....	72

## 表一覧

表 1-1 用語、略語 .....	2
表 2-1 BSRAM の構成モード一覧.....	4
表 2-2 BSRAM のデータ幅とアドレス幅の対応関係.....	4
表 2-3 デュアルポート・モードにおけるデータ幅.....	5
表 2-4 セミ・デュアルポート・モードにおけるデータ幅.....	5
表 2-5 ECC 機能付きのセミ・デュアルポート・モードにおけるデータ幅 .....	6
表 3-1 DPB/DPX9B のデータ幅とアドレス幅の対応関係 .....	13
表 3-2 DPB/DPX9B のポートの説明 .....	14
表 3-3 DPB/DPX9B のパラメータの説明.....	15
表 3-4 SP/SPX9 のデータ幅とアドレス幅の対応関係.....	20
表 3-5 SP/SPX9 のポートの説明 .....	21
表 3-6 SP/SPX9 のパラメータの説明 .....	21
表 3-7 SDPB/SDPX9B のデータ幅とアドレス幅の対応関係 .....	26
表 3-8 SDPB/SDPX9B のポートの説明 .....	27
表 3-9 SDPB/SDPX9B のパラメータの説明.....	28
表 3-10 SDP36KE のデータ幅とアドレス幅の対応関係 .....	32
表 3-11 SDP36KE のポートの説明 .....	33
表 3-12 SDP36KE のパラメータの説明.....	34
表 3-13 pROM/pROMX9 のデータ幅とアドレス幅の対応関係 .....	40
表 3-14 pROM/pROMX9 のポートの説明 .....	41
表 3-15 pROM/pROMX9 のパラメータの説明.....	42
表 5-1 SSRAM のモード .....	48
表 5-2 RAM16S1 のポートの説明.....	50
表 5-3 RAM16S1 のパラメータの説明 .....	50
表 5-4 RAM16S2 のポートの説明.....	52
表 5-5 RAM16S2 のパラメータの説明 .....	52
表 5-6 RAM16S4 のポートの説明.....	54
表 5-7 RAM16S4 のパラメータの説明 .....	54
表 5-8 RAM16SDP1 のポートの説明.....	57



表 5-9 RAM16SDP1 のパラメータの説明 .....	57
表 5-10 RAM16SDP2 のポートの説明.....	59
表 5-11 RAM16SDP2 のパラメータの説明.....	59
表 5-12 RAM16SDP4 のポートの説明.....	61
表 5-13 RAM16SDP4 のパラメータの説明 .....	62
表 5-14 ROM16 のポートの説明.....	64
表 5-15 ROM16 のパラメータの説明 .....	64

# 1 本マニュアルについて

## 1.1 マニュアルの内容

このマニュアルは、主に GOWIN セミコンダクター Arora V FPGA の BSRAM と SSRAM の特性、動作モード、プリミティブ、及び IP の呼び出しなどについて説明します。

## 1.2 関連ドキュメント

GOWIN セミコンダクターの公式 Web サイト [www.gowinsemi.com/ja](http://www.gowinsemi.com/ja) から、以下の関連ドキュメントがダウンロード、参考できます：

- Arora V シリーズ FPGA 製品データシート([DS981](#))
- Gowin ソフトウェア ユーザーガイド([SUG100](#))

## 1.3 用語、略語

表 1-1 に、本マニュアルで使用される用語、略語、及びその意味を示します。

表 1-1 用語、略語

用語、略語	正式名称	意味
BSRAM	Block SRAM	ブロック SRAM
CFU	Configurable Function Unit	コンフィギュラブル機能ユニット
DP	True Dual Port 16K Block SRAM	16K のデュアルポート BSRAM
ECC	Error Checking and Correction	誤り検出と訂正
ROM	Read-Only Memory	読み出し専用メモリ
SDP	Semi Dual Port 16K Block SRAM	16K のセミ・デュアルポート BSRAM
SDP36KE	Semi Dual Port 36K Block SRAM with ECC function	ECC 機能付きの 36K のセミ・デュアルポート BSRAM
SP	Single Port 16K Block SRAM	16K のシングルポート BSRAM
SSRAM	Shadow SRAM	分散 SRAM

## 1.4 テクニカル・サポートとフィードバック

GOWIN セミコンダクターは、包括的な技術サポートをご提供しています。使用に関するご質問、ご意見については、直接弊社までお問い合わせください。

ホームページ : [www.gowinsemi.com/ja](http://www.gowinsemi.com/ja)

E-mail : [support@gowinsemi.com](mailto:support@gowinsemi.com)

# 2 概要

Gowin セミコンダクターの Arora V FPGA 製品には、ブロック SRAM(BSRAM)と分散 SRAM(SSRAM)を含む豊富なメモリリソースがあります。

各 BSRAM は最大 36K ビットに構成でき、そのデータ幅やアドレスの深さも構成可能です。各 BSRAM には、A ポートと B ポートの 2 つの独立したポートがあります。この 2 つのポートには独立したクロック、アドレス、データ、及び制御信号があるため、個別に読み出し/書き込みを行うことができます。なお、この 2 つのポートは 1 つのメモリ領域を共有します。

Arora V FPGA の基本構成要素であるコンフィギュラブル機能ユニット (CFU) は、アプリケーションシナリオに応じて、16 x 4 ビットの SRAM と ROM(ROM16)を含む SSRAM として構成できます。

## 2.1 BSRAM の特性

- 各 BSRAM の最大容量は 18K ビット
- クロック周波数は最大 380MHz(Read-before-Write<sup>[1]</sup>モードの場合は 230MHz)
- シングルポート・モード(SP)をサポート
- デュアルポート・モード(DP)をサポート
- セミ・デュアルポート・モード(SDP)をサポート
- ECC 機能付きのセミ・デュアルポート・モード (SDP36KE) をサポート
- 読み出し専用モード(ROM)をサポート
- 最大 72 ビットのデータ幅をサポート
- デュアルポート・モードとセミ・デュアルポート・モードは、独立した読み出し/書き込みクロックと独立したデータ幅をサポート
- 読み出しはレジスタ出力またはバイパス出力をサポート

- 書き込みは Normal モード、Read-before-Write<sup>[1]</sup>モード、および Write-through モードをサポート

注記：

- <sup>[1]</sup> Arora V 138K デバイスは Read-before-Write モードをサポートしません。
- <sup>[1]</sup> Arora V 25K デバイスはデュアルポート・モードでの Read-before-Write をサポートしません

## 2.2 BSRAM の構成モード

各 BSRAM(SDP36KE を除く)は 16K ビットまたは 18K ビットに構成でき、SDP36KE は 36K ビットをサポートします。5 つのモードの構成可能なアドレス深さとデータ幅を表 2-1 に示します。

表 2-1 BSRAM の構成モード一覧

容量	シングルポート・モード	デュアルポート・モード	セミ・デュアルポート・モード	ECC 機能付きのセミ・デュアルポート・モード	ROM モード
16Kbits	16K x 1	16K x 1	16K x 1	–	16K x 1
	8K x 2	8K x 2	8K x 2	–	8K x 2
	4K x 4	4K x 4	4K x 4	–	4K x 4
	2K x 8	2K x 8	2K x 8	–	2K x 8
	1K x 16	1K x 16	1K x 16	–	1K x 16
	512 x 32	–	512 x 32	–	512 x 32
18Kbits	2K x 9	2K x 9	2K x 9	–	2K x 9
	1K x 18	1K x 18	1K x 18	–	1K x 18
	512 x 36	–	512 x 36	–	512 x 36
36Kbits	–	–	–	1K x 36	–
	–	–	–	512 x 72	–

SDP36KE 以外の各 BSRAM のアドレスの幅は 14 ビット(すなわち、AD[13:0])であるため、最大アドレス深さは 16,384 になります。SDP36KE のアドレスの幅は 10 ビット(すなわち、AD[9:0])であるため、最大アドレス深さは 1024 になります。データ幅とアドレス幅の対応関係は表 2-2 に示す通りです。

表 2-2 BSRAM のデータ幅とアドレス幅の対応関係

容量	構成モード	データ幅	アドレス深さ	アドレス幅
16Kbits	16K x 1	[0:0]	16,384	[13:0]
	8K x 2	[1:0]	8,192	[13:1]
	4K x 4	[3:0]	4,096	[13:2]
	2K x 8	[7:0]	2,048	[13:3]
	1K x 16	[15:0]	1,024	[13:4]

容量	構成モード	データ幅	アドレス深さ	アドレス幅
	512 x 32	[31:0]	512	[13:5]
18Kbits	2K x 9	[8:0]	2,048	[13:3]
	1K x 18	[17:0]	1,024	[13:4]
	512 x 36	[35:0]	512	[13:5]
36Kbits	1K x 36	[35:0]	1024	[9:0]
	512 x 72	[71:0]	512	[9:1]

デュアルポート・モード、セミ・デュアルポート・モード、および ECC 機能付きのセミ・デュアルポート・モードの書き込みクロックと読み出しクロックは独立しており、独立した読み出し/書き込みのデータ幅がサポートされています。デュアルポート・モードでは、A ポートと B ポートがサポートするデータ幅は表 2-3 に示す通りです。セミ・デュアルポート・モードでは、A ポートと B ポートがサポートするデータ幅は表 2-4 に示す通りです。ECC 機能付きのセミ・デュアルポート・モードでは、A ポートと B ポートがサポートするデータ幅は表 2-5 に示す通りです。

表 2-3 デュアルポート・モードにおけるデータ幅

容量	B ポート	A ポート						
		16K x 1	8K x 2	4K x 4	2K x 8	1K x 16	2K x 9	1K x 18
16Kbits	16K x 1	Yes	Yes	Yes	Yes	Yes	N/A	N/A
	8K x 2	Yes	Yes	Yes	Yes	Yes	N/A	N/A
	4K x 4	Yes	Yes	Yes	Yes	Yes	N/A	N/A
	2K x 8	Yes	Yes	Yes	Yes	Yes	N/A	N/A
	1K x 16	Yes	Yes	Yes	Yes	Yes	N/A	N/A
18Kbits	2K x 9	N/A	N/A	N/A	N/A	N/A	Yes	Yes
	1K x 18	N/A	N/A	N/A	N/A	N/A	Yes	Yes

表 2-4 セミ・デュアルポート・モードにおけるデータ幅

容量	B ポート	A ポート										
		16K x 1	8K x 2	4K x 4	2K x 8	1K x 16	512 x 32	2K x 9	1K x 18	512 x 36	1K x 36	512 x 72
16Kbits	16K x 1	Yes	Yes	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A
	8K x 2	Yes	Yes	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A
	4K x 4	Yes	Yes	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A
	2K x 8	Yes	Yes	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A
	1K x 16	Yes	Yes	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A
	512 x 32	Yes	Yes	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	N/A
18Kbits	2K x 9	N/A	N/A	N/A	N/A	N/A	N/A	Yes	Yes	Yes	N/A	N/A
	1K x 18	N/A	N/A	N/A	N/A	N/A	N/A	Yes	Yes	Yes	N/A	N/A

表 2-5 ECC 機能付きのセミ・デュアルポート・モードにおけるデータ幅

容量	B ポート	A ポート	
		1K x 36	512 x 72
36Kbits	1K x 36	Yes	N/A
	512 x 72	N/A	Yes

# 3 BSRAM プリミティブ

Block SRAM は、静的アクセス機能を備えたブロック状のスタティック RAM です。BSRAM の特性によれば、シングルポート・モード (SP/SPX9)、デュアルポート・モード (DPB/DPX9B)、セミ・デュアルポート・モード (SDPB/SDPX9B)、ECC 機能付きのセミ・デュアルポート・モード (SDP36KE)、および読み出し専用モード (pROM/pROMX9) に分類できます。

## 3.1 デュアルポート・モード

### プリミティブの紹介

DPB/DPX9B (True Dual Port 16K Block SRAM/True Dual Port 18K Block SRAM) は、16K/18K のデュアルポート BSRAM です。

### 機能の説明

DPB/DPX9B はそれぞれメモリ領域が 16K ビット/18K ビットであるデュアルポート・モードの BSRAM です。A ポートと B ポートは個別に読み出し/書き込みを実現できます<sup>[1]</sup>。2 つの読み出しモード (Bypass モードと Pipeline モード) と 3 つの書き込みモード (Normal モード、Read-before-Write モード<sup>[2]</sup>、Write-through モード) がサポートされます。

#### 注記：

- [1] 同じアドレスに対して、同時に読み出しと書き込みを行うことは推奨されません。
- [2] 138K デバイスは Read-before-Write モードをサポートしません。
- [2] Arora V 25K デバイスはデュアルポート・モードでの Read-before-Write をサポートしません。
- 読み出しモード  
パラメータの READ\_MODE0、READ\_MODE1 は、A ポートおよび B ポートの出力 pipeline レジスタを有効または無効にするために使用されます。出力 pipeline レジスタを使用する場合、読み出しには追加のクロックサイクルが必要です。
- 書き込みモード



Normal モード、Write-through モード、Read-before-Write モードがあります。A ポートおよび B ポートの書き込みモードは、それぞれパラメータ `WRITE_MODE0` および `WRITE_MODE1` によって構成されます。各モードに対応する内部タイミング波形を図 3-1~図 3-4 に示します。

図 3-1 DPB/DPX9B Normal 書き込みモードのタイミング図 (Bypass 読み出しモード)

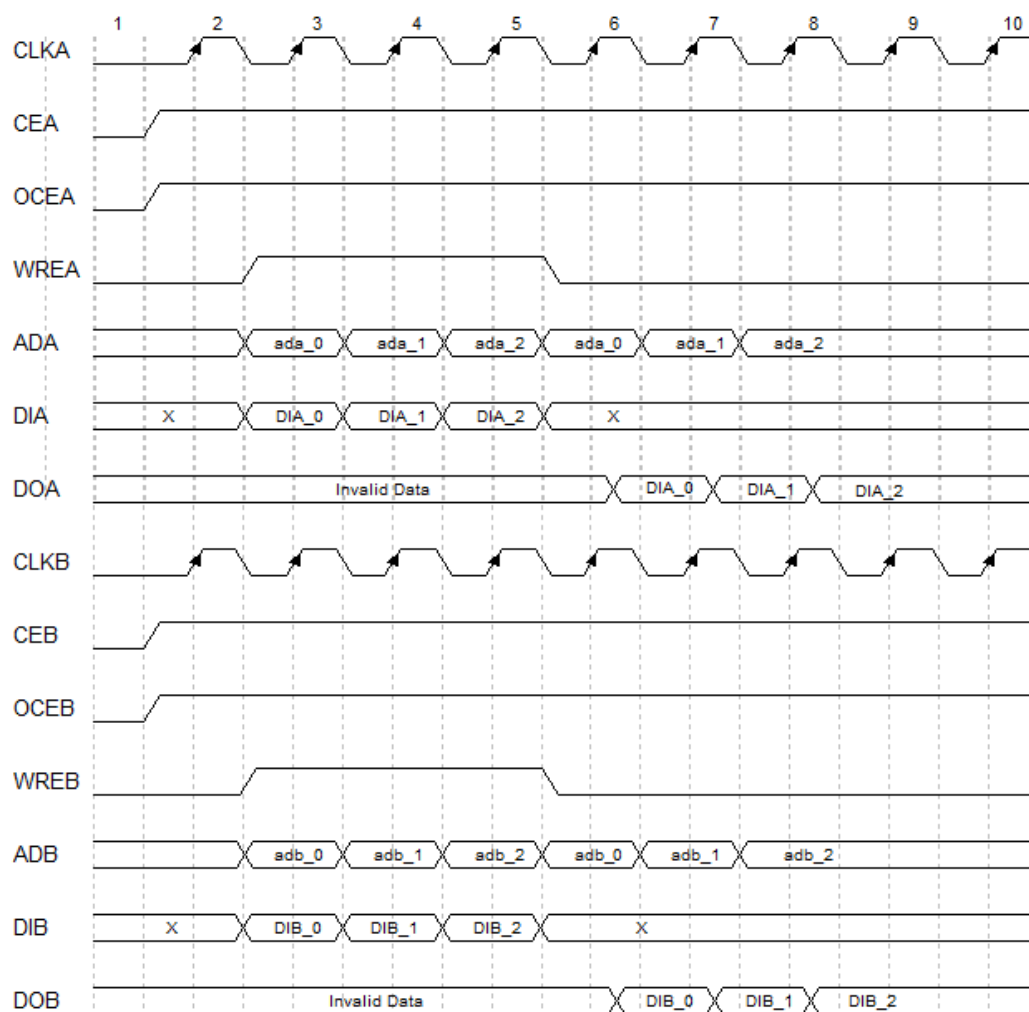


図 3-2 DPB/DPX9B Normal 書き込みモードのタイミング図 (Pipeline 読み出しモード)

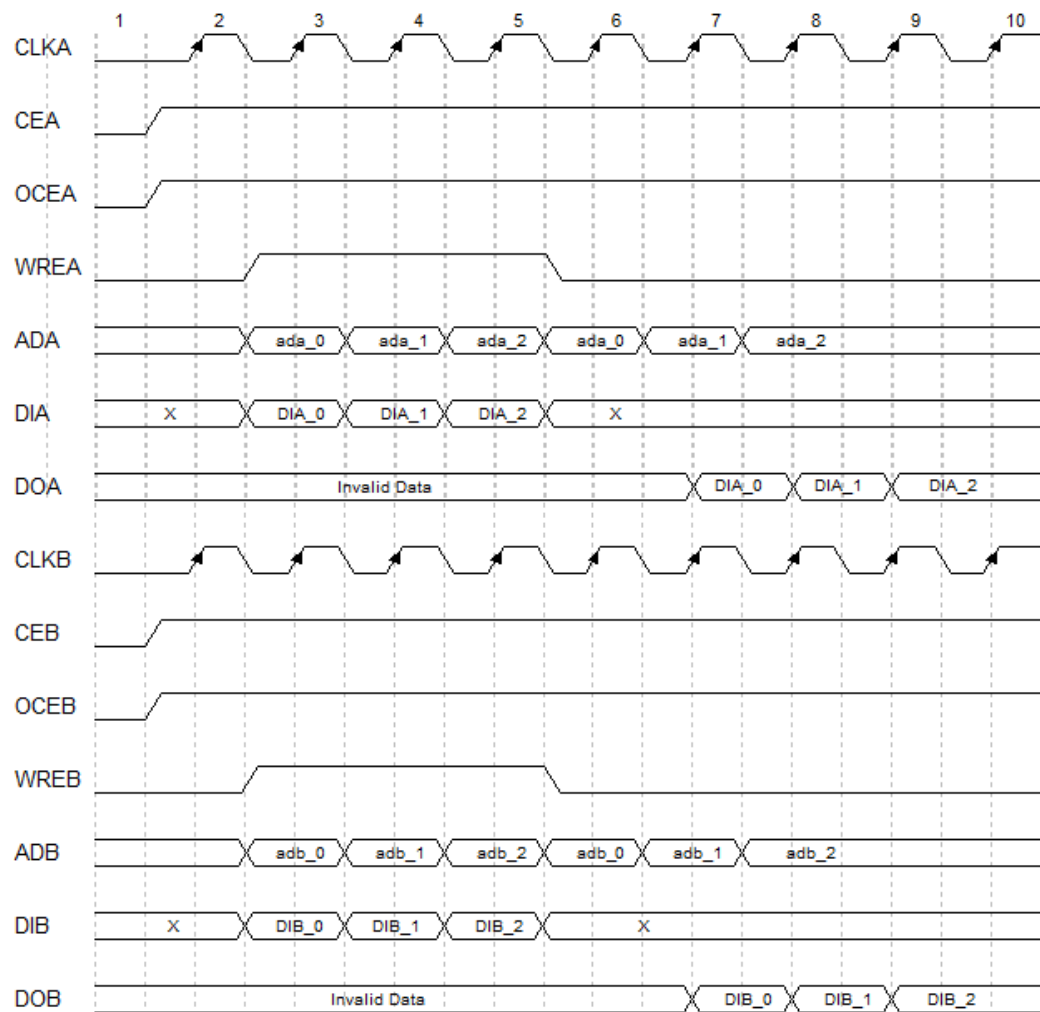


図 3-3 DPB/DPX9B Write-through 書き込みモードのタイミング図 (Bypass 読み出しモード)

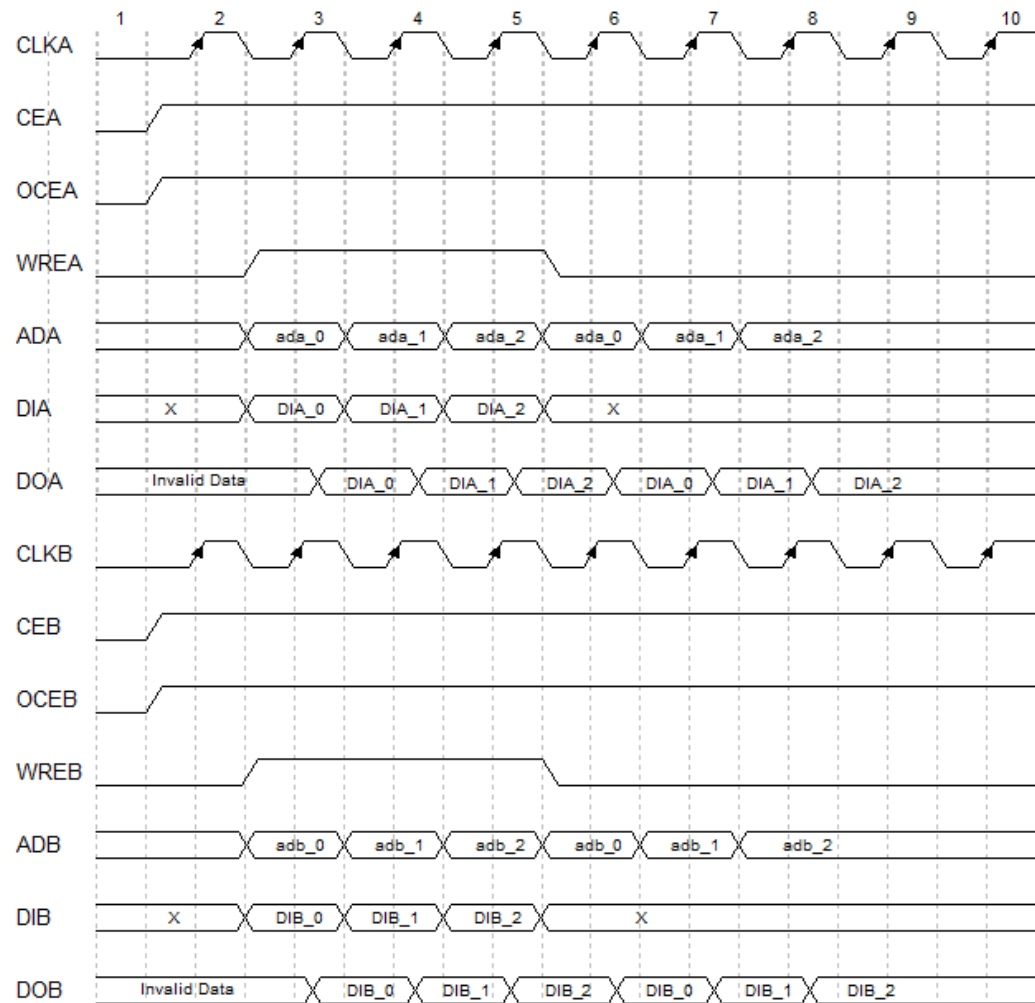


図 3-4 DPB/DPX9B Write-through 書き込みモードのタイミング図 (Pipeline 読み出しモード)

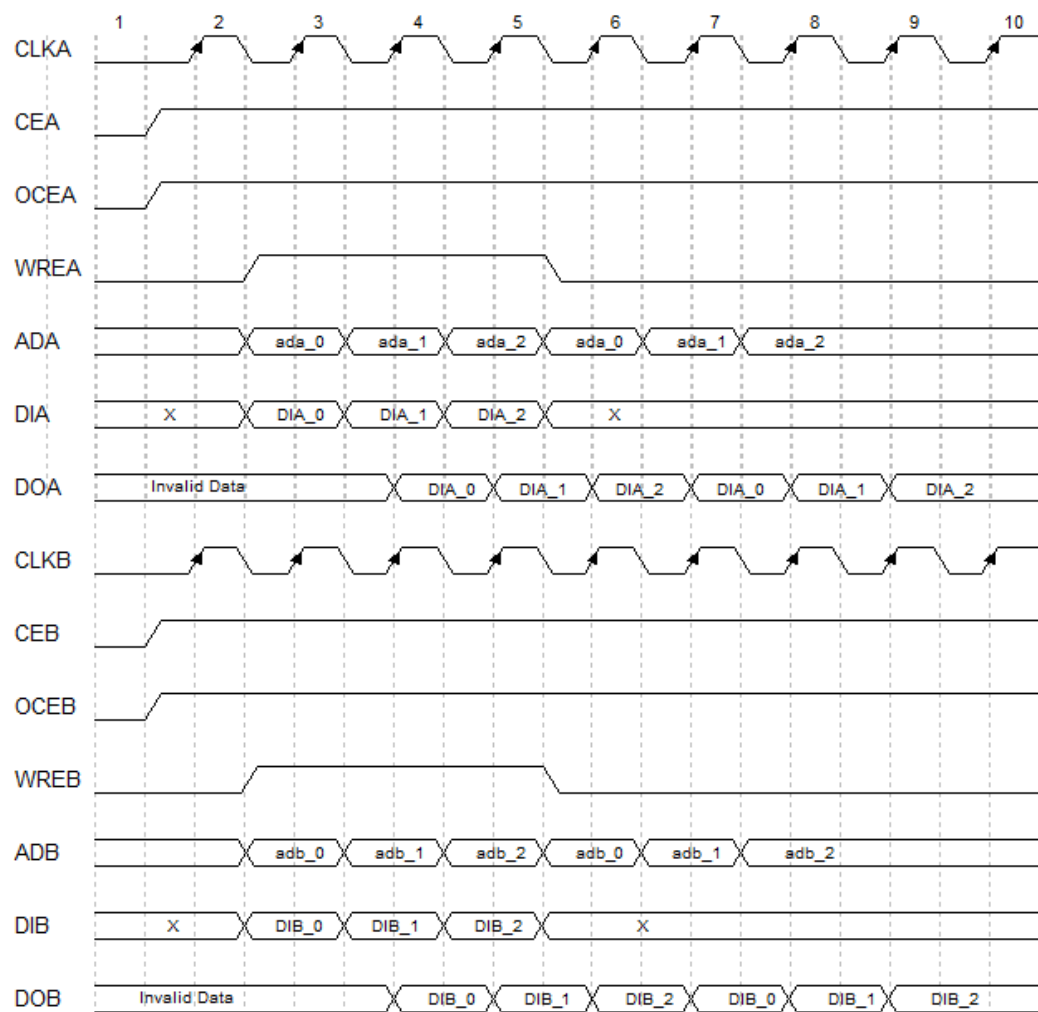


図 3-5 DPB/DPX9B Read-before-Write 書き込みモードのタイミング図 (Bypass 読み出しモード)

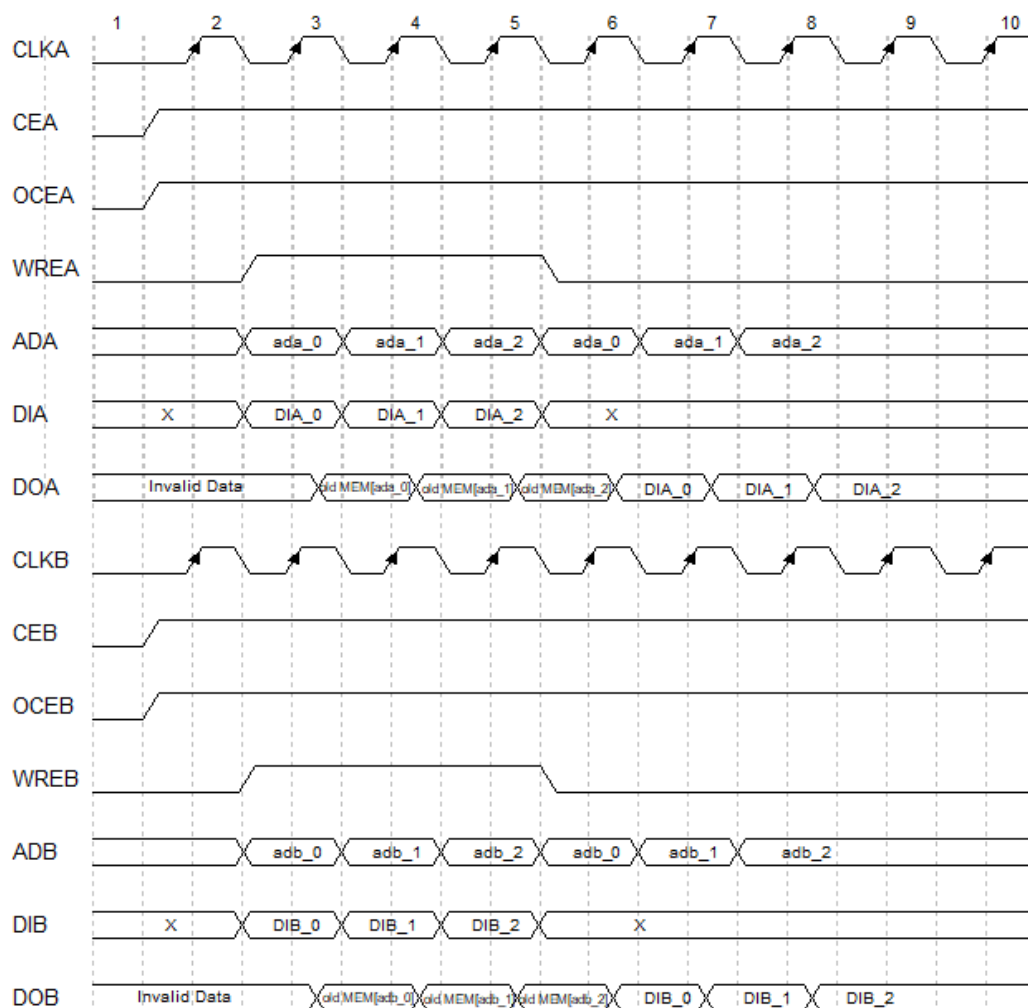
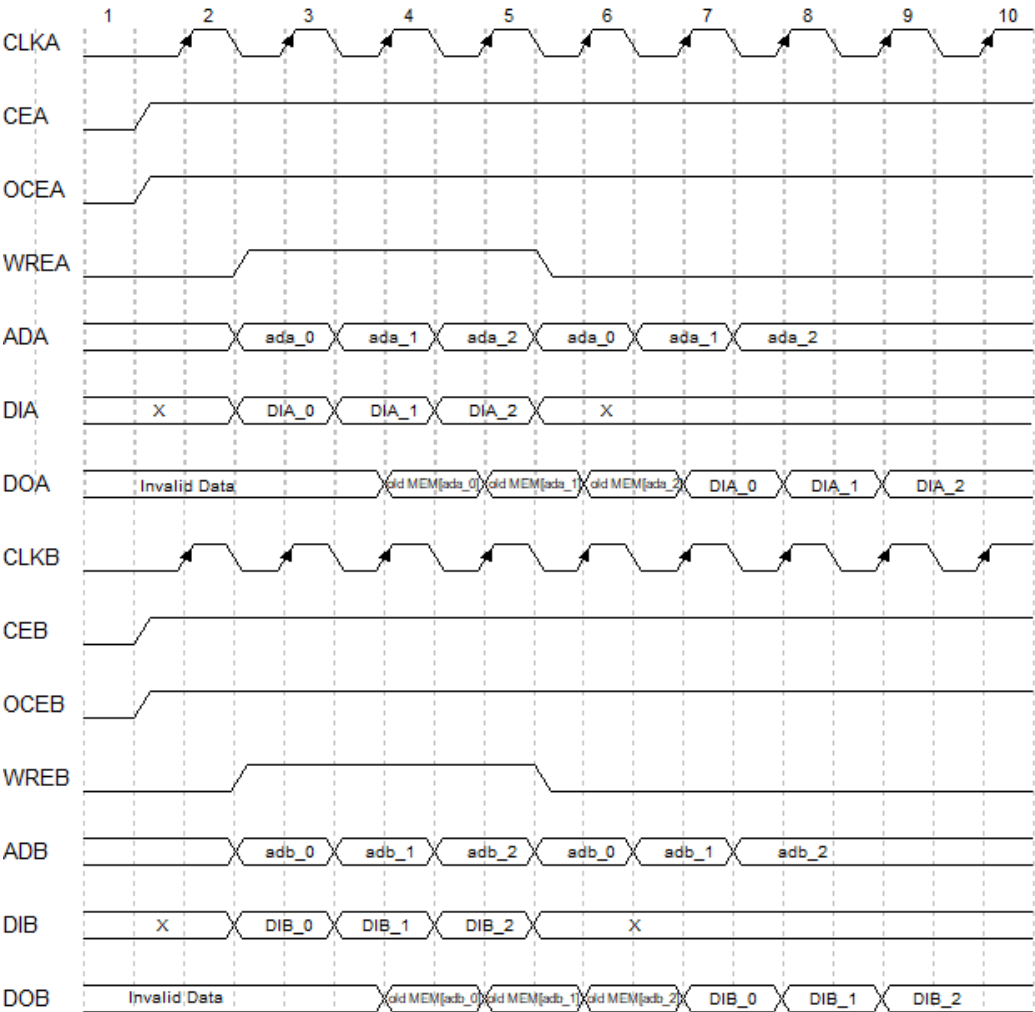


図 3-6 DPB/DPX9B Read-before-Write 書き込みモードのタイミング図(Pipeline 読み出しモード)



- リセットモード  
同期リセット、非同期リセット、およびグローバルリセットをサポートします。

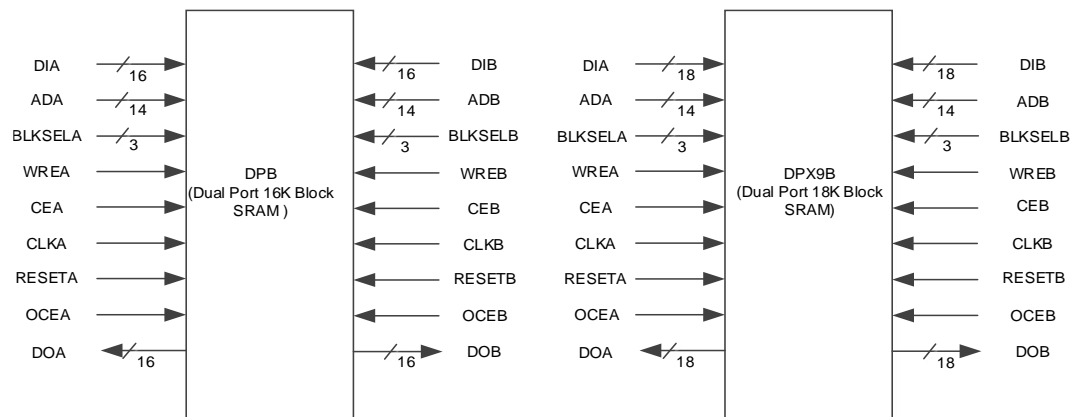
対応関係

表 3-1 DPB/DPX9B のデータ幅とアドレス幅の対応関係

デュアルポート・モード	BSRAM の容量	データ幅	アドレス幅
DPB	16Kbits	1	14
		2	13
		4	12
		8	11
		16	10
DPX9B	18Kbits	9	11
		18	10

## ポート図

図 3-7 DPB/DPX9B のポート図



## ポートの説明

表 3-2 DPB/DPX9B のポートの説明

ポート名	I/O	説明
DOA[15:0]/DOA[17:0]	出力	A ポートのデータ出力信号
DOB[15:0]/DOB[17:0]	出力	B ポートのデータ出力信号
DIA[15:0]/DIA[17:0]	入力	A ポートのデータ入力信号
DIB[15:0]/DIB[17:0]	入力	B ポートのデータ入力信号
ADA[13:0]	入力	A ポートのアドレス入力信号
ADB[13:0]	入力	B ポートのアドレス入力信号
WREA	入力	A ポートの書き込みイネーブル入力信号 ● 1: 書き込み ● 0: 読み出し
WREB	入力	B ポートの書き込みイネーブル入力信号 ● 1: 書き込み ● 0: 読み出し
CEA	入力	A ポートのクロックイネーブル信号、アクティブ High
CEB	入力	B ポートのクロックイネーブル信号、アクティブ High
CLKA	入力	A ポートのクロック入力信号
CLKB	入力	B ポートのクロック入力信号
RESETA	入力	A ポートのリセット入力信号。同期リセットおよび非同期リセットをサポート。アクティブ High RESETA は、メモリ内の値をリセットするのではなく、レジスタをリセットします

ポート名	I/O	説明
RESETB	入力	B ポートのリセット入力信号。同期リセットおよび非同期リセットをサポート。アクティブ HighRESETB は、メモリ内の値をリセットするのではなく、レジスタをリセットします
OCEA	入力	A ポートの出力クロックイネーブル信号。A ポートの Pipeline モードに使用され、Bypass モードでは無効です
OCEB	入力	B ポートの出力クロックイネーブル信号。B ポートの Pipeline モードに使用され、Bypass モードでは無効です
BLKSELA[2:0]	入力	A ポートのブロック選択信号。容量拡張のために複数の BSRAM をカスケード接続する際に使用されます
BLKSELB[2:0]	入力	B ポートのブロック選択信号。容量拡張のために複数の BSRAM をカスケード接続する際に使用されます

### パラメータの説明

表 3-3 DPB/DPX9B のパラメータの説明

パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
READ_MODE0	Integer	1'b0, 1'b1	1'b0	A ポートの読み出しモードの構成 <ul style="list-style-type: none"> <li>● 1'b0 : Bypass モード</li> <li>● 1'b1 : Pipeline モード</li> </ul>
READ_MODE1	Integer	1'b0, 1'b1	1'b0	B ポートの読み出しモードの構成 <ul style="list-style-type: none"> <li>● 1'b0 : Bypass モード</li> <li>● 1'b1 : Pipeline モード</li> </ul>
WRITE_MODE0	Integer	2'b00, 2'b01, 2'b10	2'b00	A ポートの書き込みモードの構成 <ul style="list-style-type: none"> <li>● 2'b00 : Normal モード</li> <li>● 2'b01 : Write-through モード</li> <li>● 2'b10 : Read-before-Write モード</li> </ul>
WRITE_MODE1	Integer	2'b00, 2'b01, 2'b10	2'b00	B ポートの書き込みモードの構成 <ul style="list-style-type: none"> <li>● 2'b00 : Normal モード</li> <li>● 2'b01 : Write-through モード</li> <li>● 2'b10 : Read-before-Write モード</li> </ul>
BIT_WIDTH_0	Integer	DPB: 1,2,4,8,16 DPX9B: 9,18	DPB:16 DPX9B:18	A ポートのデータ幅の構成



パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
BIT_WIDTH_1	Integer	DPB: 1, 2, 4, 8, 16 DPX9B: 9, 18	DPB:16 DPB:18	B ポートのデータ幅の構成
BLK_SEL_0	Integer	3'b000~3'b111	3'b000	A ポートのブロック選択パラメータの設定。ポート BLKSELA の値と同じ場合にこの BSRAM が選択されます。IP Core Generator を使用してメモリ拡張を行う場合、拡張はソフトウェアにより自動的に実行されます。
BLK_SEL_1	Integer	3'b000~3'b111	3'b000	B ポートのブロック選択パラメータの設定。ポート BLKSELB の値と同じ場合にこの BSRAM が選択されます。IP Core Generator を使用してメモリ拡張を行う場合、拡張はソフトウェアにより自動的に実行されます。
RESET_MODE	String	"SYNC", "ASYN"	"SYNC"	リセットモードの構成 ● SYNC : 同期リセット ● ASYN : 非同期リセット
INIT_RAM_00~ INIT_RAM_3F	Integer	DPB: 256'h0... 0~256'h1...1 DPX9B: 288'h0 ...0~288'h1...1	DPB:256' h0...0 DPX9B:2 88'h0...0	BSRAM の初期化データを設定するために使用されます

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

DPB のインスタンス化を例に説明します：

**Verilog** でのインスタンス化：

```
DPB bram_dpb_0 (
    .DOA({doa[15:8],doa[7:0]}),
    .DOB({doa[15:8],dob[7:0]}),
    .CLKA(clka),
    .OCEA(ocea),
    .CEA(cea),
    .RESETA(reseta),
    .WREA(wrea),
```

```

        .CLKB(clkb),
        .OCEB(oceb),
        .CEB(ceb),
        .RESETB(resetb),
        .WREB(wreb),
        .BLKSELA({3'b000}),
        .BLKSELB({3'b000}),
        .ADA({ada[10:0],3'b000}),
        .DIA({{8{1'b0}},dia[7:0]})
        .ADB({adb[10:0],3'b000}),
        .DIB({{8{1'b0}},dib[7:0]})
    );

    defparam bram_dpb_0.READ_MODE0 = 1'b0;
    defparam bram_dpb_0.READ_MODE1 = 1'b0;
    defparam bram_dpb_0.WRITE_MODE0 = 2'b00;
    defparam bram_dpb_0.WRITE_MODE1 = 2'b00;
    defparam bram_dpb_0.BIT_WIDTH_0 = 8;
    defparam bram_dpb_0.BIT_WIDTH_1 = 8;
    defparam bram_dpb_0.BLK_SEL_0 = 3'b000;
    defparam bram_dpb_0.BLK_SEL_1 = 3'b000;
    defparam bram_dpb_0.RESET_MODE = "SYNC";

    defparam bram_dpb_0.INIT_RAM_00 =
256'h00A0000000000000B00A000000000000B00A000000000000B00A00
0000000000B;

    defparam bram_dpb_0.INIT_RAM_3E =
256'h00A0000000000000B00A000000000000B00A000000000000B00A00
0000000000B;

    defparam bram_dpb_0.INIT_RAM_3F =
256'h00A0000000000000B00A000000000000B00A000000000000B00A00
0000000000B;

```

**VHDL** でのインスタンス化 :

COMPONENT DPB

GENERIC (

    BIT\_WIDTH\_0:integer:=16;

    BIT\_WIDTH\_1:integer:=16;

    READ\_MODE0:bit:='0';

    READ\_MODE1:bit:='0';

```

        WRITE_MODE0:bit_vector:="00";
        WRITE_MODE1:bit_vector:="00";
        BLK_SEL_0:bit_vector:="000";
        BLK_SEL_1:bit_vector:="000";
        RESET_MODE:string:="SYNC";
        INIT_RAM_00:bit_vector:=X"0000000000000000
000000000000000000000000000000000000000000000";
        INIT_RAM_01:bit_vector:=X"0000000000000000
000000000000000000000000000000000000000000000";
        INIT_RAM_3F:bit_vector:=X"0000000000000000
000000000000000000000000000000000000000000000"
    );
    PORT (
        DOA,DOB:OUT std_logic_vector(15 downto 0):
=conv_std_logic_vector(0,16);
        CLKA,CLKB,CEA,CEB,OCEA,OCEB,RESETA,
RESETB,WREA,WREB:IN std_logic;
        ADA,ADB:IN std_logic_vector(13 downto 0);
        BLKSELA:IN std_logic_vector(2 downto 0);
        BLKSELB:IN std_logic_vector(2 downto 0);
        DIA,DIB:IN std_logic_vector(15 downto 0)
    );
END COMPONENT;
 uut:DPB
    GENERIC MAP(
        BIT_WIDTH_0=>16,
        BIT_WIDTH_1=>16,
        READ_MODE0=>'0',
        READ_MODE1=>'0',
        WRITE_MODE0=>"00",
        WRITE_MODE1=>"00",
        BLK_SEL_0=>"000",
        BLK_SEL_1=>"000",
        RESET_MODE=>"SYNC",
        INIT_RAM_00=>X"00000000000000000000000000000000
0000000000000000000000000000000000000000000",
        INIT_RAM_01=>X"00000000000000000000000000000000
0000000000000000000000000000000000000000000"
    )

```

```
000000000000000000000000",  
    INIT_RAM_3F=>X"0000000000000000000000000000000000000000  
0000000000000000000000"  
)  
PORT MAP(  
    DOA=>doa,  
    DOB=>dob,  
    CLKA=>clka,  
    CLKB=>clkb,  
    CEA=>ceb,  
    CEB=>ceb,  
    OCEA=>ocea,  
    OC EB=>oceb,  
    RESETA=>reseta,  
    RESETB=>resetb,  
    WREA=>>wrea,  
    WREB=>>wreb,  
    ADA=>ada,  
    ADB=>adb,  
    BLKSELA=>blksela,  
    BLKSELB=>blk selb,  
    DIA=>dia,  
    DIB=>dib  
);
```

### 3.2 シングルポート・モード

## プリミティブの紹介

SP/SPX9(Single Port 16K BSRAM/Single Port 18K BSRAM)は 16K/18K  
のシングルポート BSRAM です。

## 機能の説明

SP/SPX9 はメモリ領域が 16K ビット/18K ビットであるシングルポート・モードの BSRAM です。シングルポートの読み出し/書き込みは 1 つのクロックにより制御されます。2 つの読み出しモード(Bypass モードと Pipeline モード)と 3 つの書き込みモード(Normal モード、Write-through モード、Read-before-Write モード<sup>[1]</sup>)がサポートされます。

注記：

[1]138K デバイスは Read-before-Write モードをサポートしません。

- 読み出しモード  
パラメータの **READ\_MODE** は、出力 **pipeline** レジスタを有効または無効にするために使用されます。出力 **pipeline** レジスタを使用する場合、読み出しには追加のクロックサイクルが必要です。
- 書き込みモード  
Normal モード、Write-through モード、および Read-before-Write モードを含む書き込みモードは、パラメータの **WRITE\_MODE** により構成されます。  
シングルポート **BSRAM** の各読み出し書き込みモードの内部タイミング波形については、デュアルポート **BSRAM** の場合のタイミング図である図 3-1～図 3-6 を参照してください。
- リセットモード  
同期リセット、非同期リセット、およびグローバルリセットをサポートします。

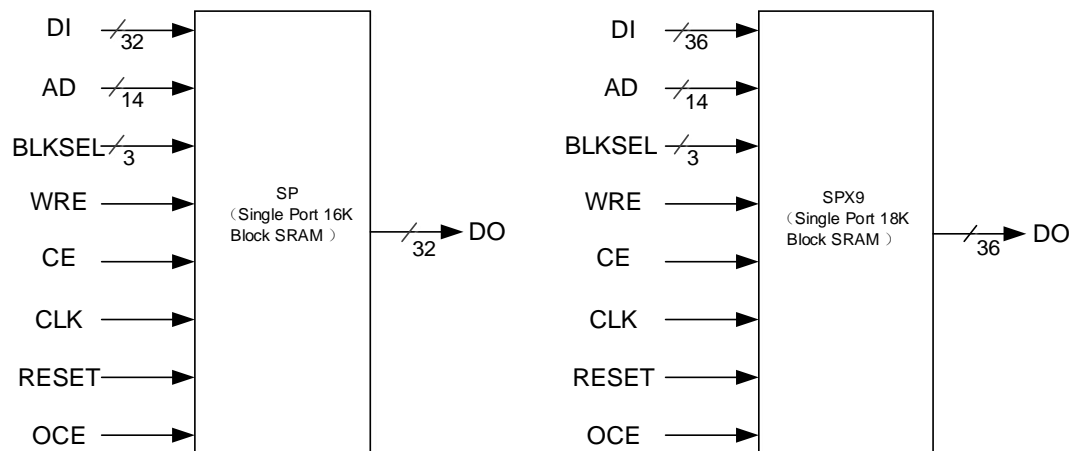
## 対応関係

表 3-4 SP/SPX9 のデータ幅とアドレス幅の対応関係

シングルポート・モード	BSRAM の容量	データ幅	アドレス幅
SP	16Kbits	1	14
		2	13
		4	12
		8	11
		16	10
		32	9
SPX9	18Kbits	9	11
		18	10
		36	9

## ポート図

図 3-8 SP/SPX9 のポート図



## ポートの説明

表 3-5 SP/SPX9 のポートの説明

ポート名	I/O	説明
DO[31:0]/DO[35:0]	出力	データ出力信号
DI[31:0]/DI[35:0]	入力	データ入力信号
AD[13:0]	入力	アドレス入力信号
WRE	入力	書き込みイネーブル入力信号 ● 1 : 書き込み ● 0 : 読み出し
CE	入力	クロックイネーブル入力信号、アクティブ High
CLK	入力	クロック入力信号
RESET	入力	リセット入力信号。同期リセットおよび非同期リセットをサポート。アクティブ High RESET は、メモリ内の値をリセットするのではなく、レジスタをリセットします
OCE	入力	出力クロックイネーブル信号。Pipeline モードに使用され、Bypass モードでは無効です
BLKSEL[2:0]	入力	BSRAM ブロック選択信号。容量拡張のために複数の BSRAM をカスケード接続する際に使用されます

## パラメータの説明

表 3-6 SP/SPX9 のパラメータの説明

パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
READ_MODE	Integer	1'b0, 1'b1	1'b0	読み出しモードの構成

パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
				<ul style="list-style-type: none"> <li>● 1'b0 : Bypass モード</li> <li>● 1'b1 : Pipeline モード</li> </ul>
WRITE_MODE	Integer	2'b00, 2'b01, 2'b10	2'b00	書き込みモードの構成 <ul style="list-style-type: none"> <li>● 2'b00 : Normal モード</li> <li>● 2'b01 : Write-through モード</li> <li>● 2'b10 : Read-before-Write モード</li> </ul>
BIT_WIDTH	Integer	SP: 1, 2, 4, 8, 16, 32 SPX9: 9, 18, 36	SP:32 SPX9:36	データ幅の構成
BLK_SEL	Integer	3'b000~3'b111	3'b000	BSRAM ブロック選択パラメータの設定。ポート BLKSEL の値と同じ場合にこの BSRAM が選択されます。IP Core Generator を使用してメモリ拡張を行う場合、拡張はソフトウェアにより自動的に実行されます。
RESET_MODE	String	"SYNC", "ASYNC"	"SYNC"	リセットモードの構成 <ul style="list-style-type: none"> <li>● SYNC : 同期リセット</li> <li>● ASYNC : 非同期リセット</li> </ul>
INIT_RAM_00~ INIT_RAM_3F	Integer	SP:256'h0...0~256'h1...1 SPX9: 288'h0...0~288'h1...1	SP:256'h0...0 SPX9:288'h0...0	BSRAM の初期化データを設定するために使用されます

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

SP のインスタンス化を例に説明します：

**Verilog** でのインスタンス化：

```
SP bram_sp_0 (
    .DO({dout[31:8], dout[7:0]}),
    .CLK(clk),
    .OCE(oce),
    .CE(ce),
    .RESET(reset),
```

```

        .WRE(wre),
        .BLKSEL({3'b000}),
        .AD({ad[10:0], 3'b000}),
        .DI({{24{1'b0}}, din[7:0]})
    );
    defparam bram_sp_0.READ_MODE = 1'b0;
    defparam bram_sp_0.WRITE_MODE = 2'b00;
    defparam bram_sp_0.BIT_WIDTH = 8;
    defparam bram_sp_0.BLK_SEL = 3'b000;
    defparam bram_sp_0.RESET_MODE = "SYNC";
    defparam bram_sp_0.INIT_RAM_00 =
        256'h00A00000000000B00A000000000000B00A000000000000B00
        A0000000000000B;
    defparam bram_sp_0.INIT_RAM_01 =
        256'h00A00000000000B00A000000000000B00A000000000000B00
        A0000000000000B;
    defparam bram_sp_0.INIT_RAM_3F =
        256'h00A00000000000B00A000000000000B00A000000000000B00
        A0000000000000B;

```

**VHDL** でのインスタンス化 :

```

    COMPONENT SP
    GENERIC(
        BIT_WIDTH:integer:=32;
        READ_MODE:bit:='0';
        WRITE_MODE:bit_vector:="01";
        BLK_SEL:bit_vector:="000";
        RESET_MODE:string:"SYNC";
        INIT_RAM_00:bit_vector:=X"00A000000000000B
00A000000000000B00A000000000000B00A000000000000B ";
        INIT_RAM_01:bit_vector:=X"00A000000000000B
00A000000000000B00A000000000000B00A000000000000B ";
        INIT_RAM_3F:bit_vector:=X"00A000000000000B
00A000000000000B00A000000000000B00A000000000000B ";
    );
    PORT(
        DO:OUT std_logic_vector(31 downto 0):=conv_
std_logic_vector(0,32);
        CLK,CE,OCE,RESET,WRE:IN std_logic;

```



```

        AD:IN std_logic_vector(13 downto 0);
        BLKSEL:IN std_logic_vector(2 downto 0);
        DI:IN std_logic_vector(31 downto 0)

    );
END COMPONENT;
uut:SP
    GENERIC MAP(
        BIT_WIDTH=>32,
        READ_MODE=>'0',
        WRITE_MODE=>"01",
        BLK_SEL=>"000",
        RESET_MODE=>"SYNC",
        INIT_RAM_00=>X"00A0000000000000B00A00
0000000000B00A00000000000000B00A000000000000B ",
        INIT_RAM_01=>X"00A0000000000000B00A00
0000000000B00A00000000000000B00A000000000000B ",
        INIT_RAM_02=>X"00A0000000000000B00A00
0000000000B00A00000000000000B00A000000000000B ",
        INIT_RAM_3F=>X"00A0000000000000B00A00
0000000000B00A00000000000000B00A000000000000B "
    )
    PORT MAP (
        DO=>dout,
        CLK=>clk,
        OCE=>oce,
        CE=>ce,
        RESET=>reset,
        WRE=>wre,
        BLKSEL=>blkssel,
        AD=>ad,
        DI=>din
    );

```

### 3.3 セミ・デュアルポート・モード

#### プリミティブの紹介

SDPB/SDPX9B(Semi Dual Port 16K Block SRAM /Semi Dual Port 18K Block SRAM )は、16K/18K のセミ・デュアルポート BSRAM です。

## 機能の説明

SDPB/SDPX9B はそれぞれメモリ領域が 16K ビット/18K ビットである セミ・デュアルポート・モードの BSRAM です。セミ・デュアルポート・モードでは、A ポートでは書き込み、B ポートでは読み出しが行われます<sup>[1]</sup>。2 つの読み出しモード(Bypass モードと Pipeline モード)と 1 つの書き込みモード(Normal モード)がサポートされます。

注記：

[1]同じアドレスに対して、同時に読み出しと書き込みを行うことは推奨されません。

- 読み出しモード

パラメータの READ\_MODE は、出力 pipeline レジスタを有効または無効にするために使用されます。出力 pipeline レジスタを使用する場合、読み出しには追加のクロックサイクルが必要です。

- 書き込みモード

SDPB/SDPX9B の A ポートは書き込み、B ポートは読み出しに使用されます。Normal モードをサポートします。セミ・デュアルポート BSRAM の各モードの内部タイミング波形を図 3-9 および図 3-10 に示します。

図 3-9 セミ・デュアルポート BSRAM の normal 書き込みモードのタイミング図 (Bypass 読み出しモード)

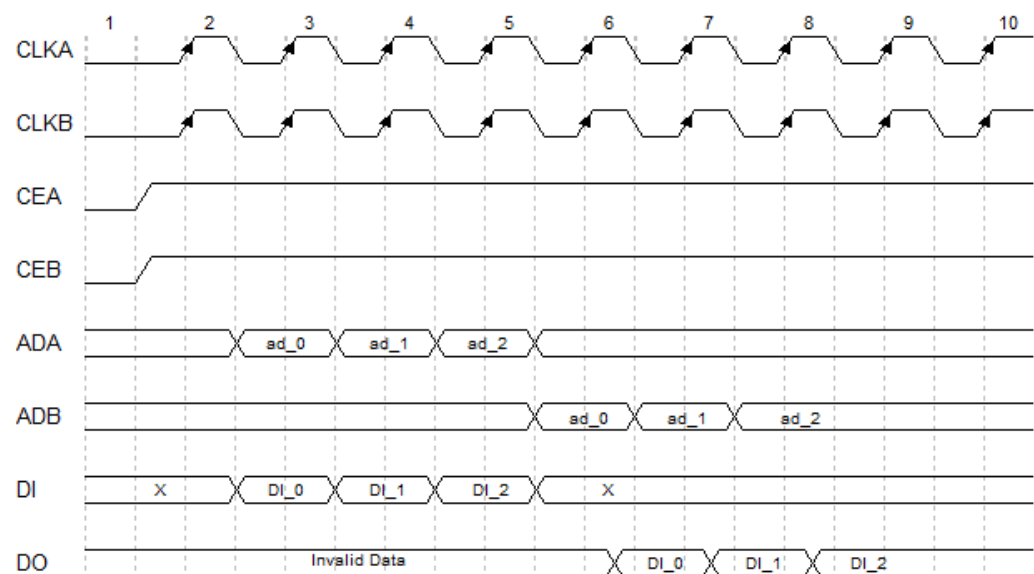
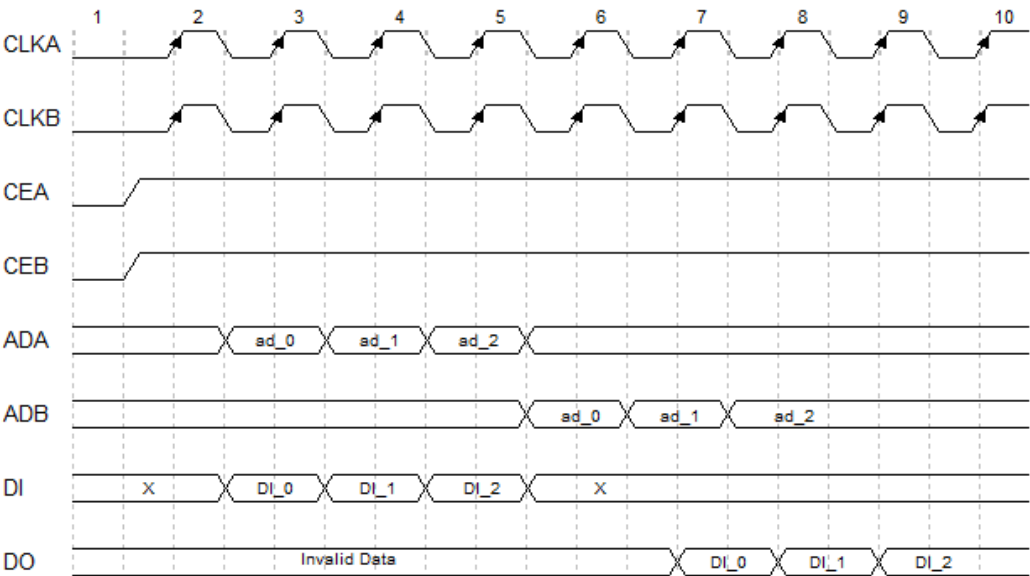


図 3-10 セミ・デュアルポート BSRAM の normal 書き込みモードのタイミング図 (Pipeline 読み出しモード)



- リセットモード  
同期リセット、非同期リセット、およびグローバルリセットをサポートします。
- byte\_enable 機能  
byte\_enable 機能は、8 ビットのバイトイネーブル信号により制御されます。
- カスケード接続機能  
カスケード接続をサポートします。

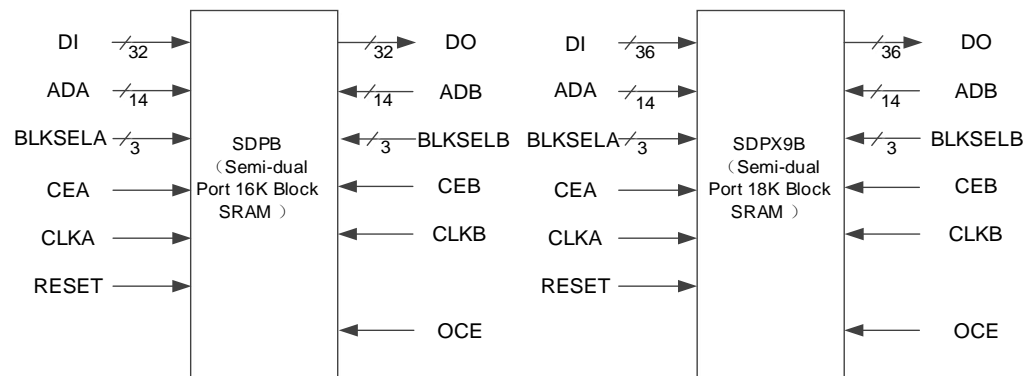
対応関係

表 3-7 SDPB/SDPX9B のデータ幅とアドレス幅の対応関係

セミ・デュアルポート・モード	BSRAM の容量	データ幅	アドレス幅
SDPB	16Kbits	1	14
		2	13
		4	12
		8	11
		16	10
		32	9
SDPX9B	18Kbits	9	11
		18	10
		36	9

## ポート図

図 3-11 SDPB/SDPX9B のポート図



## ポートの説明

表 3-8 SDPB/SDPX9B のポートの説明

ポート名	I/O	説明
DO[31:0]/DO[35:0]	出力	データ出力信号
DI[31:0]/DI[35:0]	入力	データ入力信号
ADA[13:0]	入力	A ポートのアドレス入力信号
ADB[13:0]	入力	B ポートのアドレス入力信号
CEA	入力	A ポートのクロックイネーブル信号、アクティブ High
CEB	入力	B ポートのクロックイネーブル信号、アクティブ High
CLKA	入力	A ポートのクロック入力信号
CLKB	入力	B ポートのクロック入力信号
RESET	入力	リセット入力信号。同期リセットおよび非同期リセットをサポート。アクティブ High。RESET は、メモリ内の値をリセットするのではなく、レジスタをリセットします
OCE	入力	出力クロックイネーブル信号。Pipeline モードに使用され、Bypass モードでは無効です
BLKSELA[2:0]	入力	A ポートのブロック選択信号。容量拡張のために複数の BSRAM をカスケード接続する際に使用されます
BLKSELB[2:0]	入力	B ポートのブロック選択信号。容量拡張のために複数の BSRAM をカスケード接続する際に使用されます

## パラメータの説明

表 3-9 SDPB/SDPX9B のパラメータの説明

パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
READ_MODE	Integer	1'b0,1'b1	1'b0	読み出しモードの構成 <ul style="list-style-type: none"> <li>● 1'b0 : Bypass モード</li> <li>● 1'b1 : Pipeline モード</li> </ul>
BIT_WIDTH_0	Integer	SDPB:1,2,4,8,16,32 SDPX9B:9,18,36	SDPB:32 SDPX9B:36	A ポートのデータ幅の構成
BIT_WIDTH_1	Integer	SDPB:1,2,4,8,16,32 SDPX9B:9,18,36	SDPB:32 SDPX9B:36	B ポートのデータ幅の構成
BLK_SEL_0	Integer	3'b000~3'b111	3'b000	A ポートのブロック選択パラメータの設定。ポート BLKSELA の値と同じ場合にこの BSRAM が選択されます。IP Core Generator を使用してメモリ拡張を行う場合、拡張はソフトウェアにより自動的に実行されます。
BLK_SEL_1	Integer	3'b000~3'b111	3'b000	B ポートのブロック選択パラメータの設定。ポート BLKSELB の値と同じ場合にこの BSRAM が選択されます。IP Core Generator を使用してメモリ拡張を行う場合、拡張はソフトウェアにより自動的に実行されます。
RESET_MODE	String	"SYNC","ASYN"	"SYNC"	リセットモードの構成 <ul style="list-style-type: none"> <li>● SYNC : 同期リセット</li> <li>● ASYN : 非同期リセット</li> </ul>
INIT_RAM_00~ INIT_RAM_3F	Integer	SDPB:256'h0... 0~256'h1...1 SDPX9B:288'h0... 0~288'h1...1	SDPB:256'h0... 0 SDPX9B:288' h0...0	BSRAM の初期化データを設定するために使用されます

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6IP の呼び出し](#)を参照してください。

SDPB のインスタンス化を例に説明します：

### Verilog でのインスタンス化：

```
SDPB bram_sdpb_0 (
    .DO({dout[31:16],dout[15:0]}),
    .CLKA(clka),
    .CEA(cea),
    .CLKB(clkb),
    .CEB(ceb),
    .RESET(reset),
    .OCE(oce),
    .BLKSELA({3'b000}),
    .BLKSELB({3'b000}),
    .ADA({ada[9:0], 2'b00, byte_en[1:0]}),
    .DI({{16{1'b0}},din[15:0]}),
    .ADB({adb[9:0],4'b0000})
);

defparam bram_sdpb_0.READ_MODE = 1'b1;
defparam bram_sdpb_0.BIT_WIDTH_0 = 16;
defparam bram_sdpb_0.BIT_WIDTH_1 = 16;
defparam bram_sdpb_0.BLK_SEL_0 = 3'b000;
defparam bram_sdpb_0.BLK_SEL_1 = 3'b000;
defparam bram_sdpb_0.RESET_MODE = "SYNC";
defparam bram_sdpb_0.INIT_RAM_00 =
256'h00A0000000000000B00A000000000000B00A000000000000B00
A0000000000000B;
defparam bram_sdpb_0.INIT_RAM_3F =
256'h00A0000000000000B00A000000000000B00A000000000000B00
A0000000000000B;
```

### VHDL でのインスタンス化：

```
COMPONENT SDPB
    GENERIC(
        BIT_WIDTH_0:integer:=16;
        BIT_WIDTH_1:integer:=16;
```

```

        READ_MODE:bit:='0';
        BLK_SEL_0:bit_vector:="000";
        BLK_SEL_1:bit_vector:="000";
        RESET_MODE:string:="SYNC";
        INIT_RAM_00:bit_vector:=X"00A0000000000000
B00A000000000000B00A000000000000B00A000000000000B";
        INIT_RAM_01:bit_vector:=X"00A0000000000000
B00A000000000000B00A000000000000B00A000000000000B";
        INIT_RAM_3F:bit_vector:=X"00A0000000000000
B00A000000000000B00A000000000000B00A000000000000B"
    );
    PORT(
        DO:OUT std_logic_vector(31 downto 0):=conv_std
        _logic_vector(0,32);
        CLKA,CLKB,CEA,CEB:IN std_logic;
        OCE,RESET:IN std_logic;
        ADA,ADB:IN std_logic_vector(13 downto 0);
        BLKSELA:IN std_logic_vector(2 downto 0);
        BLKSELB:IN std_logic_vector(2 downto 0);
        DI:IN std_logic_vector(31 downto 0)
    );
END COMPONENT;
uut:SDPB
    GENERIC MAP(
        BIT_WIDTH_0=>16,
        BIT_WIDTH_1=>16,
        READ_MODE=>'0',
        BLK_SEL_0=>"000",
        BLK_SEL_1=>"000",
        RESET_MODE=>"SYNC",
        INIT_RAM_00=>X"00A0000000000000B00A00
0000000000B00A000000000000B00A000000000000B",
        INIT_RAM_01=>X"00A0000000000000B00A00
0000000000B00A000000000000B00A000000000000B",
        INIT_RAM_3F=>X"00A0000000000000B00A00
0000000000B00A000000000000B00A000000000000B"
    )

```

```

PORT MAP(
    DO=>dout,
    CLKA=>clka,
    CEA=>cea,
    CLKB=>clkb,
    CEB=>ceb,
    RESET=>reset,
    OCE=>oce,
    BLKSELA=>blksela,
    BLKSELB=>blkseleb,
    ADA=>ada,
    DI=>din,
    ADB=>adb
);

```

### 3.4 ECC 機能付きのセミ・デュアルポート・モード

#### プリミティブの紹介

SDP36KE(Semi Dual Port 36K Block SRAM with ECC function)は、ECC 機能付きの 36K のセミ・デュアルポート BSRAM です。

#### 機能の説明

SDP36KE はメモリ領域が 36K ビットであるセミ・デュアルポート・モードの BSRAM です。セミ・デュアルポート・モードでは、A ポートでは書き込み、B ポートでは読み出しが行われます<sup>[1]</sup>。2 つの読み出しモード(Bypass モードと Pipeline モード)と 1 つの書き込みモード(Normal モード)がサポートされます。

#### 注記：

[1]同じアドレスに対して、同時に読み出しと書き込みを行うことは推奨されません。

- 読み出しモード  
パラメータの READ\_MODE は、出力 pipeline レジスタを有効または無効にするために使用されます。出力 pipeline レジスタを使用する場合、読み出しには追加のクロックサイクルが必要です。
- 書き込みモード  
SDP36KE の A ポートは書き込み、B ポートは読み出しに使用されます。Normal モードをサポートします。
- リセットモード  
同期リセット、非同期リセット、およびグローバルリセットをサポートします。



- パリティチェック

データポートは、 $64+8=72$  ビット幅の組み合わせです(64 ビットの入出力データ (DI/DO) + 8 ビットの入出力データ (DIP/DOP))。8 ビットの DIP/DOP はパリティチェック機能付きの入力/出力です。非 ECC モードでは、DIP/DOP はデータ転送に使用できますが、ECC モードでは、DIP/DOP はデータ転送に使用できません。

- ECC チェック

ECC チェックをサポートします。データ幅は 72 ビットです。  
standard、encoder-only、および decoder-only の 3 つの ECC モードがサポートされます。

- Standard ECC : エンコーダとデコーダを同時に有効にし、エンコーダとデコーダを使用して ECC 機能を実現できます。
- Encoder-only ECC : エンコーダのみを有効にし、読み出される値はデコードせずにすぐに出力されます。
- Decoder-only ECC : デコーダのみを有効にします。

- byte\_enable 機能

非 ECC モードでは、byte\_enable 機能は 8 ビットのバイトイネーブル信号により制御され、ECC モードでは、デフォルトですべてのバイトがイネーブルされます。

- カスケード接続機能

カスケード接続をサポートします。

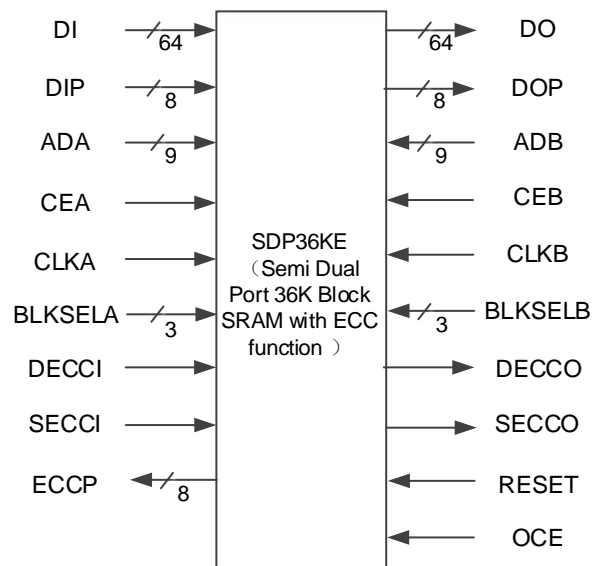
## 対応関係

表 3-10 SDP36KE のデータ幅とアドレス幅の対応関係

セミ・デュアルポート・モード	BSRAM の容量	データ幅	アドレス幅
SDP36KE	36Kbits	36	10
		72	9

## ポート図

図 3-12 SDP36KE のポート図



## ポートの説明

表 3-11 SDP36KE のポートの説明

ポート名	I/O	説明
DO[63:0]	出力	データ出力信号
DI[63:0]	入力	データ入力信号
DIP[7:0]	入力	DIP はパリティチェック入力として使用することができます。非 ECC モードでは、DIP はデータ入力に使用できますが、ECC モードでは、DIP はデータ入力に使用できません。
DOP[7:0]	出力	DOP はパリティチェック出力として使用することができます。非 ECC モードでは、DOP はデータ出力に使用できますが、ECC モードでは、DIP はデータ出力に使用できません。
ECCP[7:0]	出力	Ecc encoder パリティビット
ADA[9:0]	入力	A ポートのアドレス入力信号
ADB[9:0]	入力	B ポートのアドレス入力信号
CEA	入力	A ポートのクロックイネーブル信号、アクティブ High
CEB	入力	B ポートのクロックイネーブル信号、アクティブ High
CLKA	入力	A ポートのクロック入力信号
CLKB	入力	B ポートのクロック入力信号
RESET	入力	出力リセット信号
OCE	入力	出力クロックイネーブル信号。Pipeline モードに使用され、Bypass モードでは無効です

ポート名	I/O	説明
BLKSELA[2:0]	入力	A ポートのブロック選択信号。容量拡張のために複数の BSRAM をカスケード接続する際に使用されます
BLKSELB[2:0]	入力	B ポートのブロック選択信号。容量拡張のために複数の BSRAM をカスケード接続する際に使用されます
BWE[7:0]	入力	バイトイネーブル信号
DECCI	入力	ダブルビット・エラーを挿入します
SECCI	入力	シングルビット・エラーを挿入します
DECCO	出力	ダブルビット・エラーの検出
SECCO	出力	シングルビット・エラーの検出

### パラメータの説明

表 3-12 SDP36KE のパラメータの説明

パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
READ_MODE	Integer	1'b0,1'b1	1'b0	読み出しモードの構成 <ul style="list-style-type: none"> <li>● 1'b0 : Bypass モード</li> <li>● 1'b1 : Pipeline モード</li> </ul>
BIT_WIDTH_A	Integer	36,72	36	A ポートのデータ幅の構成
BIT_WIDTH_B	Integer	36,72	36	B ポートのデータ幅の構成
BLK_SEL_A	Integer	3'b000~3'b111	3'b000	A ポートのブロック選択パラメータの設定。ポート BLKSELA の値と同じ場合にこの BSRAM が選択されます。IP Core Generator を使用してメモリ拡張を行う場合、拡張はソフトウェアにより自動的に実行されます。

パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
BLK_SEL_B	Integer	3'b000~3'b111	3'b000	B ポートのブロック選択パラメータの設定。ポート BLKSELB の値と同じ場合にこの BSRAM が選択されます。IP Core Generator を使用してメモリ拡張を行う場合、拡張はソフトウェアにより自動的に実行されます。
RESET_MODE	String	"SYNC","ASYN"	"SYNC"	リセットモードの構成 <ul style="list-style-type: none"> <li>● SYNC : 同期リセット</li> <li>● ASYN : 非同期リセット</li> </ul>
ECC_WRITE_EN	String	"TRUE","FALSE"	"FALSE"	ECC Encoder の構成 <ul style="list-style-type: none"> <li>● TRUE : ECC Encoder を有効にします</li> <li>● FALSE : ECC Encoder を無効にします</li> </ul>
ECC_READ_EN	String	"TRUE","FALSE"	"FALSE"	ECC decoder の構成 <ul style="list-style-type: none"> <li>● TRUE : ECC decoder を有効にします</li> <li>● FALSE : ECC decoder を無効にします</li> </ul>
INIT_RAM_00~ INIT_RAM_7F	Integer	256'h0...0~256'h1...1	256'h0...0	32K ビットのメモリの初期値を指定するために使用され、DO から出力されます。
INITP_RAM_00~ INITP_RAM_0F	Integer	256'h0...0~256'h1...1	256'h0...0	4K ビットのメモリの初期値を指定するために使用され、DOP から出力されます。

パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
INIT_FILE	String	“NONE” , “*.ini”	“NONE”	NONE : 初期化ファイルはありません。この場合のメモリ値は、INIT_RAM_00~INIT_RAM_7F と INITP_RAM_00~INITP_RAM_0F により指定されます。

### プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

#### Verilog でのインスタンス化 :

```
SDP36KE bram_sdp36ke_0 (
    .DI({28{1'b0}},din[35:0]),
    .DO({dout[63:36],dout[35:0]}),
    .DIP({8{1'b0}}),
    .DOP({dout[71:64]}),
    .ECCP(eccp),
    .ADA(ada),
    .ADB(adb),
    .CLKA(clka),
    .CLKB(clkb),
    .CEA(cea),
    .CEB(ceb),
    .OCE(oce),
    .RESET(reset),
    .BLKSELA({3'b000}),
    .BLKSELB({3'b000}),
    .BWE(bwe),
    .DECCI(decci),
    .SECCI(secci),
    .DECCO(decco),
```

```

        .SECCO(secco)
    );
    defparam bram_sdp36ke_0.ECC_WRITE_EN = "FALSE";
    defparam bram_sdp36ke_0.ECC_READ_EN = "FALSE";
    defparam bram_sdp36ke_0.READ_MODE = 1'b0;
    defparam bram_sdp36ke_0.BIT_WIDTH_A = 36;
    defparam bram_sdp36ke_0.BIT_WIDTH_B = 36;
    defparam bram_sdp36ke_0.BLK_SEL_A = 3'b000;
    defparam bram_sdp36ke_0.BLK_SEL_B = 3'b000;
    defparam bram_sdp36ke_0.RESET_MODE = "SYNC";
    defparam bram_sdp36ke_0.INIT_FILE = "NONE";

    defparam bram_sdp36ke_0.INIT_RAM_00 =
        256'h00A0000000000000B00A000000000000B00A000000000000B00
        A0000000000000B;
    defparam bram_sdp36ke_0.INIT_RAM_7F =
        256'h00A0000000000000B00A000000000000B00A000000000000B00
        A0000000000000B;
    defparam bram_sdp36ke_0.INITP_RAM_00 =
        256'h00A0000000000000B00A000000000000B00A000000000000B00
        A0000000000000B;
    defparam bram_sdp36ke_0.INITP_RAM_0F =
        256'h00A0000000000000B00A000000000000B00A000000000000B00
        A0000000000000B;

```

**VHDL** でのインスタンス化 :

```

    COMPONENT SDP36KE
    GENERIC(
        ECC_WRITE_EN:string:="FALSE";
        ECC_READ_EN:string:="FALSE";
        READ_MODE:bit:='0';
        BIT_WIDTH_A:integer:=36;
        BIT_WIDTH_B:integer:=36;
        BLK_SEL_A:bit_vector:="000";
        BLK_SEL_B:bit_vector:="000";
        RESET_MODE:string:="SYNC";
        INIT_FILE:string:="NONE";
        INIT_RAM_00:bit_vector:=X"00A0000000000000
B00A000000000000B00A000000000000B00A000000000000B";
        INIT_RAM_7F:bit_vector:=X"00A0000000000000

```

```

B00A000000000000B00A000000000000B00A000000000000B";
        INITP_RAM_00:bit_vector:=X"00A00000000000
0B00A0000000000000B00A000000000000B00A000000000000B";
        INITP_RAM_0F:bit_vector:=X"00A00000000000
0B00A0000000000000B00A000000000000B00A000000000000B"

    );
    PORT(
        DO:OUT std_logic_vector(63 downto 0):=conv_std
        _logic_vector(0,64);
        DOP:OUT std_logic_vector(7 downto 0):=conv_st
        d_logic_vector(0,8);
        ECCP:OUT std_logic_vector(7 downto 0):=conv
        _std_logic_vector(0,8);
        DECCO,SECCO:OUT std_logic:=conv_std_logic;
        DECCI,SECCI:IN std_logic;
        ADA,ADB:IN std_logic_vector(9 downto 0);
        CLKA,CLKB,CEA,CEB:IN std_logic;
        OCE,RESET:IN std_logic;
        BLKSELA:IN std_logic_vector(2 downto 0);
        BLKSELB:IN std_logic_vector(2 downto 0);
        BWE:IN std_logic_vector(7 downto 0);
        DIP:IN std_logic_vector(7 downto 0);
        DI:IN std_logic_vector(63 downto 0)

    );
END COMPONENT;
 uut:SDP36KE
    GENERIC MAP(
        ECC_WRITE_EN=>"FALSE";
        ECC_READ_EN=>"FALSE";
        READ_MODE=>'0';
        BIT_WIDTH_A=>36;
        BIT_WIDTH_B=>36;
        BLK_SEL_A=>"000";
        BLK_SEL_B=>"000";
        RESET_MODE=>"SYNC";
        INIT_FILE=>"NONE";
        INIT_RAM_00=>X"00A0000000000000B00A00

```

```

0000000000B00A00000000000000B00A000000000000B",
                                INIT_RAM_7F=>X"00A0000000000000B00A00
0000000000B00A00000000000000B00A000000000000B",
                                INITP_RAM_00=>X"00A0000000000000B00A0
0000000000B00A00000000000000B00A000000000000B",
                                INITP_RAM_0F=>X"00A0000000000000B00A0
0000000000B00A00000000000000B00A000000000000B"
                                )

```

```

PORT MAP(
    DI=>din,
    DO=>dout,
    DIP=>dip,
    DOP=>dop,
    ECCP=>eccp,
    ADA=>ada,
    ADB=>adb,
    CLKA=>clka,
    CLKB=>clkb,
    CEA=>cea,
    CEB=>ceb,
    OCE=>oce,
    RESET=>reset,
    BLKSELA=>blksela,
    BLKSELB=>blksele,
    BWE=>bwe,
    DECCI=>decci,
    SECCI=>secci,
    DECCO=>decco,
    SECCO=>secco
);

```

## 3.5 ROM モード

### プリミティブの紹介

pROM/pROMX9(16K/18K Block ROM)は 16K/18K のブロック ROM です。



機能の説明

pROM/pROMX9 はメモリ領域がそれぞれ 16K ビット/18K ビットである読み出し専用メモリで、2 つの読み出しモード(Bypass モードと Pipeline モード)をサポートします。

パラメータの READ\_MODE は、出力 pipeline レジスタを有効または無効にするために使用されます。出力 pipeline レジスタを使用する場合、読み出しには追加のクロックサイクルが必要です。

ROM の各読み出しモードの内部タイミング波形については、セミ・デュアルポート BSRAM の B ポートのタイミング図である図 3-13 および図 3-14 を参照してください。

図 3-13 ROM のタイミング図(Bypass モード)

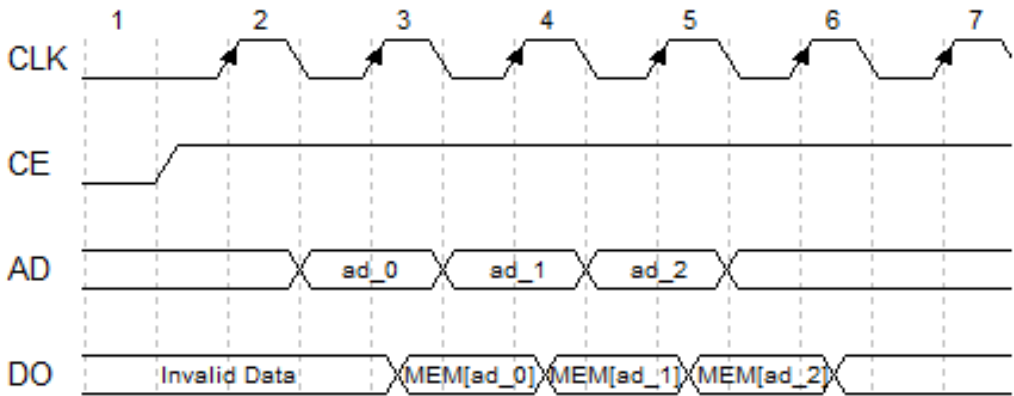
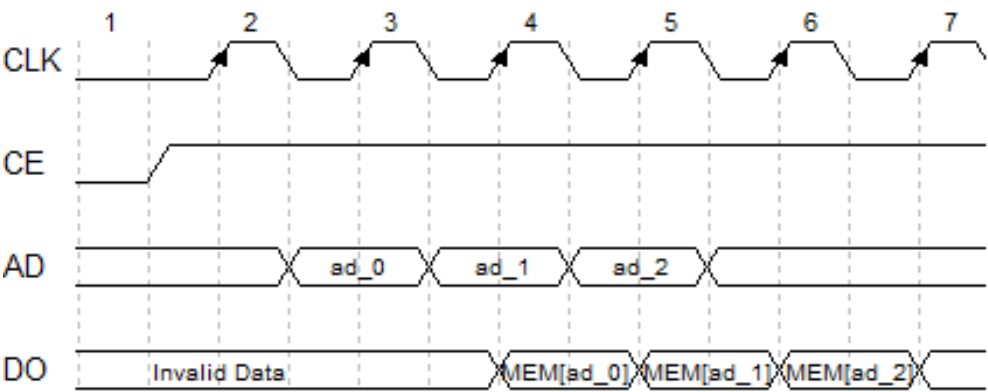


図 3-14 ROM のタイミング図 (Pipeline モード)



● リセットモード

同期リセット、非同期リセット、およびグローバルリセットをサポートします。

対応関係

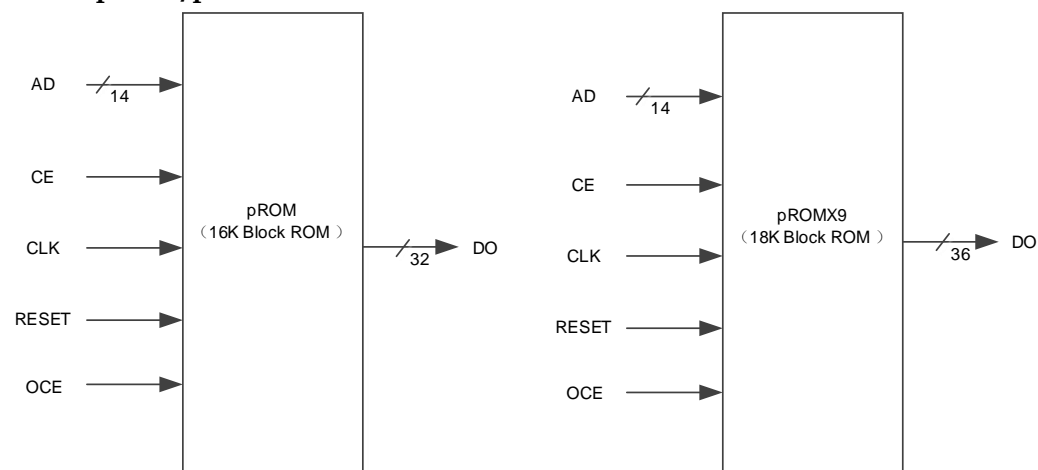
表 3-13 pROM/pROMX9 のデータ幅とアドレス幅の対応関係

ROM モード	BSRAM の容量	データ幅	アドレス幅
pROM	16Kbits	1	14

ROM モード	BSRAM の容量	データ幅	アドレス幅
		2	13
		4	12
		8	11
		16	10
		32	9
pROMX9	18Kbits	9	11
		18	10
		36	9

### ポート図

図 3-15 pROM/pROMX9 のポート図



### ポートの説明

表 3-14 pROM/pROMX9 のポートの説明

ポート名	I/O	説明
DO[31:0]/DO[35:0]	出力	データ出力信号
AD[13:0]	入力	アドレス入力信号
CE	入力	クロックイネーブル入力信号、アクティブ High
CLK	入力	クロック入力信号
RESET	入力	リセット入力信号。同期リセットおよび非同期リセットをサポート。アクティブ High RESET は、メモリ内の値をリセットするのではなく、レジスタをリセットします
OCE	入力	出力クロックイネーブル信号。Pipeline モードに使用され、Bypass モードでは無効です

## パラメータの説明

表 3-15 pROM/pROMX9 のパラメータの説明

パラメータ名	パラメータのタイプ	値の範囲	デフォルト値	説明
READ_MODE	Integer	1'b0,1'b1	1'b0	読み出しモードの構成 <ul style="list-style-type: none"> <li>1'b0 : Bypass モード</li> <li>1'b1 : Pipeline モード</li> </ul>
BIT_WIDTH	Integer	pROM:1,2,4,8,16,32 pROMX9:9,18,36	pROM:32 pROMX9:36	データ幅の構成
RESET_MODE	String	"SYNC","ASYN"	"SYNC"	リセットモードの構成 <ul style="list-style-type: none"> <li>SYNC : 同期リセット</li> <li>ASYN : 非同期リセット</li> </ul>
INIT_RAM_00~ INIT_RAM_3F	Integer	pROM:256'h0... 0~256'h1...1 pROMX9:288'h0... 0~288'h1...1	pROM:256'h 0...0 pROMX9:28 8'h0...0	BSRAM の初期化データを設定するために使用されます

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

pROM のインスタンス化を例に説明します：

**Verilog** でのインスタンス化：

```
pROM bram_prom_0 (
    .DO({dout[31:8],dout[7:0]}),
    .CLK(clk),
    .OCE(oce),
    .CE(ce),
    .RESET(reset),
    .AD({ad[10:0],3'b000})
);
defparam bram_prom_0.READ_MODE = 1'b0;
defparam bram_prom_0.BIT_WIDTH = 8;
defparam bram_prom_0.RESET_MODE = "SYNC";
```

```
defparam bram_prom_0.INIT_RAM_00 =
256'h9C23645D0F78986FFC3E36E141541B95C19F2F7164085E63
1A819860D8FF0000;
```

[illegible]

## VHDL でのインスタンス化：

## COMPONENT pROM

GENERIC(

```
BIT_WIDTH:integer:=1;
```

```
READ_MODE:bit:='0';
```

```
RESET  MODE:string:="SYNC";
```

```
INIT_RAM_00:bit_vector=X"9C23645D0F78986FF  
C3E36E141541B95C19F2F7164085E631A819860D8FF0000";
```

```
INIT_RAM_01.bit_vector:=X"000000000000000000  
00000000000000000000000000000000FFFFFBD CF"
```

$$);$$

PORT(

```
DO:OUT std_logic_vector(31 downto 0):=conv_std
logic_vector(0,32);
```

```
CLK,CE,OCE,RESET:IN std logic;
```

```
AD:IN std logic vector(13 downto 0)
```

$$);$$

END COMPONENT;

uut:pROM

## GENERIC MAP(

BIT WIDTH=>1,

READ MODE=>'0'.

```
RESET  MODE=>"SYNC",
```

```
INIT_RAM_00=>"9C23645D0F78986FFC3E36
E141541B95C19F2F7164085E631A819860D8FF0000".
```

[illegible]

)

PORT MAP(

DO=>do.

AD=>ad,

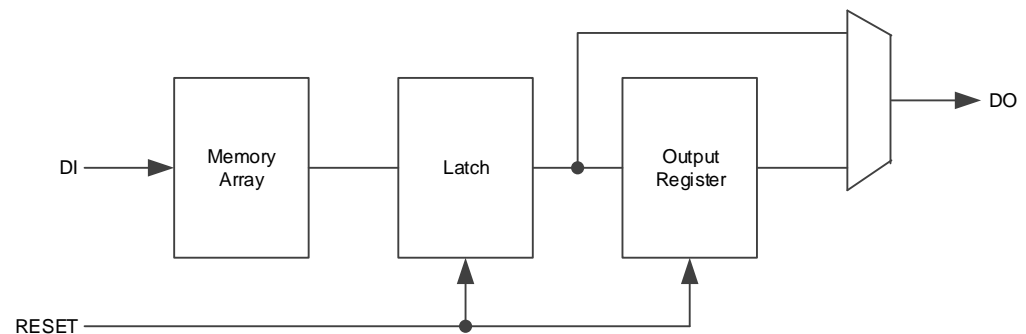
CLK=>clk,

```
CE=>ce,  
OCE=>oce,  
RESET=>reset  
);
```

# 4 BSRAM 出力のリセット

出力モジュールは **RESET** 信号をサポートします。リセットされると、**0** を出力することになります。そのブロック図は図 4-1 に示す通りです。

図 4-1 出力リセットのブロック図



**RESET** 信号が有効な場合(アクティブ High)、出力ポートは **0** を出力します。

同期リセット及び非同期リセットがサポートされます。ユーザーが直接プリミティブを呼び出しする場合、パラメータ **RESET\_MODE** を設定することでリセットモードを選択できます。**IP Core Generator** を使用する場合、**GUI** でリセットモードを選択できます。詳細については **6IP** の呼び出しを参照してください。

**RESET** 信号はラッチ及び出力レジスタをリセットします。そのため、**RESET** 信号が有効な場合、ユーザーがレジスタ出力モードまたはバイパス出力モードを使用しているかに関わらず、出力は **0** になります。

図 4-2~図 4-5 は各モードにおけるリセットタイミング図です。そのうち **DO\_RAM** はメモリアレイのデータであり、**DO** は出力ポートのデータです。

レジスタ出力モード：

- 同期リセットが有効な場合、**CLK** の立ち上がりエッジで **DO** が **0** にリセットされます。
- 非同期リセットが有効な場合、**DO** はすぐに **0** にリセットされます。

- リセットが無効で OCE が有効な場合、DO は"DO\_RAM"を出力します。
- リセットが無効で OCE が無効な場合、DO は前の出力データを保持します。

バイパス出力モード：

- 同期リセットが有効な場合、CLK の立ち上がりエッジで DO が 0 にリセットされます。
- 非同期リセットが有効な場合、DO はすぐに 0 にリセットされます。
- リセットが無効な場合、OCE が有効か無効かにかかわらず、DO は "DO\_RAM"を出力します。

図 4-2 同期リセットのタイミング図(Pipeline モード)

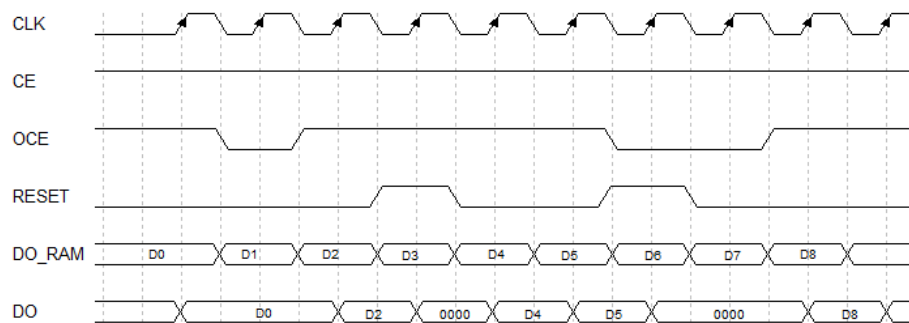


図 4-3 同期リセットのタイミング図(Bypass モード)

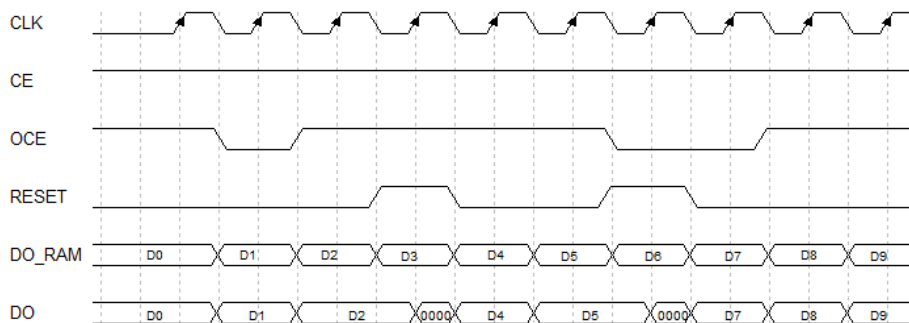


図 4-4 非同期リセットのタイミング図(Pipeline モード)

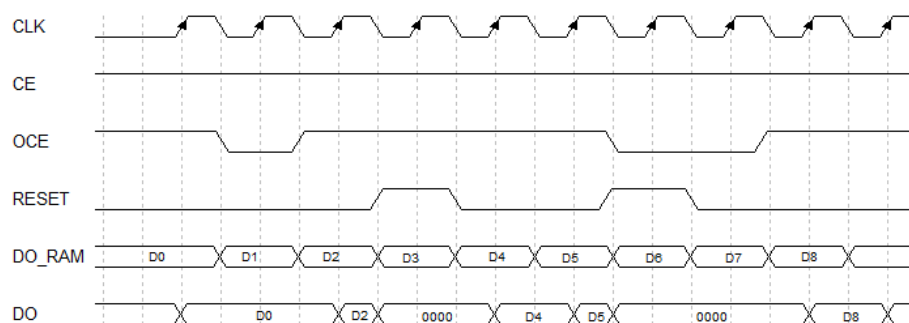
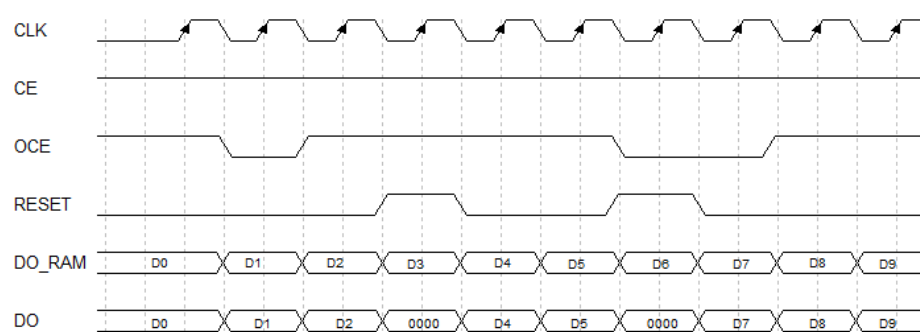


図 4-5 非同期リセットのタイミング図(Bypass モード)





# 5 SSRAM プリミティブ

Shadow SRAM は、分散スタティック RAM で、シングルポート・モード、セミ・デュアルポート・モード、および読み出し専用モードに構成できます(表 5-1)。

表 5-1 SSRAM のモード

プリミティブ	説明
RAM16S1	アドレス深さが 16、データ幅が 1 のシングルポート SSRAM
RAM16S2	アドレス深さが 16、データ幅が 2 のシングルポート SSRAM
RAM16S4	アドレス深さが 16、データ幅が 4 のシングルポート SSRAM
RAM16SDP1	アドレス深さが 16、データ幅が 1 のセミ・デュアルポート SSRAM
RAM16SDP2	アドレス深さが 16、データ幅が 2 のセミ・デュアルポート SSRAM
RAM16SDP4	アドレス深さが 16、データ幅が 4 のセミ・デュアルポート SSRAM
ROM16	アドレス深さが 16、データ幅が 1 の ROM

注記：

GW5AST-138B、GW5A-138B、GW5AT-138B、GW5AS-138B、GW5AT-75B、GW5A-25A、GW5AS-25A、および GW5AR-25A デバイスは、RAM16S1、RAM16S2、RAM16S4、RAM16SDP1、RAM16SDP2、および RAM16SDP4 プリミティブをサポートしていません。

## 5.1 RAM16S1

### プリミティブの紹介

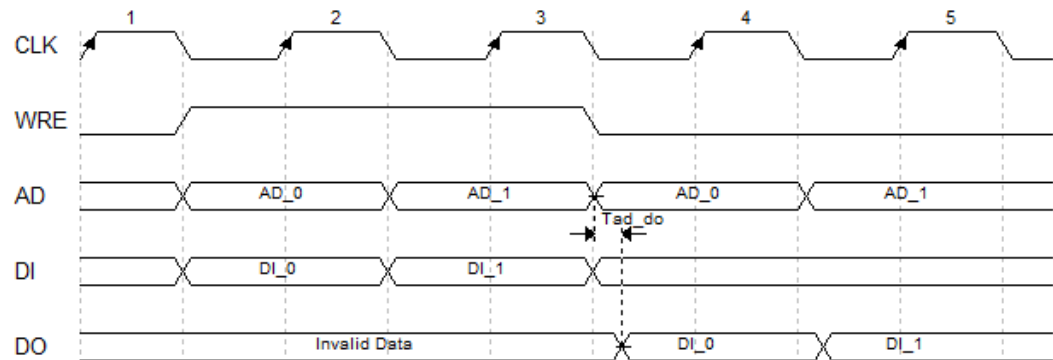
RAM16S1(16-Deep by 1-Wide Single-port SSRAM)は、アドレス深さが 16、データ幅が 1 のシングルポート SSRAM です。

### 機能の説明

RAM16S1 はデータ幅が 1 のシングルポート SSRAM で、その読み出し

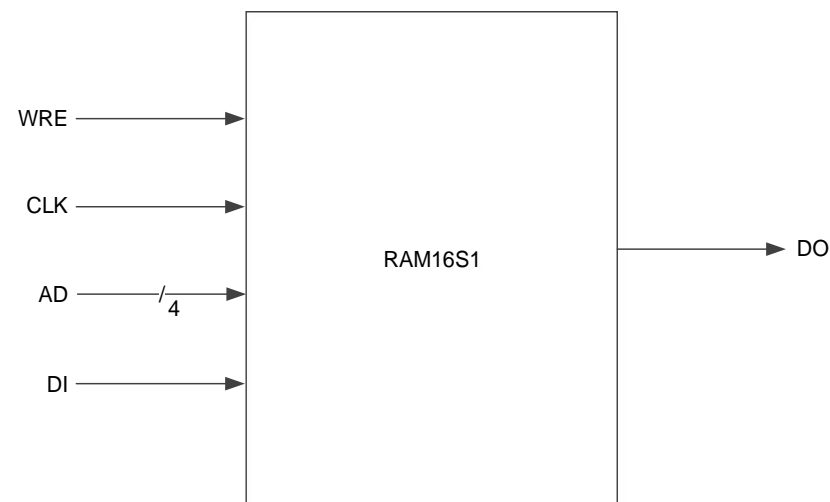
アドレスと書き込みアドレスは同じです。WRE が High の場合に書き込みが実行され、この場合、CLK の立ち上がりエッジでデータがメモリの対応するアドレスにロードされます。読み出し操作では、対応するアドレスのデータが出力されます。つまり、SSRAM は、CFU の LUT によって構成されており、同期的に書き込まれ、非同期的に読み出されます。ただし、必要な場合は、各 LUT に関連付けられるレジスタを使用して同期読み出しを実現することもできます。その Normal モードのタイミング図を図 5-1 に示します。

図 5-1 RAM16S1 モードのタイミング図



## ポート図

図 5-2 RAM16S1 のポート図



## ポートの説明

表 5-2 RAM16S1 のポートの説明

ポート	I/O	説明
DI	入力	データ入力信号
CLK	入力	クロック入力信号
WRE	入力	書き込みイネーブル入力信号
AD[3:0]	入力	アドレス入力信号
DO	出力	データ出力信号

## パラメータの説明

表 5-3 RAM16S1 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT_0	16'h0000~16'hffff	16'h0000	RAM16S1 の初期値

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

### Verilog でのインスタンス化 :

```
RAM16S1 instName(
    .DI(DI),
    .WRE(WRE),
    .CLK(CLK),
    .AD(AD[3:0]),
    .DO(DOUT)
);
defparam instName.INIT_0=16'h1100;
```

### VHDL でのインスタンス化 :

```
COMPONENT RAM16S1
    GENERIC (INIT:bit_vector:=X"0000");
    PORT(
        DO:OUT std_logic;
        DI:IN std_logic;
        CLK:IN std_logic;
        WRE:IN std_logic;
        AD:IN std_logic_vector(3 downto 0)
```

```

    );
END COMPONENT;
uut:RAM16S1
    GENERIC MAP(INIT=>X"0000")
    PORT MAP (
        DO=>DOUT,
        DI=>DI,
        CLK=>CLK,
        WRE=>WRE,
        AD=>AD
    );

```

## 5.2 RAM16S2

### プリミティブの紹介

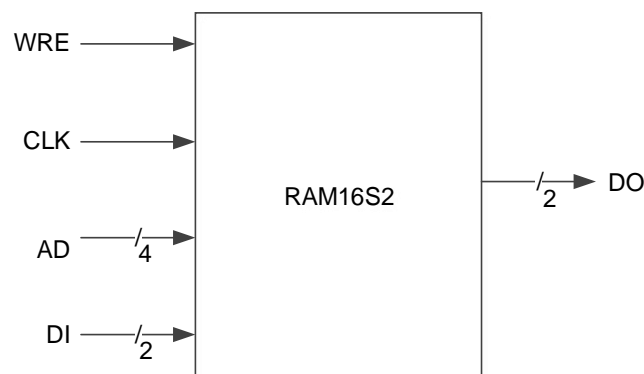
RAM16S2(16-Deep by 2-Wide Single-port SSRAM)は、アドレス深さが16、データ幅が2のシングルポート SSRAM です。

### 機能の説明

RAM16S2 はデータ幅が2のシングルポート SSRAM で、その読み出しアドレスと書き込みアドレスは同じです。WRE が High の場合に書き込みが実行され、この場合、CLK の立ち上がりエッジでデータがメモリの対応するアドレスにロードされます。読み出し操作では、対応するアドレスのデータが出力されます。つまり、SSRAM は、CFU の LUT によって構成されており、同期的に書き込まれ、非同期的に読み出されます。ただし、必要な場合は、各 LUT に関連付けられるレジスタを使用して同期読み出しを実現することもできます。そのタイミング図を図 5-1 に示します。

### ポート図

図 5-3 RAM16S2 のポート図



## ポートの説明

表 5-4 RAM16S2 のポートの説明

ポート	I/O	説明
DI[1:0]	入力	データ入力信号
CLK	入力	クロック入力信号
WRE	入力	書き込みイネーブル入力信号
AD[3:0]	入力	アドレス入力信号
DO[1:0]	出力	データ出力信号

## パラメータの説明

表 5-5 RAM16S2 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT_0~ INIT_1	16'h0000~16'hffff	16'h0000	RAM16S2 の初期値

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

### Verilog でのインスタンス化：

```
RAM16S2 instName(
    .DI(DI[1:0]),
    .WRE(WRE),
    .CLK(CLK),
    .AD(AD[3:0]),
    .DO(DOUT[1:0])
);
defparam instName.INIT_0=16'h0790;
defparam instName.INIT_1=16'h0f00;
```

### VHDL でのインスタンス化：

```
COMPONENT RAM16S2
    GENERIC (INIT_0:bit_vector:=X"0000";
             INIT_1:bit_vector:=X"0000"
    );
    PORT(
        DO:OUT std_logic_vector(1 downto 0);
```

```

        DI:IN std_logic_vector(1 downto 0);
        CLK:IN std_logic;
        WRE:IN std_logic;
        AD:IN std_logic_vector(3 downto 0)

    );
END COMPONENT;
uut:RAM16S2
    GENERIC MAP(INIT_0=>X"0000",
                INIT_1=>X"0000"
    )
    PORT MAP (
        DO=>DOUT,
        DI=>DI,
        CLK=>CLK,
        WRE=>WRE,
        AD=>AD
    );

```

## 5.3 RAM16S4

### プリミティブの紹介

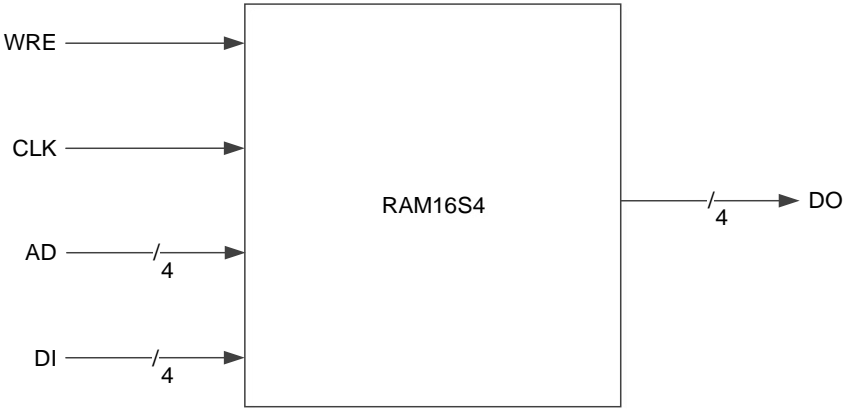
RAM16S4(16-Deep by 4-Wide Single-port SSRAM)は、アドレス深さが 16、データ幅が 4 のシングルポート SSRAM です。

### 機能の説明

RAM16S4 はデータ幅が 4 のシングルポート SSRAM で、その読み出しアドレスと書き込みアドレスは同じです。WRE が High の場合に書き込みが実行され、この場合、CLK の立ち上がりエッジでデータがメモリの対応するアドレスにロードされます。読み出し操作では、対応するアドレスのデータが出力されます。つまり、SSRAM は、CFU の LUT によって構成されており、同期的に書き込まれ、非同期的に読み出されます。ただし、必要な場合は、各 LUT に関連付けられるレジスタを使用して同期読み出しを実現することもできます。そのタイミング図を図 5-1 に示します。

ポート図

図 5-4 RAM16S4 のポート図



ポートの説明

表 5-6 RAM16S4 のポートの説明

ポート	I/O	説明
DI[3:0]	入力	データ入力信号
CLK	入力	クロック入力信号
WRE	入力	書き込みイネーブル入力信号
AD[3:0]	入力	アドレス入力信号
DO[3:0]	出力	データ出力信号

パラメータの説明

表 5-7 RAM16S4 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT_0~ INIT_3	16'h0000~16'hffff	16'h0000	RAM16S4 の初期値

プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

Verilog でのインスタンス化 :

```
RAM16S4 instName(  
    .DI(DI[3:0]),  
    .WRE(WRE),  
    .CLK(CLK),
```

```

        .AD(AD[3:0]),
        .DO(DOUT[3:0])
    );
    defparam instName.INIT_0=16'h0450;
    defparam instName.INIT_1=16'h1ac3;
    defparam instName.INIT_2=16'h1240;
    defparam instName.INIT_3=16'h045c;
VHDL でのインスタンス化 :
    COMPONENT RAM16S4
        GENERIC (INIT_0:bit_vector:=X"0000";
                  INIT_1:bit_vector:=X"0000";
                  INIT_2:bit_vector:=X"0000";
                  INIT_3:bit_vector:=X"0000"
        );
        PORT(
            DO:OUT std_logic_vector(3 downto 0);
            DI:IN std_logic_vector(3 downto 0);
            CLK:IN std_logic;
            WRE:IN std_logic;
            AD:IN std_logic_vector(3 downto 0)
        );
    END COMPONENT;
    uut:RAM16S4
        GENERIC MAP(INIT_0=>X"0000",
                    INIT_1=>X"0000",
                    INIT_2=>X"0000",
                    INIT_3=>X"0000"
        )
        PORT MAP (
            DO=>DOUT,
            DI=>DI,
            CLK=>CLK,
            WRE=>WRE,
            AD=>AD
        );

```



## 5.4 RAM16SDP1

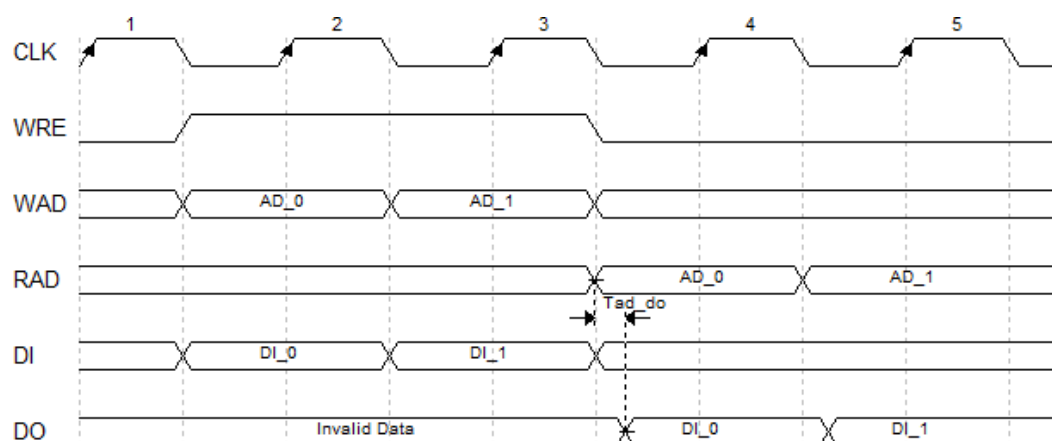
### プリミティブの紹介

RAM16SDP1(16-Deep by 1-Wide Semi Dual-port SSRAM)は、アドレス深さが 16、データ幅が 1 のセミ・デュアルポート SSRAM です。

### 機能の説明

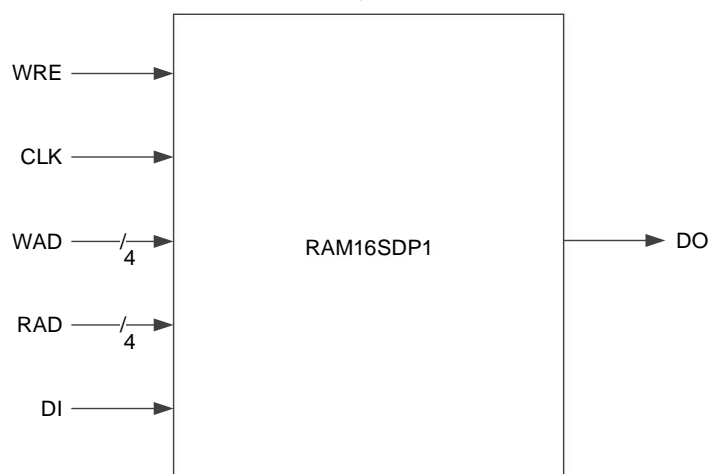
RAM16SDP1 には、書き込みアドレス WAD および読み出しアドレス RAD があります。この 2 つのアドレスポートは非同期です。WRE が High の場合に書き込みが実行されます。この場合、CLK の立ち上がりエッジでデータがメモリの対応する書き込みアドレスにロードされます。読み出し操作では、対応する読み出しアドレスのデータが出力されます。その Normal モードのタイミング図を図 5-5 に示します。

図 5-5 RAM16SDP1 モードのタイミング図



### ポート図

図 5-6 RAM16SDP1 のポート図



## ポートの説明

表 5-8 RAM16SDP1 のポートの説明

ポート	I/O	説明
DI	入力	データ入力信号
CLK	入力	クロック入力信号
WRE	入力	書き込みイネーブル入力信号
WAD[3:0]	入力	書き込みアドレス信号
RAD[3:0]	入力	読み出しアドレス信号
DO	出力	データ出力信号

## パラメータの説明

表 5-9 RAM16SDP1 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT_0	16'h0000~16'hffff	16'h0000	RAM16SDP1 の初期値

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP](#) の呼び出しを参照してください。

### Verilog でのインスタンス化：

```
RAM16SDP1 instName(
    .DI(DI),
    .WRE(WRE),
    .CLK(CLK),
    .WAD(WAD[3:0]),
    .RAD(RAD[3:0]),
    .DO(DOUT)
);
defparam instName.INIT_0=16'h0100;
```

### VHDL でのインスタンス化：

```
COMPONENT RAM16SDP1
    GENERIC (INIT_0:bit_vector:=X"0000");
    PORT(
        DO:OUT std_logic;
```

```

        DI:IN std_logic;
        CLK:IN std_logic;
        WRE:IN std_logic;
        WAD:IN std_logic_vector(3 downto 0);
        RAD:IN std_logic_vector(3 downto 0)

    );
END COMPONENT;
 uut:RAM16SDP1
    GENERIC MAP(INIT_0=>X"0000")
    PORT MAP (
        DO=>DOUT,
        DI=>DI,
        CLK=>CLK,
        WRE=>WRE,
        WAD=>WAD,
        RAD=>RAD
    );

```

## 5.5 RAM16SDP2

### プリミティブの紹介

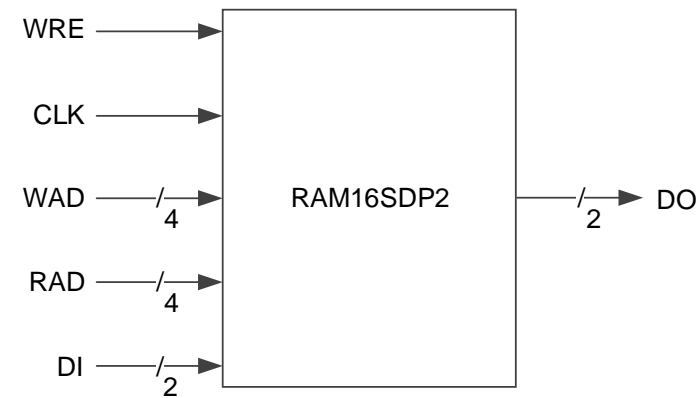
RAM16SDP2(16-Deep by 2-Wide Semi Dual-port SSRAM)は、アドレス深さが 16、データ幅が 2 のセミ・デュアルポート SSRAM です。

### 機能の説明

RAM16SDP2 には、書き込みアドレス WAD および読み出しアドレス RAD があります。この 2 つのアドレスポートは非同期です。WRE が High の場合に書き込みが実行されます。この場合、CLK の立ち上がりエッジでデータがメモリの対応する書き込みアドレスにロードされます。読み出し操作では、対応する読み出しアドレスのデータが出力されます。そのタイミング図を図 5-5 に示します。

ポート図

図 5-7 RAM16SDP2 のポート図



ポートの説明

表 5-10 RAM16SDP2 のポートの説明

ポート	I/O	説明
DI[1:0]	入力	データ入力信号
CLK	入力	クロック入力信号
WRE	入力	書き込みイネーブル入力信号
WAD[3:0]	入力	書き込みアドレス信号
RAD[3:0]	入力	読み出しアドレス信号
DO[1:0]	出力	データ出力信号

パラメータの説明

表 5-11 RAM16SDP2 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT_0~ INIT_1	16'h0000~16'hffff	16'h0000	RAM16SDP2 の初期値

プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

**Verilog** でのインスタンス化：

```
RAM16SDP2 instName(  
    .DI(DI[1:0]),  
    .WRE(WRE),
```

```

        .CLK(CLK),
        .WAD(WAD[3:0]),
        .RAD(RAD[3:0]),
        .DO(DOUT[1:0])
    );
    defparam instName.INIT_0=16'h5600;
    defparam instName.INIT_1=16'h0af0;
VHDL でのインスタンス化 :
    COMPONENT RAM16SDP2
        GENERIC (INIT_0:bit_vector:=X"0000";
                 INIT_1:bit_vector:=X"0000"
        );
        PORT(
            DO:OUT std_logic_vector(1 downto 0);
            DI:IN std_logic_vector(1 downto 0);
            CLK:IN std_logic;
            WRE:IN std_logic;
            WAD:IN std_logic_vector(3 downto 0);
            RAD:IN std_logic_vector(3 downto 0)
        );
    END COMPONENT;
    uut:RAM16SDP2
        GENERIC MAP(INIT_0=>X"0000",
                    INIT_1=>X"0000"
        )
        PORT MAP (
            DO=>DOUT,
            DI=>DI,
            CLK=>CLK,
            WRE=>WRE,
            WAD=>WAD,
            RAD=>RAD
        );

```

# 5.6 RAM16SDP4

## プリミティブの紹介

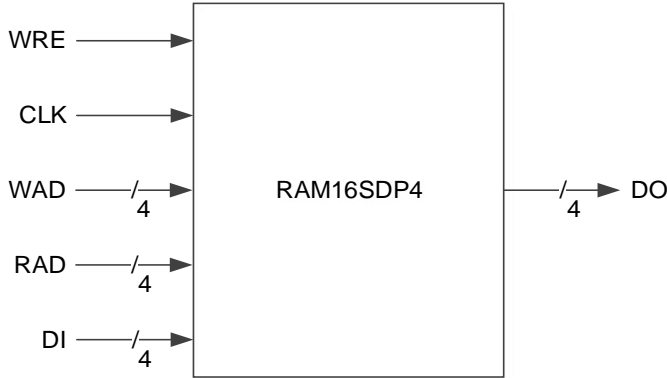
RAM16SDP4(16-Deep by 4-Wide Semi Dual-port SSRAM)は、アドレス深さが 16、データ幅が 4 のセミ・デュアルポート SSRAM です。

## 機能の説明

RAM16SDP4 には、書き込みアドレス WAD および読み出しアドレス RAD があります。この 4 つのアドレスポートは非同期です。WRE が High の場合に書き込みが実行されます。この場合、CLK の立ち上がりエッジでデータがメモリの対応する書き込みアドレスにロードされます。読み出し操作では、対応する読み出しアドレスのデータが出力されます。そのタイミング図を図 5-5 に示します。

## ポート図

図 5-8 RAM16SDP4 のポート図



## ポートの説明

表 5-12 RAM16SDP4 のポートの説明

ポート	I/O	説明
DI[3:0]	入力	データ入力信号
CLK	入力	クロック入力信号
WRE	入力	書き込みイネーブル入力信号
WAD[3:0]	入力	書き込みアドレス信号
RAD[3:0]	入力	読み出しアドレス信号
DO[3:0]	出力	データ出力信号

## パラメータの説明

表 5-13 RAM16SDP4 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT_0~ INIT_3	16'h0000~16'hffff	16'h0000	RAM16SDP4 の初期値

## プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

### Verilog でのインスタンス化：

```
RAM16SDP4 instName(
    .DI(DI[3:0]),
    .WRE(WRE),
    .CLK(CLK),
    .WAD(WAD[3:0]),
    .RAD(RAD[3:0]),
    .DO(DOUT[3:0])
);
defparam instName.INIT_0=16'h0340;
defparam instName.INIT_1=16'h9065;
defparam instName.INIT_2=16'hac12;
defparam instName.INIT_3=16'h034c;
```

### VHDL でのインスタンス化：

```
COMPONENT RAM16SDP2
    GENERIC (INIT_0:bit_vector:=X"0000";
             INIT_1:bit_vector:=X"0000";
             INIT_2:bit_vector:=X"0000";
             INIT_3:bit_vector:=X"0000";
    );
    PORT(
        DO:OUT std_logic_vector(3 downto 0);
        DI:IN std_logic_vector(3 downto 0);
        CLK:IN std_logic;
        WRE:IN std_logic;
        WAD:IN std_logic_vector(3 downto 0);
        RAD:IN std_logic_vector(3 downto 0)
```

```

    );
END COMPONENT;
uut:RAM16SDP2
    GENERIC MAP(INIT_0=>X"0000",
                INIT_1=>X"0000",
                INIT_2=>X"0000",
                INIT_3=>X"0000"
    )
    PORT MAP (
        DO=>DOUT,
        DI=>DI,
        CLK=>CLK,
        WRE=>WRE,
        WAD=>WAD,
        RAD=>RAD
    );

```

## 5.7 ROM16

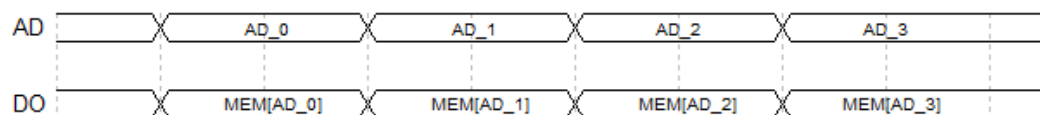
### プリミティブの紹介

ROM16 は、アドレス深さが 16、データ幅が 1 の読み出し専用メモリで、メモリの内容は INIT によって初期化されます。

### 機能の説明

ROM16 の読み出し操作では、対応するアドレスのデータが出力されます。そのタイミング図を図 5-9 に示します。

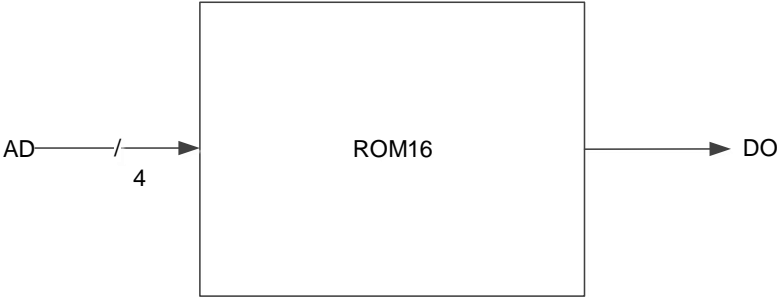
図 5-9 ROM16 モードのタイミング図





ポート図

図 5-10 ROM16 のポート図



ポートの説明

表 5-14 ROM16 のポートの説明

ポート	I/O	説明
AD[3:0]	入力	アドレス入力信号
DO	出力	データ出力信号

パラメータの説明

表 5-15 ROM16 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT_0	16'h0000~16'hffff	16'h0000	ROM16 の初期値

プリミティブのインスタンス化

プリミティブを直接インスタンス化するか、IP Core Generator で生成できます。詳しくは、[6 IP の呼び出し](#)を参照してください。

Verilog でのインスタンス化 :

```
ROM16 instName (  
    .AD(AD[3:0]),  
    .DO(DOUT)  
);  
defparam instName.INIT_0=16'hfc00;
```

VHDL でのインスタンス化 :

```
COMPONENT ROM16  
    GENERIC (INIT:bit_vector:=X"0000");  
    PORT(  
        DO:OUT std_logic;
```

```
        AD:IN std_logic_vector(3 downto 0)
    );
END COMPONENT;
uut:ROM16
    GENERIC MAP(INIT=>X"0000")
    PORT MAP (
        DO=>DOUT,
        AD=>AD
    );
```

# 6 IP の呼び出し

Gowin ソフトウェアの IP Core Generator は、IP コアの呼び出しをサポートします。ユーザーは、IP Core Generator の GUI でデータ幅、アドレス深さ、書き込みモード、及び読み出しモードを設定して IP モジュールを生成することができます。さらに、BSRAM と SSRAM を実装する方法は他に 2 つあります。1 つは、Gowin ソフトウェアのライブラリ・ファイルを読み出し、そのポート及びパラメータを設定して IP モジュールを生成する方法です。もう 1 つは、コードの合成時、合成ツールで自動的に BSRAM または SSRAM モードに合成する方法です。

IP Core Generator では、BSRAM モジュールは、シングルポート・モード、セミ・デュアルポート・モード、ECC 機能付きのセミ・デュアルポート・モード、デュアルポート・モード、および読み出し専用モードをサポートし、SSRAM モジュールはシングルポート・モード、セミ・デュアルポート・モード、および読み出し専用モードをサポートします。以下では、デュアルポート・モードの BSRAM、ECC 機能付きのセミ・デュアルポート・モードの BSRAM、およびシングルポート・モード SSRAM のを例に IP の呼び出しを紹介し、その他のモードについては、デュアルポート・モードの BSRAM とシングルポート・モードの SSRAM の呼び出し方法を参照してください。

## 6.1 デュアルポート・モードの BSRAM

デュアルポート・モードの BSRAM は、プリミティブの DPB および DPX9B により実装できます。IP Core Generator のインターフェースで“DPB”をクリックすると、右側に DPB の概要が表示されます。

### IP の構成

IP Core Generator インターフェースで“DPB”をダブルクリックすると、DPB の“IP Customization”ウィンドウがポップアップします。このウィンドウには“File”構成タブ、“Options”構成タブ、およびポート図があります(図 6-1)。

図 6-1 DPB IP の構成ウィンドウ

1. **File 構成タブ**。**File 構成タブ**は、生成される IP ファイルの構成に使用されます。
- **Device** : 対象デバイス。
  - **Part Number** : 部品番号。
  - **Language** : IP を実現するハードウェア記述言語。右側のドロップダウン・リストからターゲット言語(**Verilog** または **VHDL**)を選択します。
  - **Module Name** : 生成される IP ファイルのモジュール名。右側のテキストボックスで編集できます。**Module Name** をプリミティブ名と同じにすることはできません。同じである場合、エラーが報告されます。
  - **File Name** : 生成される IP ファイルのファイル名。右側のテキストボックスで再編集できます。
  - **Create In** : 生成される IP ファイルのパス。右側のテキストボックスでパスを直接編集するか、テキストボックスの右側にある選択ボタンを使用してパスを選択できます。

2. Options 構成タブ。Options 構成タブは IP のカスタマイズに使用されます。図 6-1 に示すように、A ポートと B ポートがあります。
  - Data Width & Address Depth : アドレス深さ(Address Depth)とデータ幅(Data Width)を構成します。構成されたアドレス深さとデータ幅を 1 つのモジュールで実装できない場合、IP Core Generator は複数のモジュールをインスタンス化して実装します。
  - Resource Usage : 現在の構成で使用される Block RAM、DFF、LUT、MUX の数を計算し、表示します。
  - Read/Write Mode : 読み出し/書き込みモードを構成します。DPB は以下のモードをサポートします。
    - 2 つの読み出しモード : Bypass と Pipeline。
    - 3 つの書き込みモード : Normal、Write-Through、Read-before-Write。
  - Reset Mode : リセットモード(同期モード “Synchronous” または非同期モード “Asynchronous” )を選択します。
  - Initialization : 初期値を構成します。初期値は、2 進数、16 進数、またはアドレス付き 16 進数の形式で初期化ファイルに書き込まれます。“Memory Initialization File” で選択される初期化ファイルは、手動で入力するか、Gowin ソフトウェアの “File > New > Memory Initialization File” をクリックすることにより生成できます。生成方法の詳細は『Gowin ソフトウェア ユーザーガイド([SUG100](#))』を参照してください。初期化ファイルの形式については [7 初期化ファイル](#)を参照してください。

注記 :

- Options 構成タブでは、DPB の Port A と Port B のアドレス深さ、データ幅、および読み出し/書き込みモードを個別に構成できます。
- DPB の Port A と Port B は同じ BSRAM に対して読み出しと書き込みを行うため、Port A と Port B の Address Depth\*Data Width の結果は同じでなければなりません。
- Options 構成の初期化ファイル(Memory initialization File)のデータ幅は Dimension Match で選択した Port のデータ幅と一致しなければなりません。
- Port A と Port B の Address Depth\*Data Width の結果が一致しない場合、Error メッセージがポップアップします。
- データ幅が一致しない場合、生成される DPB インスタンスの Init 値はデフォルトで 0 となり、そして Output ウィンドウで以下のメッセージがポップアップします : Error (MG2105): Initial values' width is unequal to user's width。

3. ポート図

- ポート図 : 現在の IP Core の構成結果を表示します。入力・出力ポートおよびその幅は Options 構成に従ってリアルタイムで更新されます (図 6-1)。

- Options 構成での Port A と Port B のアドレス深さ “Address Depth” の構成はアドレスの幅に影響し、データ幅 “Data Width” の構成は入力データと出力データの幅に影響します。

#### 生成されるファイル

IP の構成が完了したら、構成ファイルの “File Name” によって命名された 3 つのファイルが生成されます：

- “gowin\_dpb.v” は完全な verilog モジュールです。
- “gowin\_dpb\_tmp.v” は IP のテンプレートファイルです。
- “gowin\_dpb.ipc” は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

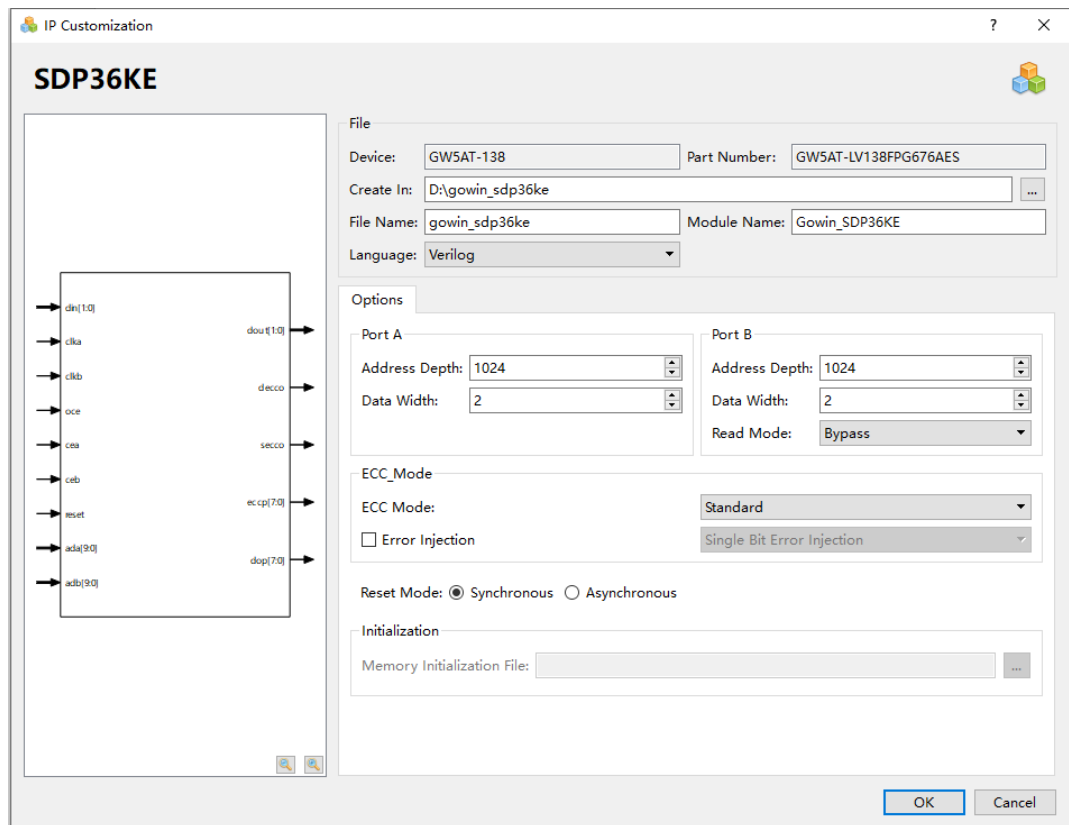
## 6.2 ECC 機能付きのセミ・デュアルポート・モードの BSRAM

ECC 機能付きのセミ・デュアルポート・モードの BSRAM(SDP36KE) は、プリミティブの SDP36KE により実装できます。IP Core Generator のインターフェースで “SDP36KE” をクリックすると、右側に SDP36KE の概要が表示されます。

#### IP の構成

IP Core Generator インターフェースで “SDP36KE” をダブルクリックすると、SDP36KE の IP Customization ウィンドウがポップアップします。このウィンドウには “File” 構成タブ、“Options” 構成タブ、およびポート図があります(図 6-2)。

図 6-2 SDP36KE IP の構成ウィンドウ



1. **File 構成タブ。** File 構成タブは、生成される IP ファイルの構成に使用されます。SDP36KE の File 構成タブの使用はデュアルポート・モードと同様です。詳細については、6.1 デュアルポート・モードの BSRAM を参照してください。
2. **Options 構成タブ。** Options 構成タブは IP のカスタマイズに使用されます。図 6-1 に示すように、A ポートと B ポートがあります。
  - **Data Width & Address Depth :** アドレス深さ(Address Depth)とデータ幅(Data Width)を構成します。構成されたアドレス深さとデータ幅を 1 つのモジュールで実装できない場合、IP Core Generator は複数のモジュールをインスタンス化して実装します。
  - **ECC Mode :** サポートされる ECC Mode を次に示します。
    - Standard : Encode と Decode をサポート。
    - Encode-Only : Encode のみをサポート。
    - Decode-Only : Decode のみをサポート。
  - **Error Injection :** 挿入されるエラーのビット数を構成します。SDP36KE は、次のエラーのビット数をサポートします。
    - Single Bit Error Injection : シングルビット・エラーを挿入。
    - Double Bit Error Injection : ダブルビット・エラーを挿入。

- Single and Double Bit Error Injection : シングルビット・エラーとダブルビット・エラーを挿入。
- Reset Mode : リセットモード(同期モード “Synchronous” または非同期モード “Asynchronous” )を選択します。

### 3. ポート図

- ポート図 : 現在の IP Core の構成結果を表示します。入力・出力ポートおよびその幅は Options 構成に従ってリアルタイムで更新されます(図 6-2)。
- Options 構成での Port A と Port B のアドレス深さ “Address Depth” の構成はアドレスの幅に影響し、データ幅 “Data Width” の構成は入力データと出力データの幅に影響します。

### 生成されるファイル

IP の構成が完了したら、“File Name” によって命名された 3 つのファイルが生成されます :

- “gowin\_sdp36ke.v” は完全な verilog モジュールです。
- ” gowin\_sdp36ke\_tmp.v” は IP のテンプレートファイルです。
- “gowin\_sdp36ke.ipc.ipc” は IP の構成ファイルです。

注記 :

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

## 6.3 シングルポート・モードの SSRAM

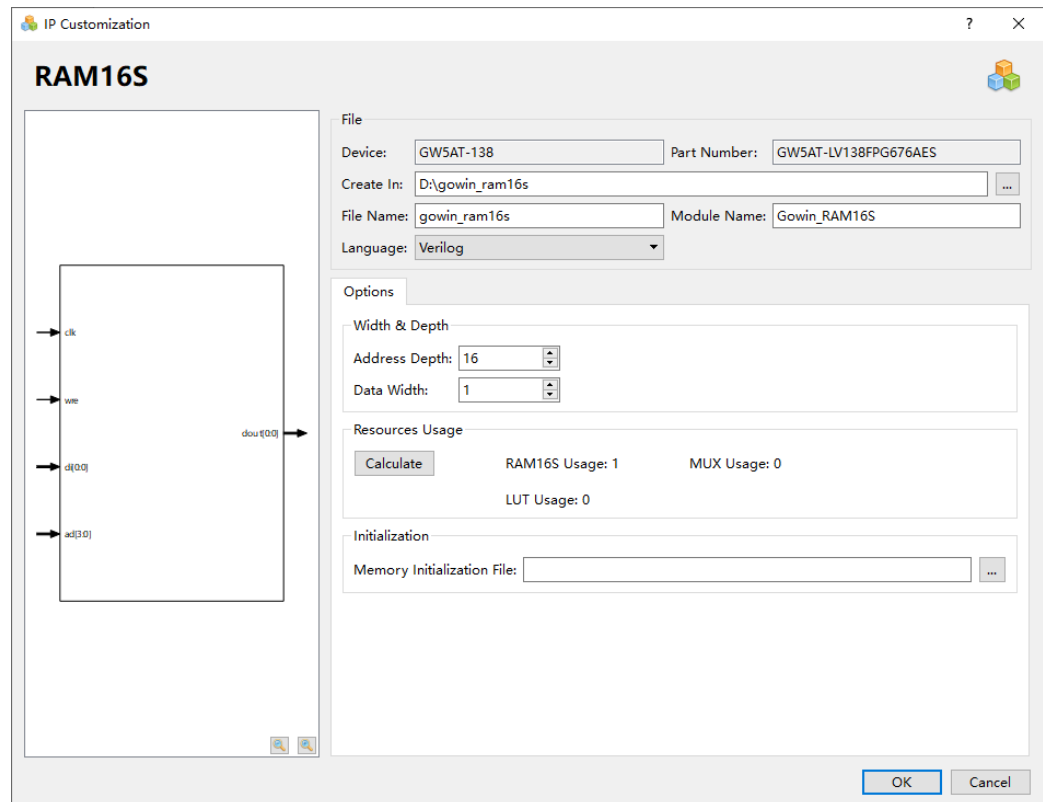
RAM16S(シングルポート・モード)は、プリミティブの RAM16S1、RAM16S2、および RAM16S4 により実装できます。IP Core Generator のインターフェースで “RAM16S” をクリックすると、右側に RAM16S の概要が表示されます。

### IP の構成

IP Core Generator インターフェースで “RAM16S” をダブルクリックすると、RAM16S の “IP Customization” ウィンドウがポップアップします。このウィンドウには “File” 構成タブ、“Options” 構成タブ、およびポート図があります(図 6-3)。



図 6-3 RAM16S IP の構成ウィンドウ



1. **File 構成タブ。**File 構成タブは、生成される IP ファイルの構成に使用されます。RAM16S の File 構成タブの使用はデュアルポート・モードと同様です。詳細については、6.1 デュアルポート・モードの BSRAM を参照してください。
2. **Options 構成タブ。**Options 構成タブは IP のカスタマイズに使用されます。Options 構成タブを図 6-3 に示します。RAM16S の Options 構成タブの使用はデュアルポート・モードと同様です。詳細については、6.1 デュアルポート・モードの BSRAM を参照してください。
3. **ポート図**
  - **ポート図：**現在の IP Core の構成結果を表示します。入力・出力ポートおよびその幅は Options 構成に従ってリアルタイムで更新されます (図 6-3)。
  - **Options 構成でのアドレス深さ “Address Depth” の構成はアドレスの幅に影響し、データ幅 “Data Width” の構成は入力データと出力データの幅に影響します。**

### 生成されるファイル

IP の構成が完了したら、“File Name” によって命名された 3 つのファイルが生成されます：

- “gowin\_ram16s.v” は完全な verilog モジュールです。

- “gowin\_ram16s\_tmp.v” は IP のテンプレートファイルです。
- “gowin\_ram16s.ipc” は IP の構成ファイルです。

注記：

VHDL が設計の言語として選択されている場合、生成される最初の 2 つのファイル名のサフィックスは.vhd になります。

# 7 初期化ファイル

BSRAM および SSRAM では、メモリの各ビットを 0 または 1 に初期化できます。初期値は、2 進数、16 進数、またはアドレス付き 16 進数の形式で初期化ファイルに書き込まれます。

## 7.1 2 進数形式(Bin File)

Bin ファイルは 2 進数 0 と 1 から成るテキストファイルです。行の数はメモリのアドレス深さ、列の数はメモリのデータ幅を表します。

```
#File_format=Bin
#Address_depth=16
#Data_width=32
00001100000010000000001001000010000
10000000001001000001000000001000000
01000000010000000010000000010000000
00100000010000010011000000011000000
```

## 7.2 16 進数形式(Hex File)

Bin ファイルと同様で、Hex ファイルは 16 進数の 0~F で構成されています。行の数はメモリのアドレス深さを表し、各行のデータの 2 進数のビット数はメモリのデータ幅を表します。

```
#File_format=Hex
#Address_depth=8
#Data_width=16
3A40
A28E
0B52
1C49
D602
```

0801

03E6

4C18

## 7.3 アドレス付き 16 進数(Address-Hex File)

Address-Hex ファイルは、データ記録があるアドレスとデータを記録します。アドレスとデータはすべて 16 進法数の 0~F から成り、各行のコロンの前はアドレスで、コロンの後はデータです。ファイルでは、書き込みデータおよびそのアドレスのみ記録し、記録のないアドレスのデータはデフォルトで 0 です。

```
#File_format=AddrHex
```

```
#Address_depth=256
```

```
#Data_width=16
```

```
9:FFFF
```

```
23:00E0
```

```
2a:001F
```

```
30:1E00
```

