



LatticeECP2/M

メモリ使用ガイド

はじめに

このテクニカルノートはLatticeECP2™とLatticeECP2M™デバイスファミリのメモリ使用法について議論します。ispLEVERでこれらのデバイスファミリ用の組み込みブロックRAM (EBR)とPFUベースのメモリを集積する際におけるガイドとして、設計エンジニアに用いられることを意図しています。

これらデバイスのアーキテクチャは、FPGAオンチップメモリを用いるアプリケーションのためのリソースを提供します。sysMEM組み込みブロックRAM (EBR)とPFUベースの分散メモリは相互に補います。シングルポートRAM、デュアルポートRAM、擬似デュアルポートRAM、FIFOおよびROMメモリはEBRを用いることで構成できます。LUTとPFUは、分散シングルポートRAM、分散デュアルポートRAM、および分散ROMを実装することができます。

EBRブロックRAMとPFU RAMの機能はプリミティブと呼ばれ、本テクニカルノートの後ろで説明されます。設計者はispLEVERソフトウェアのIPexpress™ ツールによって、2つの異なる方法でメモリプリミティブを利用することができます。IPexpressで所望のメモリタイプとサイズの指定ができます。IPexpressはその使用を得て、メモリ・プリミティブの1つかそれ以上を用いることによって、必要なメモリを実装するネットリストを生成します。

このドキュメントの残りは、IPexpress、メモリモジュール、およびメモリプリミティブの活用方法について議論します。

LatticeECP2/Mデバイスにおけるメモリ

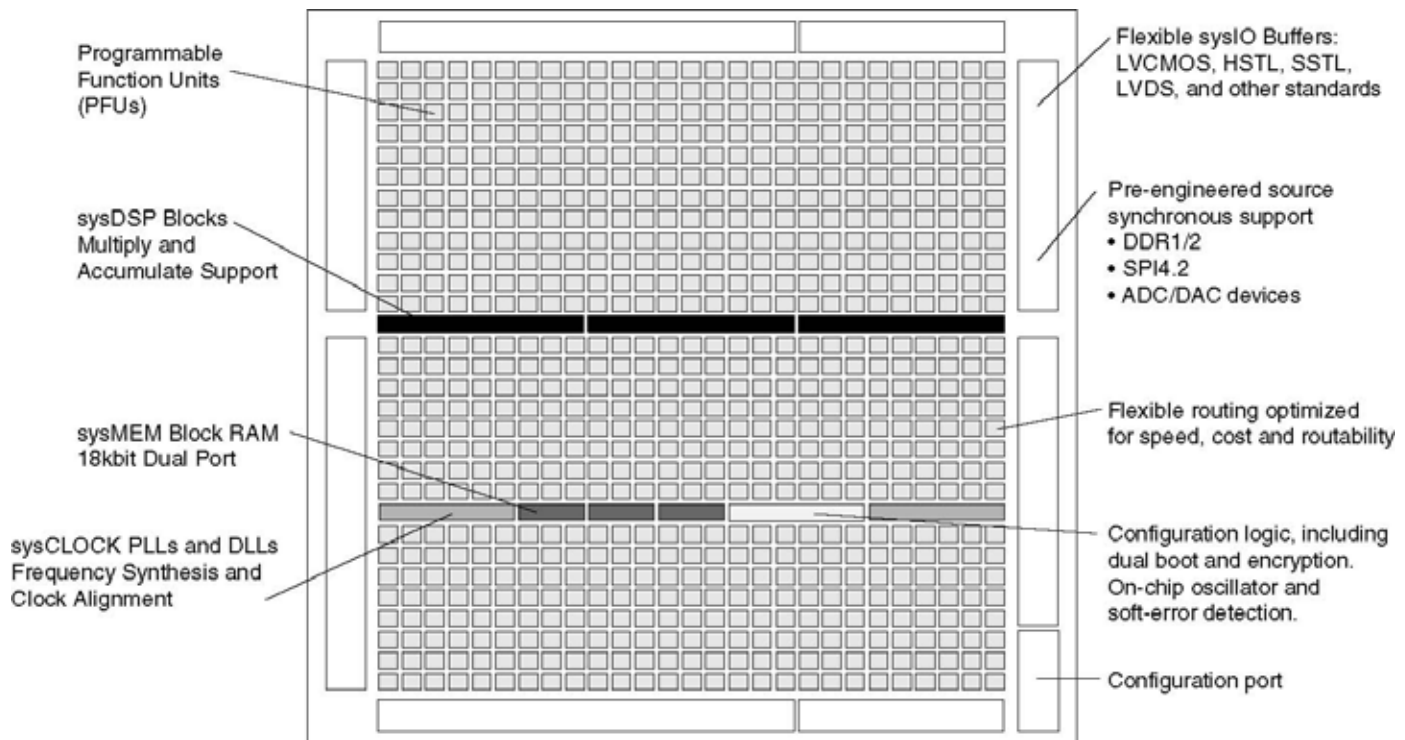
2種類の論理ブロック、即ちプログラマブル・ファンクションユニット(PFU)、およびRAM対応なしのプログラマブル・ファンクションユニット(PFF)です。PFUはロジック、演算、RAM、ROM、およびレジスタ機能のためのビルディングブロックを含みます。PFFブロックはロジック、演算、およびROM機能のためのビルディングブロックを含んでいます。複雑なデザインを短時間に、効率的に実装できる柔軟性のためにPFUとPFFブロックが共に最適化されています。論理ブロックは2次元配列でアレンジされています。同じタイプのブロックが（横方向の）列単位で用いられます。

sysMEM EBRブロックはLatticeECP2ファミリでは最大2列、LatticeECP2Mファミリには最大7列含まれています。sysMEM EBRは18Kの大きさで専用の高速メモリブロックです。各sysMEMブロックはさまざまな深さと幅でRAMやROMを構成することができます。

表 11-1 LatticeECP2/M LUTとメモリの規模

LUT数	6K	12K	20K		35K		50K		70K		100K
デバイス	ECP2-6	ECP2-12	ECP2-20	ECP2M-20	ECP2-35	ECP2M-35	ECP2-50	ECP2M-50	ECP2-70	ECP2M-70	ECP2M-100
LUT数 (k)	6	12	21	19	32	34	48	48	68	67	95
分散RAMサイズ (kb)	12	24	42	41	65	71	96	101	136	145	202
EBRブロック数	3	12	15	66	18	114	21	225	60	246	288
EBRサイズ (kb)	55	221	276	1217	332	2101	387	4147	1106	4534	5308

図 11-1 簡略化したブロック図（上面図）～ LatticeECP2-6 デバイス



IPexpressの活用

設計者は、IPexpressを利用することで、種々メモリを容易に指定することができます。これらのモジュールは、必要に応じて汎用の配線とLUTと共に1つかそれ以上のメモリプリミティブを用いることで構成されます。利用できるモジュールは以下の通りです。

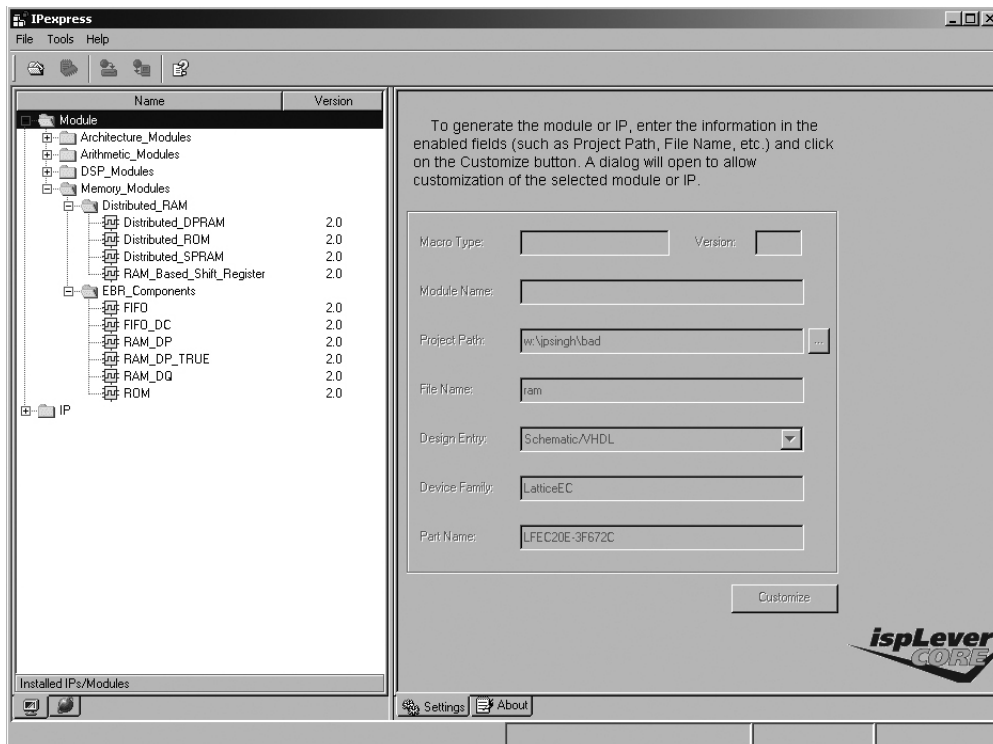
- ・シングルポートRAM (RAM_DQ) ; EBRベース
- ・デュアルポートRAM (RAM_DP_TRUE) ; EBRベース
- ・擬似デュアルポートRAM (RAM_DP) ; EBRベース
- ・リードオンリ・メモリ(ROM) ; EBRベース
- ・ファーストイン・ファーストアウト・メモリ(FIFO_DC) ; EBRベース
- ・分散シングルポートRAM (Distributed_SPRAM) ; PFUベース
- ・分散デュアルポートRAM (Distributed_DPRAM) ; PFUベース
- ・分散ROM (Distributed_ROM) ; PFU/PFFベース
- ・分散シフトレジスタ(RAM_Based_Shift_Register) ; PFUベース(IPexpressのヘルプを参照)

IPexpressフロー

これらのメモリのどれかを生成するには、まずLatticeECP2/Mデバイス用のプロジェクトを作成(オープン)します。

プロジェクト・ナビゲータから、**Tools > IPexpress**を選択します。或いは、LatticeECP2/Mデバイスがターゲットになっている場合、ツールバー内のボタンをクリックします。図11-2で示されるようにIPexpressウィンドウがオープンします。

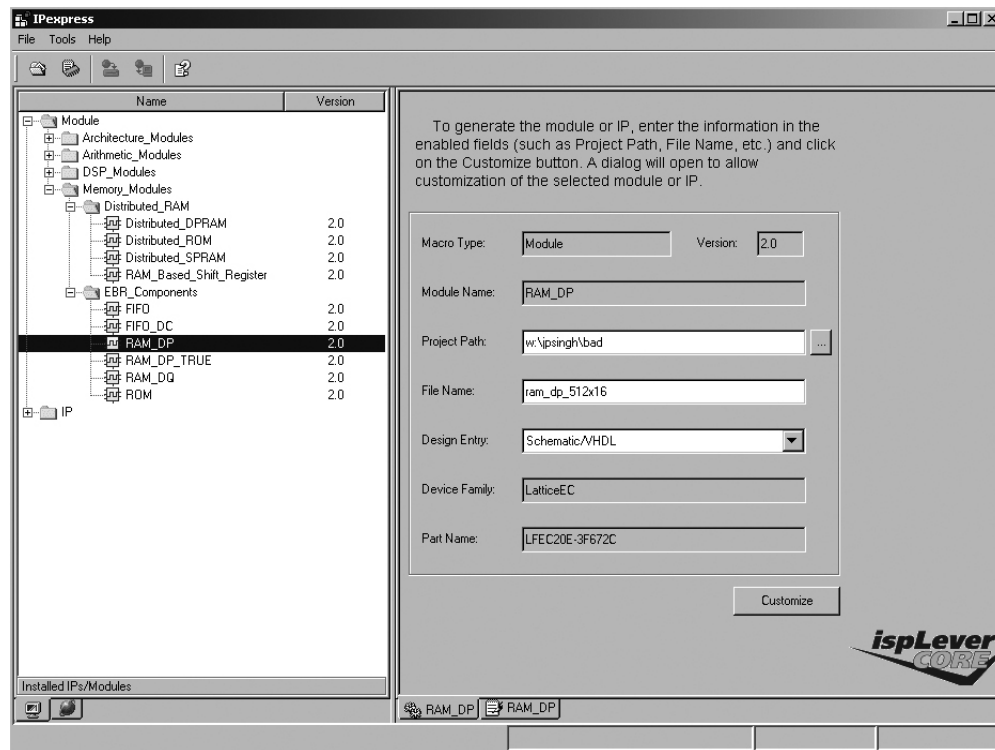
図 11-2 IPexpress メインウィンドウ



このウィンドウの左枠には、モジュールツリー(Module Tree)があります。図11-2に見られるように、EBRベースのメモリモジュールは**EBR_Components**の下に、PFUベースの分散メモリモジュールは**Storage_Components**の下にあります。

サイズ512 x 16のEBRベースの擬似デュアルポートRAM生成に関する例を見ます。まず**EBR_Components**下の**RAM_DP**を選択します。右枠は図11-3で示されるようになります。

図 11-3 IPexpress を用いることで擬似デュアルポートRAM(RAM_DP)を生成する例



ウィンドウの右枠には、デバイスファミリ (**Device Family**)、マクロタイプ (**Macro Type**)、カテゴリ (**Category**)、およびモジュール名 (**Module Name**) のような、選択されたデバイスとモジュールに依存するオプションがあります。IPexpress でこれらを変えることはできません。

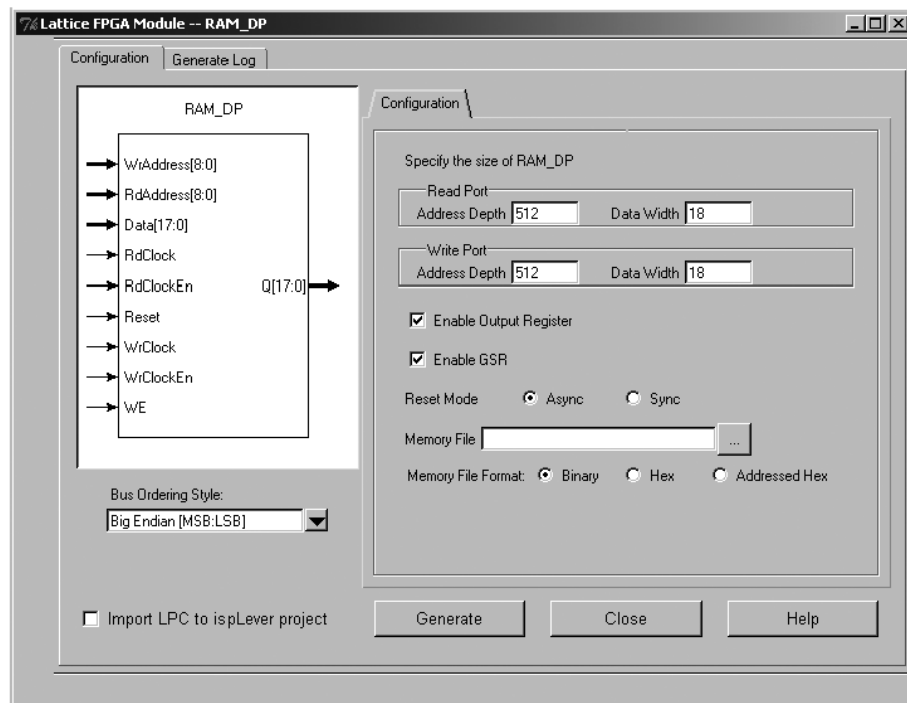
ユーザはプロジェクトパス (**Project Path**) ブラウズ (**Browse**) ボタンをクリックすることによって、生成するモジュールのファイルが置かれるディレクトリを変えることができます。

モジュール名 (**Module Name**) テキストボックスで、ユーザは生成しようとしているモジュールのエンティティ名を指定します。ユーザはこれを入力しなければなりません。

設計入力 (**Design Entry**, Verilog か VHDL) はデフォルトではプロジェクトタイプと同じです。プロジェクトが VHDL プロジェクトの場合、選択された設計入力オプションは "Schematic/VHDL" (日本語版註; 又は "VHDL" も可) で、プロジェクトタイプが Verilog-HDL では、"Schematic/Verilog-HDL" (日本語版註; 又は "Verilog HDL" も可) です。

デバイス (**Device**) プルダウンメニューでユーザは同一ファミリ内、この場合 LatticeECP2/M、で異なるデバイスを選択することが可能です。カスタマイズ (**Customize**) ボタンをクリックすることで、図 11-4 に示す別のウィンドウが開き、ここでユーザは RAM のカスタマイズができます。

図 11-4 擬似デュアルポートRAM(RAM_DP)モジュールのカスタマイズ例 ~ Configurationタブ



このウィンドウの左側はモジュールのブロック図を示します。右側にはコンフィグレーション (**Configuration**) タブがあり、RAM_DP をカスタマイズ (アドレス・ポートサイズとデータ幅を指定) するためのオプションを選択できます。

ユーザはテキストボックスに値を入力することで、リードポートとライトポートそれぞれのアドレスの深さとデータ幅を指定します。この例では、サイズ 512x16 の擬似デュアルポート RAM を生成しています。また、ユーザは擬似デュアルポートと真の (True) デュアルポート RAM の場合で異なるポート幅の RAM を作成することができます。

EBR ベースの RAM においてはクロックに同期した書き込みのみをサポートするため、データ入力とアドレス制御には常にレジスタがあります。チェックボックス **Enable Output Registers** は読み出しデータポートに出力レジスタを挿入します。EBR ベースの RAM で、出力レジスタはオプションです。

ユーザは非同期リセットか同期リセットかのリセットモード (**Reset Mode**) を選択できます。また GSR (Global Set Reset) を許可 (**Enable GSR**) するかどうかなを選択することができます。もし EBR がコンフィグレーション時に初期設定される場合、GSR 入力を無効にするか、或いはコンフィグレーション後のウェイクアップ・シーケンス中にデバイス I/O が解放される前に GSR がネゲートされなければなりません。これは全ての EBR RAM および ROM 実装に適用されます。EBR 同期リセットを用い、かつ EBR の GSR 入力が無効にされていれば、こうしたリセット要件はあてはまりません。

ユーザは **Memory File** で指定するメモリファイルを用意し、メモリを事前に (コンフィグレーション時に) 初期化することもできます。このファイルを RAM に用意するのはオプションですが、ROM の場合は必須です。ファイルはバイナリ、16 進 (Hex) または **Addresses Hex** 形式のいずれかです。このテクニカルノート の “初期設定ファイル” セクションでこれらの形式の詳細について議論します。

ここでユーザは、カスタマイズしたモジュールを生成するために **Generate** ボタンをクリックすることができます。Verilog か VHDL のネットリストが生成され、指定されたディレクトリに置かれます。ユーザはこのネットリストをその設計内に取り込むことができます。

別の重要なボタンは **Load Parameters** ボタンです。IPexpress は <module_name>.lpc ファイルの中で指定さ

れたパラメータを格納します。このファイルはモジュールと共に生成されます。ユーザはLoad Parameters ボタンをクリックすることで、以前に生成されたモジュールのパラメータをロードし、確認するか或いはその変更をすることができます。

モジュールがいったん生成されると、ユーザはその設計内で *.lpcかVerilog-HDL/VHDLファイルをトップレベル・モジュールでインスタンスすることができます。

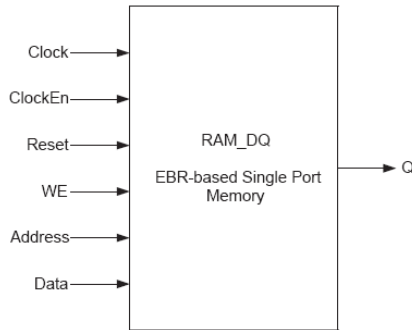
本テクニカルノートでは後に、EBRと分散メモリの種々メモリモジュールについて議論します。

メモリモジュール

シングルポートRAM (RAM_DQ) ～ EBRベース

シングルポートRAM (RAM_DQ) としてLatticeECP2/Mデバイス内のEBRブロックを構成することができます。IPexpressは、ユーザの設計要件に従ったメモリサイズ用に、EDIFネットリストと共にVerilog-HDLかVHDLを生成します。図11-5に生成されるメモリモジュールを示します。

図 11-5 IPexpressによって生成されたシングルポート・メモリモジュール



デバイスには多くのEBRブロックがあります。IPexpress GUI内でユーザによって指定されたメモリサイズを生成するために、これらEBRブロックやプリミティブを使用し連結します。1つのEBRブロックより小さいメモリでは、モジュールは1つのEBRブロックで作成されます。指定されたメモリが1つのEBRブロックより大きい場合では、そのサイズを作成するのに必要とされる深さと幅で、複数のEBRブロックを連結して生成されます。

シングルポートRAMモードで、ポートへの入力データとアドレスにはメモリアレイの入力にレジスタがあります。メモリの出力データへのレジスタはオプションです。

シングルポートメモリの様々なポートとそれらの定義は表11-2に含まれており、IPexpressによって生成されたモジュールとEBR RAM_DQプリミティブの対応するポートを記載します。

表 11-2 EBRベースのシングルポートメモリのポート定義

生成されたモジュール でのポート名	EBRブロック プリミティブのポート名	記述	アクティブなステート
Clock	CLK	クロック	立ち上がりエッジ
ClockEn	CE	クロック・イネーブル	Highアクティブ
Address	AD[x:0]	アドレスバス	—
Data	DI[y:0]	入力データ	—
Q	DO[y:0]	出力データ	—
WE	WE	ライト・イネーブル	Highアクティブ
Reset	RST	リセット	Highアクティブ
—	CS[2:0]	チップセレクト	—

リセット(RST)はRAMの入出力レジスタをリセットするだけで、メモリの内容をリセットしません。

EBRプリミティブにあるCS（チップセレクト）ポートは、メモリが複数のEBRブロックを連結される必要があるときに役立ちます。この場合3ビットのCS信号はアドレスのMSBとなり、8つのメモリまで容易に連結できます。ユーザによって指定されたメモリサイズが8つ以上のEBRブロック以上を必要とする場合、ソフトウェアは自動的にPFUで(EBRブロックの外部)で実装される追加アドレスデコード・ロジックを生成します。

それぞれのEBRブロックは18,432ビットのRAMより成り、それぞれのアドレス幅(x)とデータ幅(y)が表11-3に示されています。

表11-3 LatticeECP2/M デバイス、16Kメモリのシングルポートメモリ・サイズ

シングルポート メモリ・サイズ	入力データ	出力データ	アドレス[MSB:LSB]
16K x 1	DI	DO	AD[13:0]
8K x 2	DI[1:0]	DO[1:0]	AD[12:0]
4K x 4	DI[3:0]	DO[3:0]	AD[11:0]
2K x 9	DI[8:0]	DO[8:0]	AD[10:0]
1K x 18	DI[17:0]	DO[17:0]	AD[9:0]
512 x 36	DI[35:0]	DO[35:0]	AD[8:0]

表11-4はシングルポートメモリ(RAM_DQ)で利用できる様々なアトリビュートを示します。これらの幾つかはIPexpress GUIを通してユーザが選択可能です。詳細なアトリビュート定義については、付録Aを参照してください。

表11-4 LatticeECP2/M デバイスのシングルポートRAMアトリビュート

アトリビュート	記述	値	デフォルト値	IPexpressを通して ユーザ選択可能
Address depth	アドレス深さ	16K, 8K, 4K, 2N, 1K, 512		YES
Data Width	データ幅	1, 2, 4, 9, 18, 36	1	YES
Enable Output Registers	レジスタモード(パイプライン)	NOREG, OUTREG	NOREG	YES
Enable GSR	グローバル・セット/リセット	ENABLE, DISABLE	ENABLE	YES
Reset Mode	リセットタイプを選択	ASYN, SYNC	ASYN	YES
Memory File Format		BINARY, HEX, ADDRESSED HEX		YES
Write Mode	リード/ライト・モード	NORMAL, WRITETHROUGH	NORMAL	YES
Chip Select Decode	チップセレクト・デコード	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO
Init Value	初期化値	0x00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 0000000000000000...0xFFFFF FFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF	0x000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000000000 00000	NO

シングルポートRAM(RAM_DQ)を構成できるモードは、ノーマル(NORMAL)、またはライトスルー(WRITE THROUGH)のいずれかです。これらモードは、書き込み動作のすぐ後に読み出し動作が同じメモリアドレスに対して続いた場合、どのようなデータがメモリのポートQから出て来るかに影響します。

さらに、シングルポートRAMの出力レジスタをイネーブルするかどうかを選択することができます。図11-6から図11-9は、これらのオプションに対するシングルポートRAMの内部タイミング波形をそれぞれ示しています。

図11-6 シングルポートRAMタイミング波形 ~ NORMAL モード、出力レジスタなし

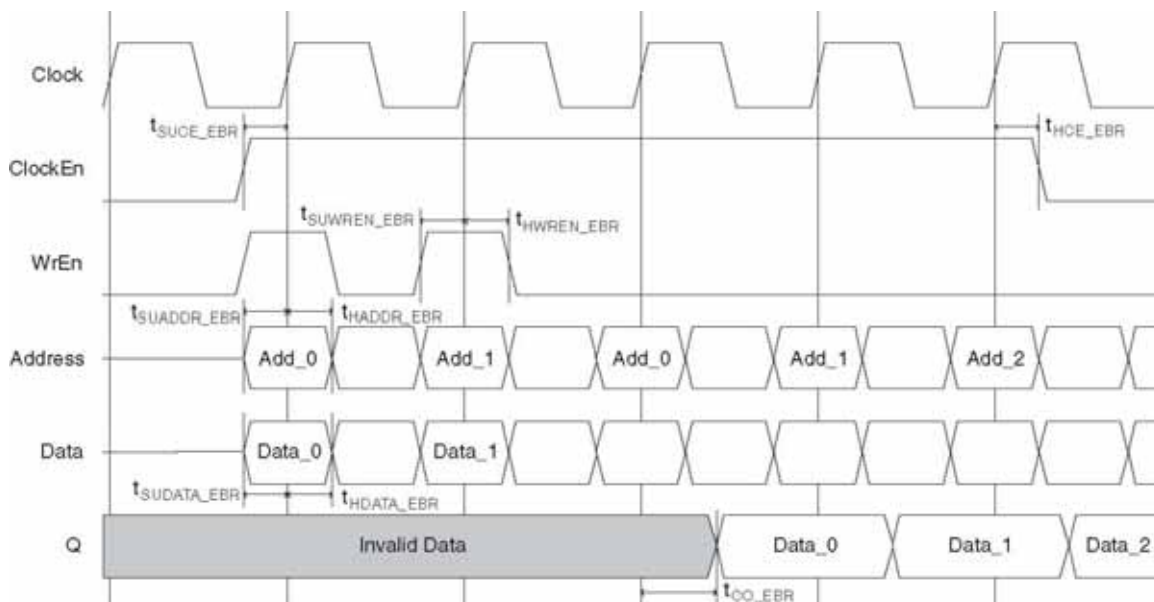


図11-7 シングルポートRAMタイミング波形 ~ NORMAL モード、出力レジスタあり

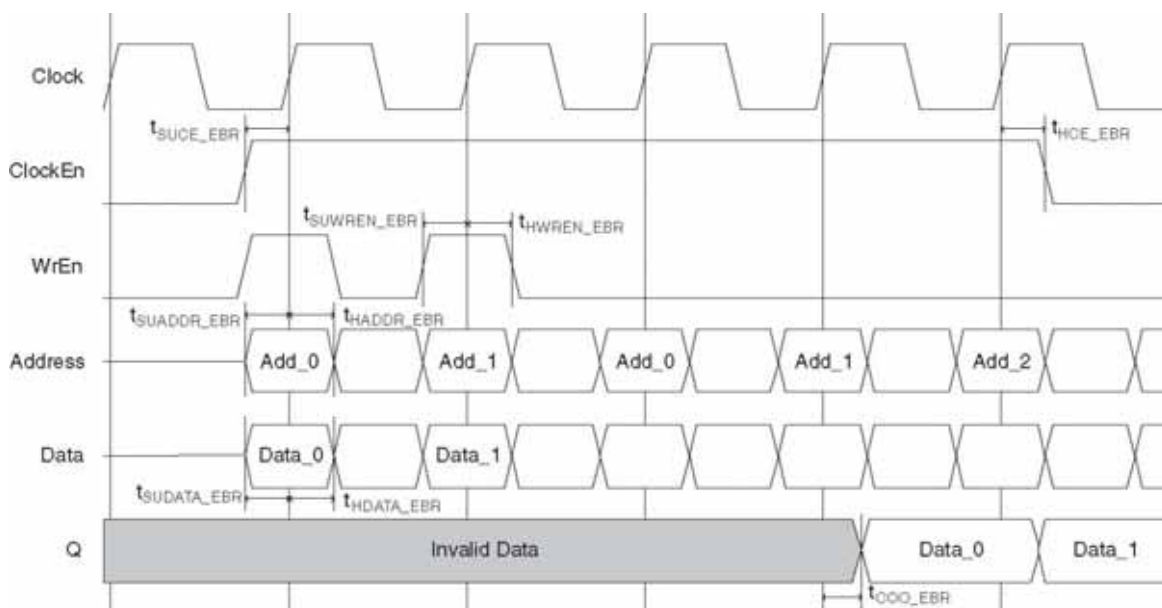
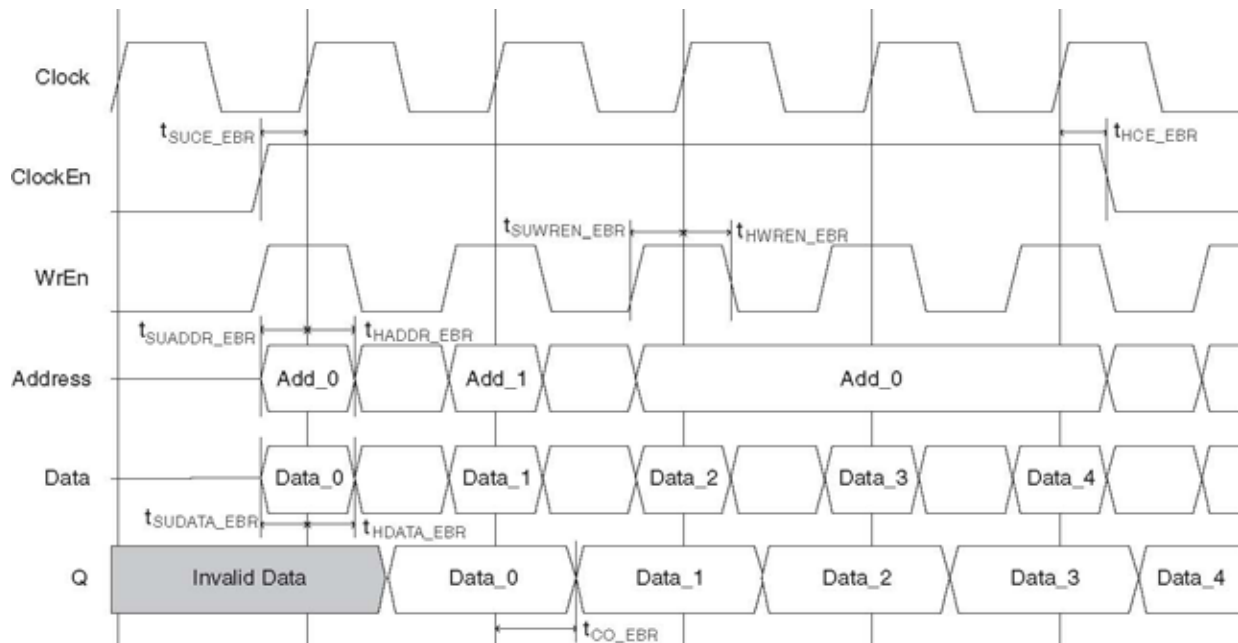
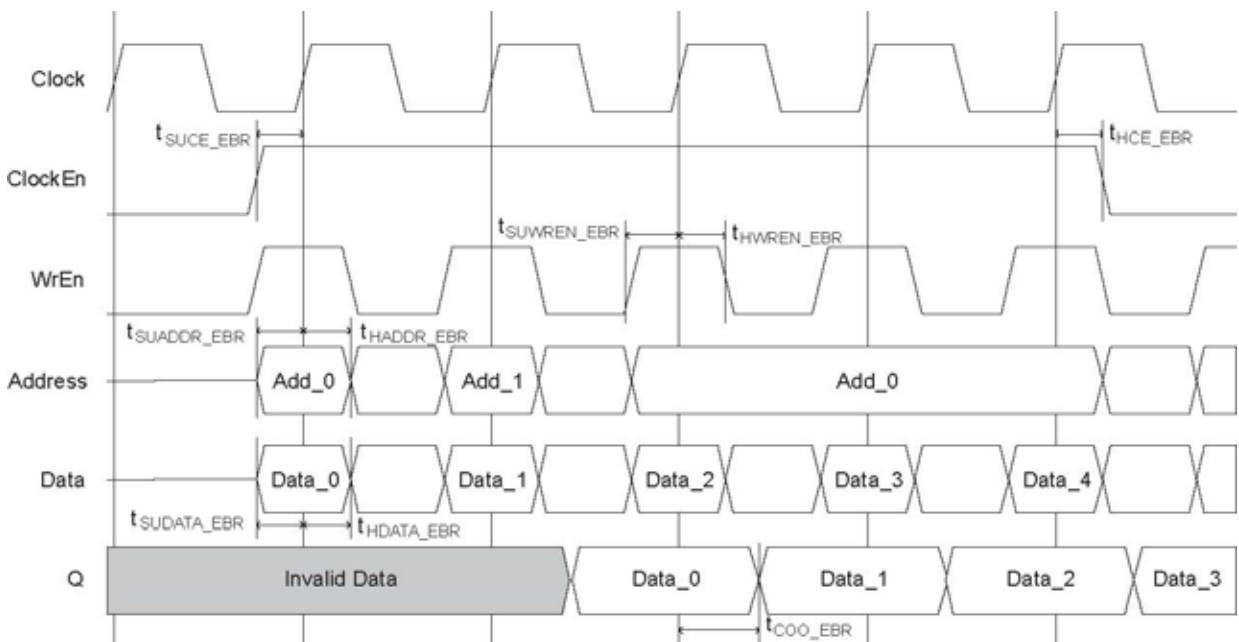
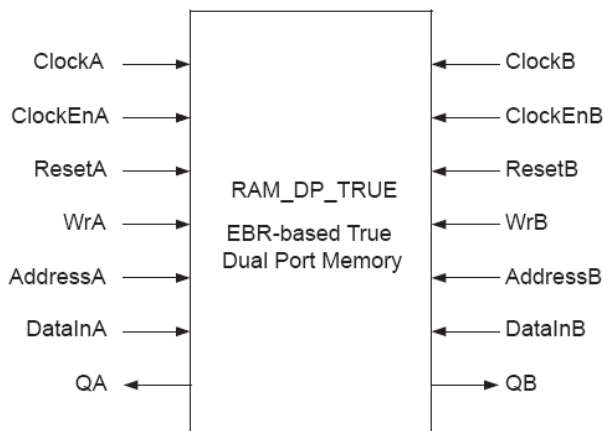


図11-8 シングルポートRAMタイミング波形 ~ **WRITE THROUGH** モード、出力レジスタなし図11-9 シングルポートRAMタイミング波形 ~ **WRITE THROUGH** モード、出力レジスタあり

真の(True)デュアルポートRAM (RAM_DP_TRUE) ; EBRベース

True-Dual Port RAM (RAM_DP_TRUE) としてLatticeECP2/MデバイスにおけるEBRブロックを構成することができます。IPexpressは、ユーザの設計要件に従ったメモリサイズ用に、EDIFネットリストと共にVerilog-HDLかVHDLを生成します。図11-10に生成されるメモリモジュールを示します。

図11-10 IPexpressで生成した真のデュアルポート・メモリモジュール



生成されたモジュールはRAM_DP_TRUEプリミティブを利用します。1つのEBRブロックより小さいメモリでは、モジュールは1つのEBRブロックで作成されます。指定されたメモリが1つのEBRブロックより大きい場合は、そのサイズを作成するのに必要とされる深さと幅で、複数のEBRブロックを連結して生成されます。

真のデュアルポートRAMモードで、入力データとアドレスポートにはメモリアレイの入力レジスタがあります。メモリの出力データへのレジスタはオプションです。

真のデュアルポート・メモリの種々ポートとそれらの定義は表11-5に示されています。IPexpressによって生成されたモジュールとEBR RAM_DP_TRUEプリミティブの対応するポートを記載します。

表11-5 EBRベースの真のデュアルポート・メモリのポート定義

生成されたモジュールのポート名	EBRブロック・プリミティブのポート名	記述	アクティブなステート
ClockA, ClockB	CLKA, CLKB	ポートA/ポートBのクロック	立ち上がりエッジ
ClockEnA, ClockEnB	CEA, CEB	ポートA/ポートBのクロック・イネーブル	Highアクティブ
AddressA, AddressB	ADA[18:19-x1], ADB[18:19-x2]	ポートA/ポートBのアドレスバス	—
DataA, DataB	DIA[y1:0], DIB[y2:0]	ポートA/ポートBの入力データ	—
QA, QB	DOA[y1:0], DOB[y2:0]	ポートA/ポートBの出力データ	—
WEA, WEB	WEA, WEB	ポートA/ポートBのライトイネーブル	Highアクティブ
ResetA, ResetB	RSTA, RSTB	ポートA/ポートBのリセット	Highアクティブ
—	CSA[2:0], CSB[2:0]	各ポートのチップセレクト	—

リセット(RST)はRAMの入出力レジスタをリセットするだけで、メモリの内容をリセットしません。

EBRプリミティブにあるCS（チップセレクト）ポートは、メモリが複数のEBRブロックを連結される必要があるときに役立ちます。この場合3ビットのCS信号はアドレスのMSBとなり、8つのメモリまで容易に連結できます。ユーザによって指定されたメモリ・サイズが8つ以上のEBRブロック以上を必要とする場合、ソフトウェアは自動的にPFUで(EBRブロックの外部)で実装される追加アドレス・デコード・ロジックを生成します。

それぞれのEBRブロックは18,432ビットのRAMより成り、それぞれのアドレス幅(x)とデータ幅(y)が表11-6に示されています

表11-6 LatticeECP2/M デバイス、16Kメモリの真のデュアルポート・メモリ・サイズ

デュアルポート メモリサイズ	入力データ ポートA	入力データ ポートB	出力データ ポートA	出力データ ポート B	アドレス・ポート A [MSB:LSB]	アドレス・ポート B [MSB:LSB]
16K x 1	DIA	DIB	DOA	DOB	ADA[13:0]	ADB[13:0]
8K x 2	DIA[1:0]	DIB[1:0]	DOA[1:0]	DOB[1:0]	ADA[12:0]	ADB[12:0]
4K x 4	DIA[3:0]	DIB[3:0]	DOA[3:0]	DOB[3:0]	ADA[11:0]	ADB[11:0]
2K x 9	DIA[8:0]	DIB[8:0]	DOA[8:0]	DOB[8:0]	ADA[10:0]	ADB[10:0]
1K x 18	DIA[17:0]	DIB[17:0]	DOA[17:0]	DOB[17:0]	ADA[9:0]	ADB[9:0]

表11-7は真のデュアルポートMemory(RAM_DP_TRUE)で利用できる様々なアトリビュートを示します。これらのいくつかはIPexpress GUIを通してユーザの選択が可能です。詳細なアトリビュート定義については、付録Aを参照してください。

表11-7 LatticeECP2/M用の真のデュアルポートRAMアトリビュート

[illegible]

真のデュアルポートRAM (True Dual Port RAM ; RAM_DP_TRUE) を構成できるモードも、ノーマル、またはライトスルーのいずれかです。これらモードはいずれも、書き込み動作のすぐ後に読み出し動作が同じメモリアドレスに対して続いた場合、どのようなデータがメモリのポートQから出て来るかに影響します。WRITEモードと真のデュアルポートの制約についてはAppendix Aにより詳しく記載されています。

さらに、真のデュアルポートRAMの出力レジスタをイネーブルするかどうかを選択することができます。図11-11から図11-14はこれらのオプションに対する真のデュアルポートRAMの内部タイミング波形をそれぞれ示しています。

図11-11 真のデュアルポートRAMタイミング波形 ～ NORMAL モード、出力レジスタなし

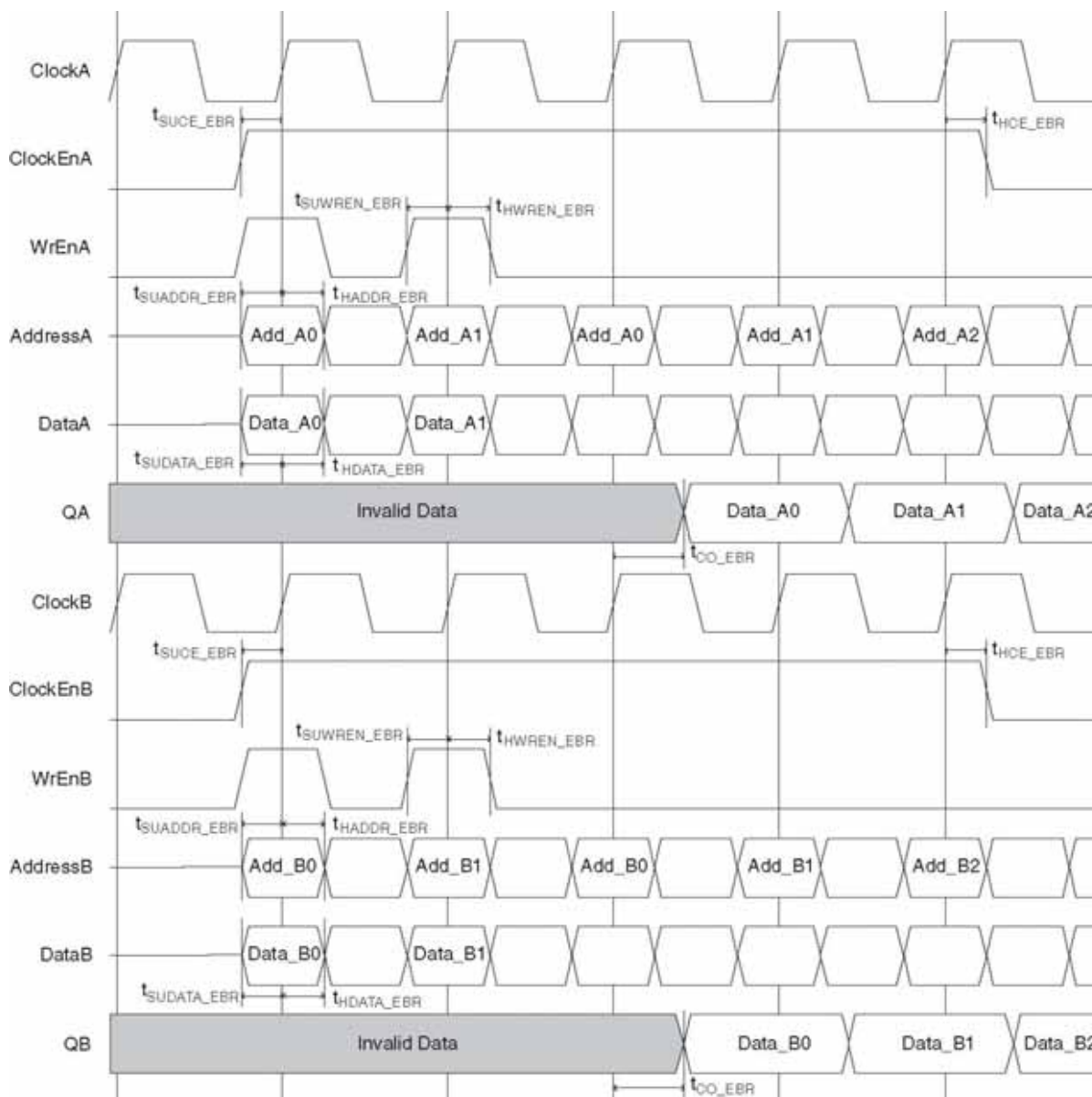


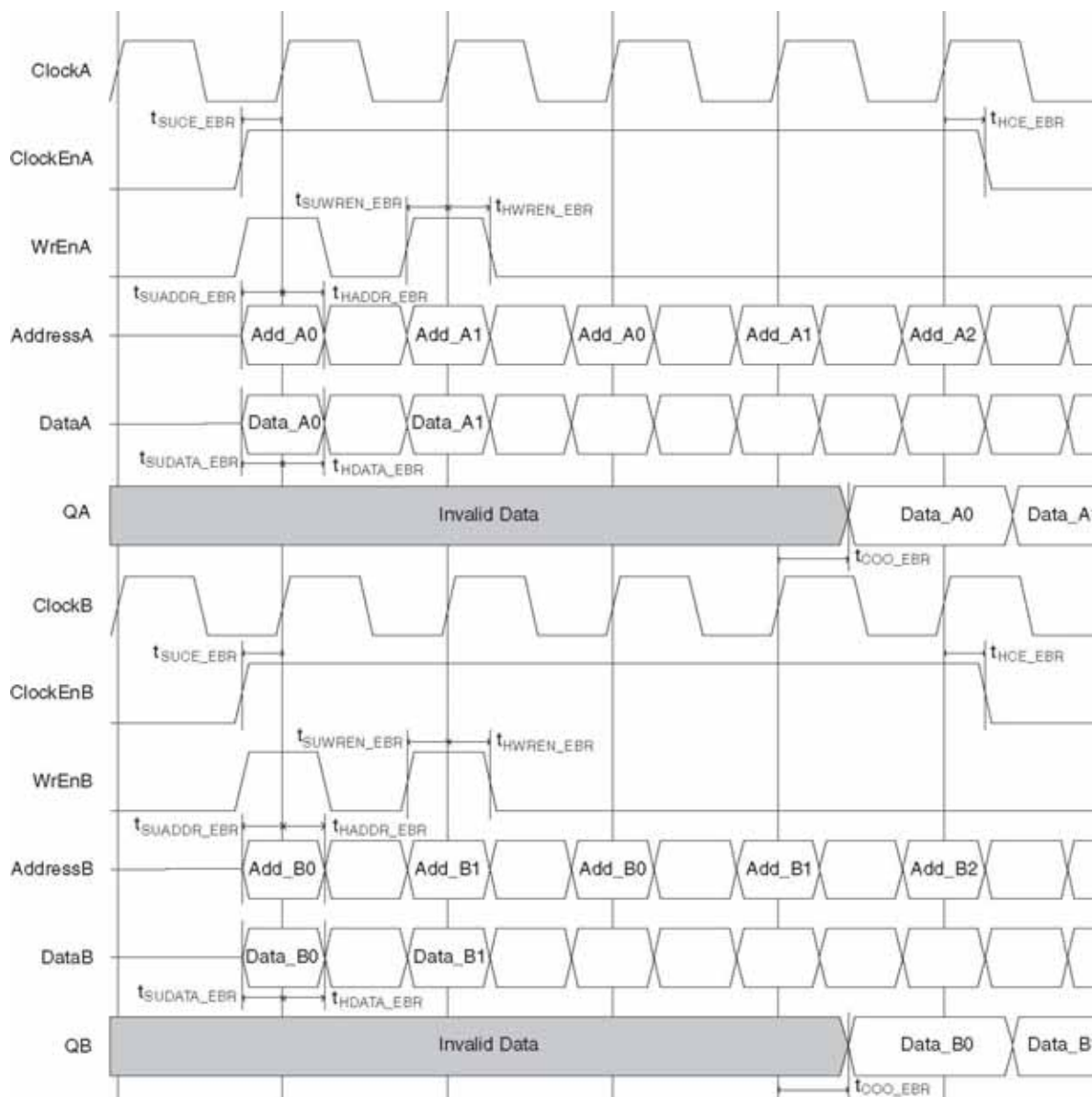
図11-12 真のデュアルポートRAMタイミング波形 ～ **NORMAL** モード、出力レジスタあり

図11-13 真のデュアルポートRAMタイミング波形 ～ WRITE THROUGH モード、出力レジスタなし

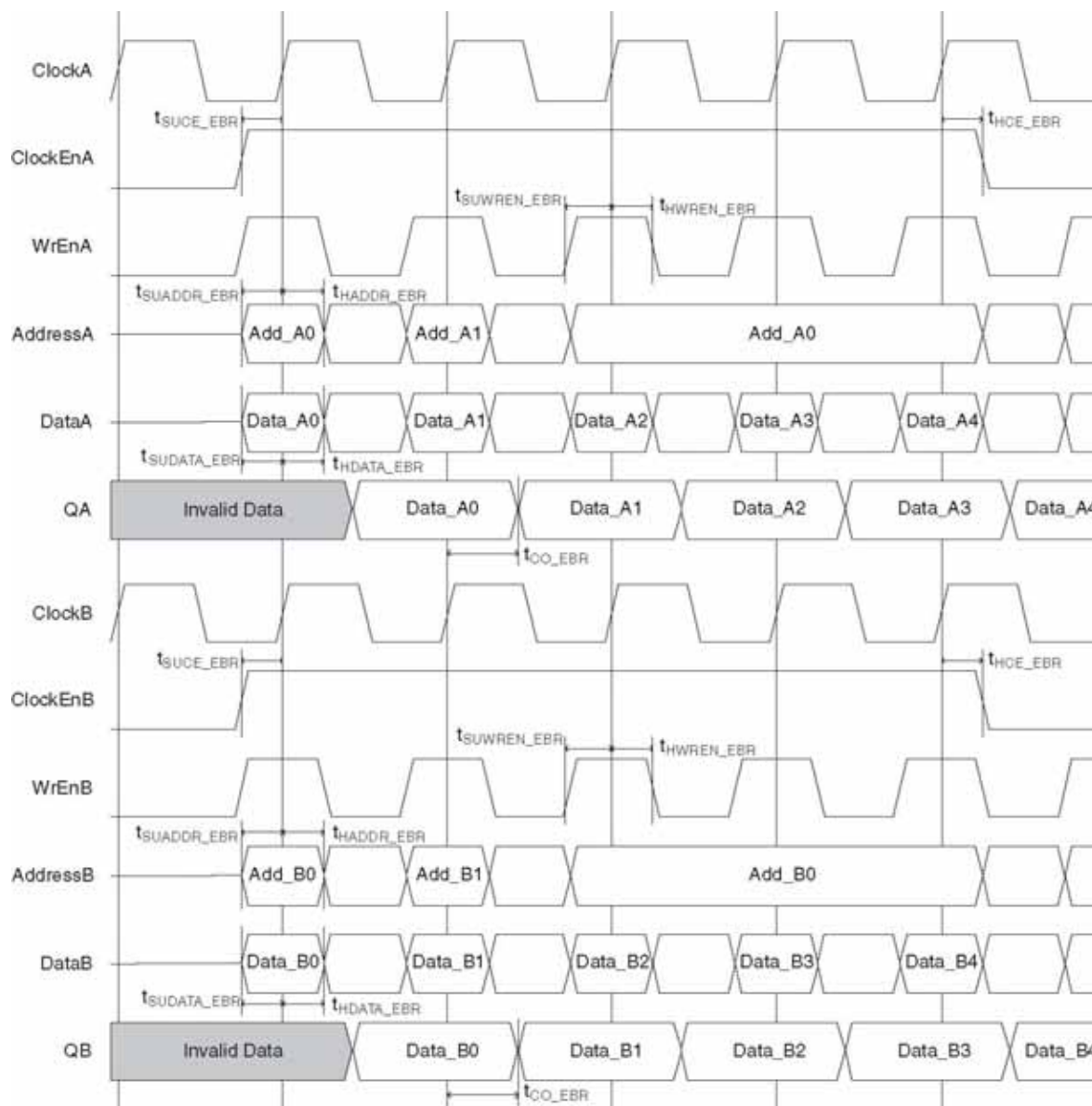
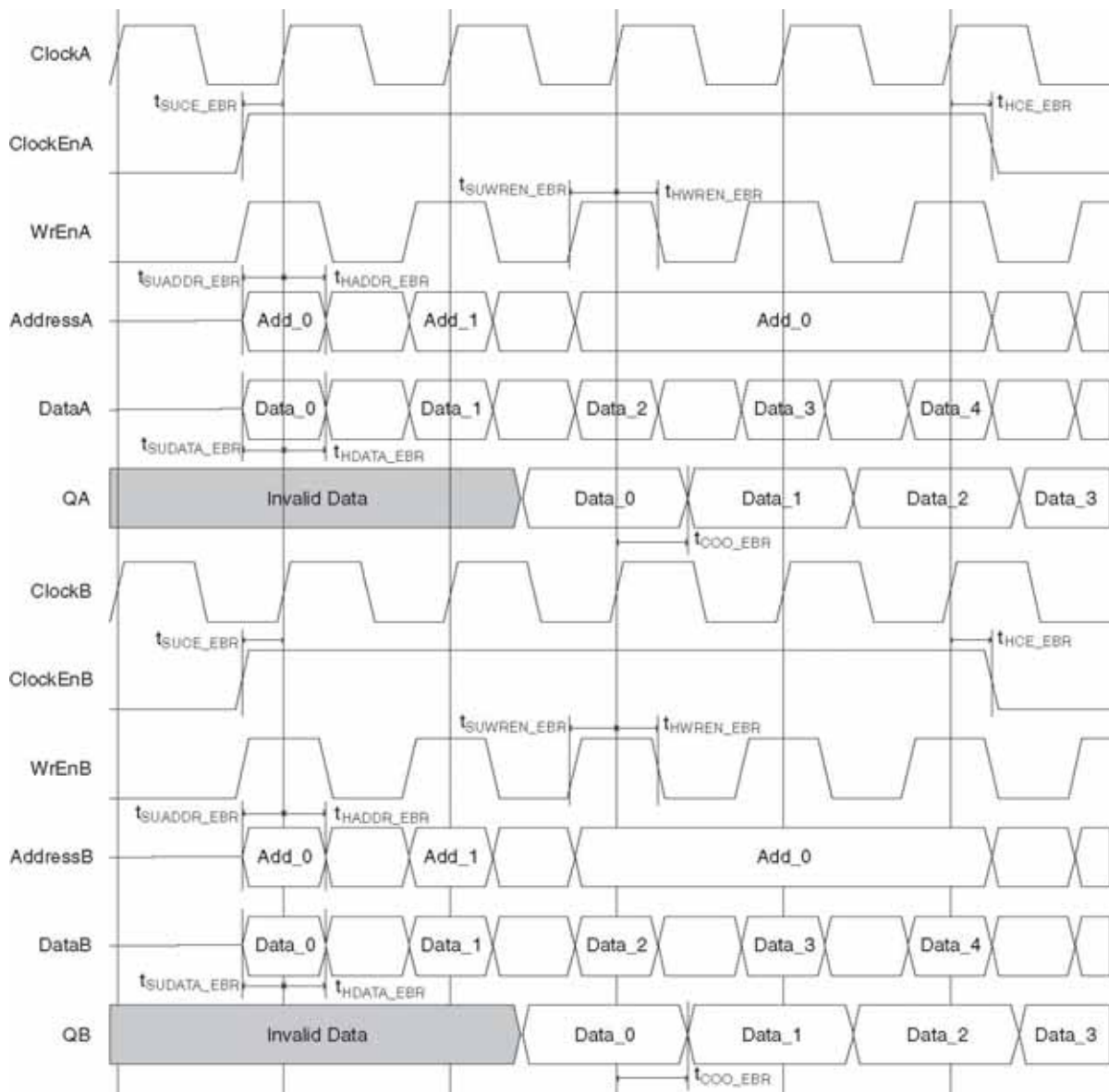


図11-14 真のデュアルポートRAMタイミング波形 ～ WRITE THROUGH モード、出力レジスタあり

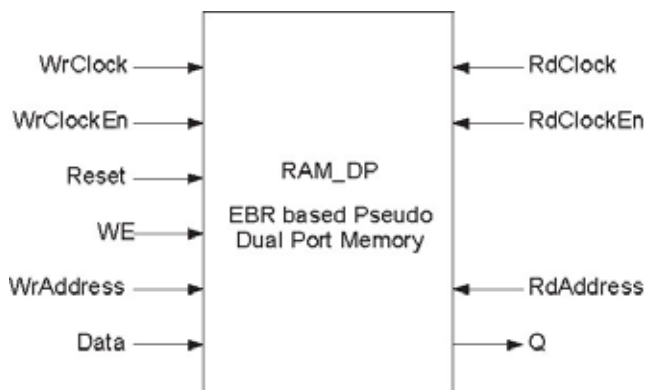


擬似デュアルポートRAM (RAM_DP) ～ EBRベース

LatticeECP2/Mデバイス内のEBRブロックを擬似デュアルポートRAMかRAM_DPとして構成することができます。IPexpressは、ユーザの設計要件に従ったメモリ・サイズ用に、EDIFネットリストと共にVerilog-HDLかVHDLを生成します。

図11-15に生成されるメモリモジュールを示します。

図11-15 IPexpress で生成した擬似デュアルポート・メモリモジュール



生成されたモジュールはこれらEBRブロックかプリミティブを利用します。1つのEBRブロックより小さいメモリでは、モジュールは1つのEBRブロックで作成されます。指定されたメモリが1つのEBRブロックより大きい場合では、そのサイズを作成するのに必要とされる深さと幅で、複数のEBRブロックを連結して生成されます。

擬似デュアルポートRAMモードで、入力データとアドレスポートにはメモリアレイの入力レジスタがあります。メモリの出力データへのレジスタはオプションです。

擬似デュアルポート・メモリの様々なポートとそれらの定義は表11-8に含まれています。表はモジュール・マネージャによって生成されたモジュールとEBR RAM_DPプリミティブの対応するポートを記載します。

表11-8 EBRベース擬似デュアルポート・メモリのポート定義

生成されたモジュールのポート名	EBRブロック・プリミティブのポート名	記述	アクティブなステート
RdAddress	ADR[x1:0]	リードアドレス	—
WrAddress	ADW[x2:0]	ライトアドレス	—
RdClock	CLKR	リードクロック	立ち上がりエッジ
WrClock	CLKW	ライトクロック	立ち上がりエッジ
RdClockEn	CER	リードクロック・イネーブル	Highアクティブ
WrClockEn	CEW	ライトクロック・イネーブル	Highアクティブ
Q	DO[y1:0]	リードデータ	—
Data	DI[y2:0]	ライトデータ	—
WE	WE	ライトイネーブル	Highアクティブ
Reset	RST	リセット	Highアクティブ
—	CS[2:0]	チップセレクト	—

リセット(RST)はRAMの入出力レジスタをリセットするだけで、メモリの内容をリセットしません。EBRプリミティブにあるCS（チップセレクト）ポートは、メモリが複数のEBRブロックを連結される必要があるときに役立ちます。この場合3ビットのCS信号はアドレスのMSBとなり、8つのメモリまで容易に連結できます。ユーザによって指定されたメモリ・サイズが8つ以上のEBRブロック以上を必要とする場合、ソフトウェアは自動的にPFUで(EBRブロックの外部)で実装される追加アドレス・デコード・ロジックを生成します。

それぞれのEBRブロックは18,432ビットのRAMより成り、それぞれのアドレス幅(x)とデータ幅(y)が表11-9に示されています。

表11-9 LatticeECP2/M デバイス、16Kメモリの擬似デュアルポート・メモリ・サイズ

擬似デュアルポート・メモリサイズ	入力データポート A	入力データポート B	出力データポート A	出力データポート B	リード・アドレス・ポート A [MSB:LSB]	ライト・アドレス・ポート B [MSB:LSB]
16K x 1	DIA	DIB	DOA	DOB	RAD[13:0]	WAD[13:0]
8K x 2	DIA[1:0]	DIB[1:0]	DOA[1:0]	DOB[1:0]	RAD[12:0]	WAD[12:0]
4K x 4	DIA[3:0]	DIB[3:0]	DOA[3:0]	DOB[3:0]	RAD[11:0]	WAD[11:0]
2K x 9	DIA[8:0]	DIB[8:0]	DOA[8:0]	DOB[8:0]	RAD[10:0]	WAD[10:0]
1K x 18	DIA[17:0]	DIB[17:0]	DOA[17:0]	DOB[17:0]	RAD[9:0]	WAD[9:0]
512 x 36	DIA[35:0]	DIB[35:0]	DOA[35:0]	DOB[35:0]	RAD[8:0]	WAD[8:0]

表11-10は擬似デュアルポート・メモリ(RAM_DP)で利用できる様々なアトリビュートを示します。これらのいくつかはIPexpress GUIを通してユーザの選択が可能です。詳細なアトリビュート定義について、付録Aを参照してください。

表11-10 LatticeECP2/M デバイスの擬似デュアルポートRAMアトリビュート

アトリビュート	記述	値	デフォルト値	IPexpressによってユーザが選択可能
Read Port Address depth	リードポートのアドレス深さ	16K, 8K, 4K, 2K, 1K		YES
Read Port Data Width	リードポートのデータ幅	1, 2, 4, 9, 18	1	YES
Write Port Address depth	ライトポートのアドレス深さ	16K, 8K, 4K, 2K, 1K		YES
Write Port Data Width	ライトポートのデータワード幅	1, 2, 4, 9, 18	1	YES
Write Port Enable Output Register	ライトポートのレジスタモード(パイプライン)	NOREG, OUTREG	NOREG	YES
Enable GSR	グローバルセット・リセットのイネーブル	ENABLE, DISABLE	ENABLE	YES
Reset Mode	リセットタイプの選択	ASYNC, SYNC	ASYNC	YES
Memory File Format		BINARY, HEX, ADDRESSED HEX		YES
Read Port Write Mode	リードポートのRead / Writeモード	NORMAL	NORMAL	YES
Write Port Write Mode	ライトポートのRead / Writeモード	NORMAL	NORMAL	YES
Chip Select Decode for Read Port	リードポートのチップセレクト・デコード	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO
Chip Select Decode for Write Port	ライトポートのチップセレクト・デコード	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO
Init Value	初期化値	0x00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000...0xFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	0x00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000	NO

擬似デュアルポートRAM (RAM_DP) の出力レジスタをイネーブルするかどうか選択できます。図11-16から図11-17はこれらのオプションに対する擬似デュアルポートRAMの内部タイミング波形をそれぞれ示しています。

図11-16 擬似デュアルポートRAMタイミング波形 ～ 出力レジスタなし

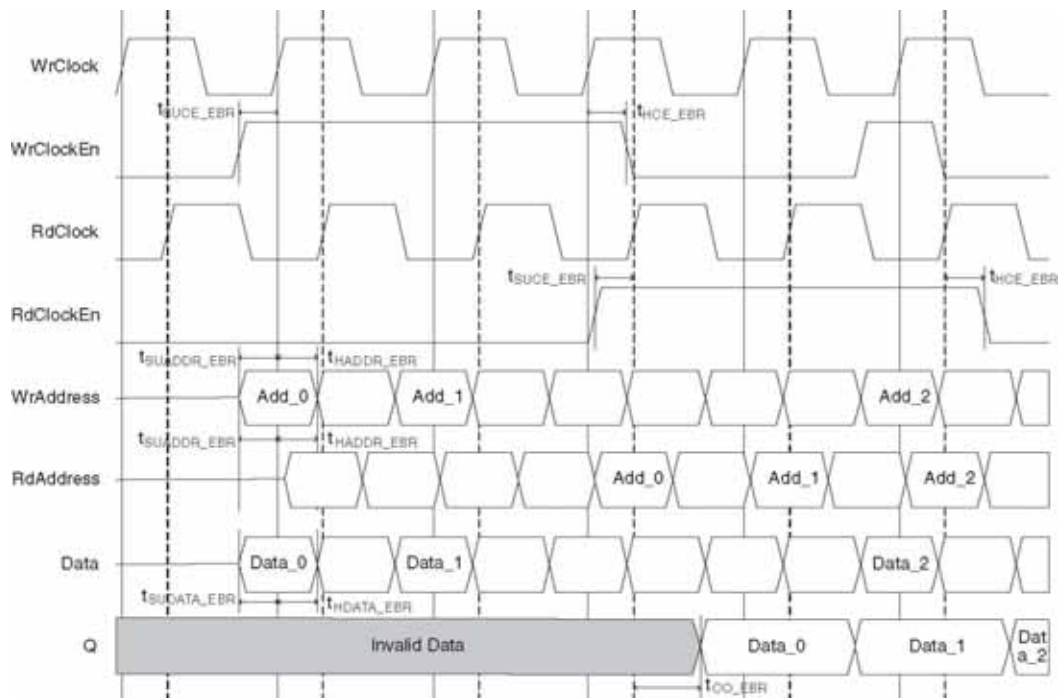
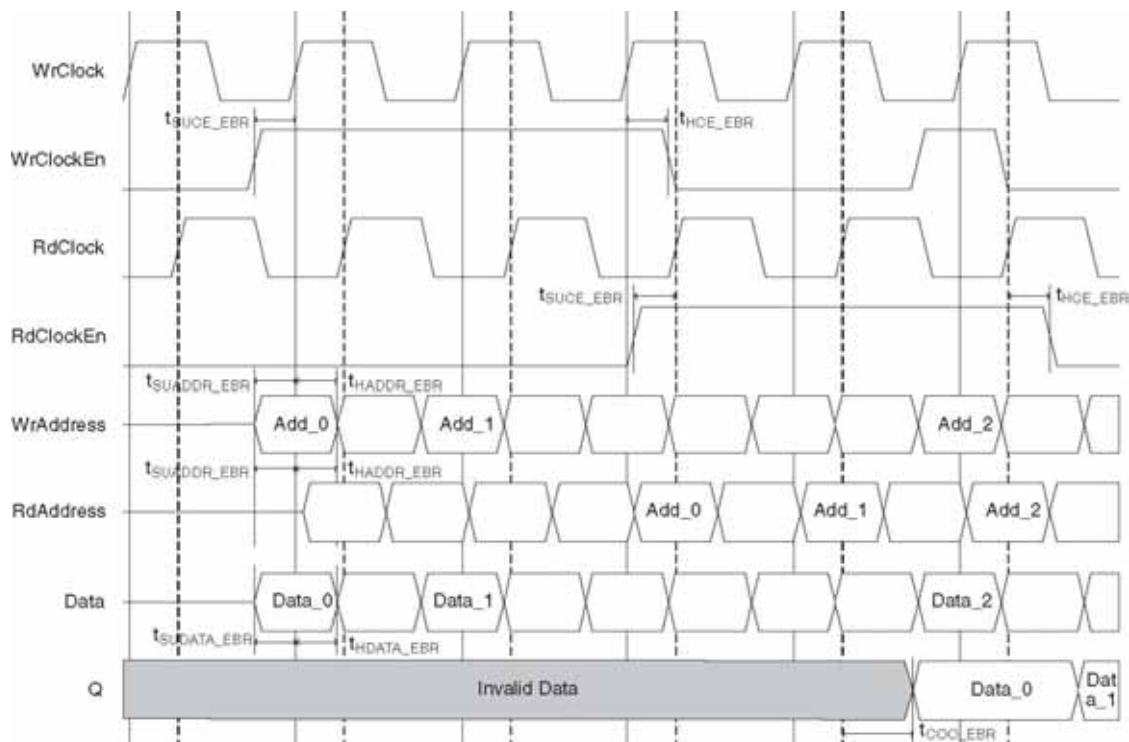


図11-17 擬似デュアルポートRAMタイミング波形 ～ 出力レジスタあり

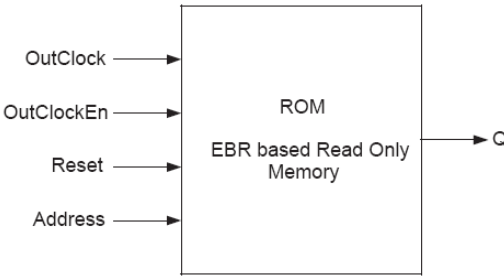


リードオンリ・メモリ(ROM)；EBRベース

LatticeECP2/MデバイスのEBRブロックをリードオンリ・メモリ（ROM）として構成することができます。IPexpressは、ユーザが設計要件に従ったメモリ・サイズ用に、EDIFネットリストと共にVerilog-HDLかVHDLを生成することを可能にします。ユーザはROM内容を初期化ファイルの所定フォーマットで用意しなければなりません。

IPexpressは図11-18で示されるようなメモリモジュールを生成します。

図11-18 IPexpressによって生成されたROM；リードオンリ・メモリ・モジュール



生成されたモジュールはこれらのEBRブロックかプリミティブを利用します。1つのEBRブロックより小さいメモリでは、モジュールは1つのEBRブロックで作成されます。指定されたメモリが1つのEBRブロックより大きい場合では、そのサイズを作成するのに必要とされる深さと幅で、複数のEBRブロックを連結して生成されます。

ROMモードで、アドレスポートはメモリアレイの入力レジスタがあります。メモリの出力データへのレジスタはオプションです。

ROMの様々なポートとそれらの定義は表11-11に示されます。表はIPexpressによって生成されたモジュールとROMプリミティブの対応するポートを記載します。

表11-11 EBRベースのROMポート定義

生成されたモジュールのポート名	EBRブロック・プリミティブでのポート名	記述	アクティブなステート
Address	AD[x:0]	リード・アドレス	—
OutClock	CLK	クロック	立ち上がりエッジ
OutClockEn	CE	クロック・イネーブル	Highアクティブ
Reset	RST	リセット	Highアクティブ
—	CS[2:0]	チップセレクト	—

リセット(RST)はRAMの入出力レジスタをリセットするだけで、メモリの内容をリセットしません。

EBRプリミティブにあるCS（チップセレクト）ポートは、メモリが複数のEBRブロックを連結される必要があるときに役立ちます。この場合3ビットのCS信号はアドレスのMSBとなり、8つのメモリまで容易に連結できます。ユーザによって指定されたメモリ・サイズが8つ以上のEBRブロック以上を必要とする場合、ソフトウェアは自動的にPFUで(EBRブロックへの外部)で実装される追加アドレス・デコード・ロジックを生成します。

IPexpressを用いてROMを生成する際に、ユーザはROMの内容を事前に初期化するために初期化ファイルを用意しなければなりません。これらのファイルは *.mem（拡張子）で、バイナリ、16進(Hex)またはAddresses Hex (ORCA)形式のいずれかです。初期化ファイルについては、このテクニカルノートの“メモリの初期化”セクションで詳細に議論します。

ユーザはリードオンリ・メモリ(ROM)の出力レジスタをイネーブルするかどうか選択できます。図11-23から図11-24はこれらのオプションに対するROMの内部タイミング波形をそれぞれ示しています。

それぞれのEBRブロックは18,432ビットのRAMから成ります。各EBRブロックのアドレス値(x)とデータ値(y)の組み合わせは表11-12の通りです。

表11-12 LatticeECP2/Mの16KメモリのROMサイズ

ROM	出力データ	アドレスポート {MSB:LSB}
16K x 1	DOA	WAD[13:0]
8K x 2	DOA[1:0]	WAD[12:0]
4K x 4	DOA[3:0]	WAD[11:0]
2K x 9	DOA[8:0]	WAD[10:0]
1K x 18	DOA[17:0]	WAD[9:0]
512 x 36	DOA[35:0]	WAD[8:0]

表11-13はリードオンリ・メモリ(ROM)で利用できる種々アトリビュートを示します。これらのアトリビュートのいくつかはIPexpress GUIを通してユーザが選択可能です。詳細なアトリビュート定義についてはAppendix Aを参照してください。

表11-13 LatticeECP2/MのROMアトリビュート定義

アトリビュート	記述	値	デフォルト値	IPexpressを通して 選択ユーザが可能
Address Depth	アドレス深さ	16K, 8K, 4K, 2K, 1K, 512		YES
Data Width	データ幅	1, 2, 4, 9, 18, 36	1	YES
Enable Output Register	レジスタモード(パイプライン)	NOREG, OUTREG	NOREG	YES
Enable GSR	グローバルセット・リセットのイネーブル	ENABLE, DISABLE	ENABLE	YES
Reset Mode	リセットタイプの選択	ASYNCR, SYNC	ASYNCR	YES
Memory File Format		BINARY, HEX, ADDRESSSED HEX		YES
Chip Select Decode	チップセレクト・デコード	0b000, 0b001, 0b010, 0b011, 0b100, 0b101, 0b110, 0b111	0b000	NO

図11-19 ROMタイミング波形 ～ 出力レジスタなし

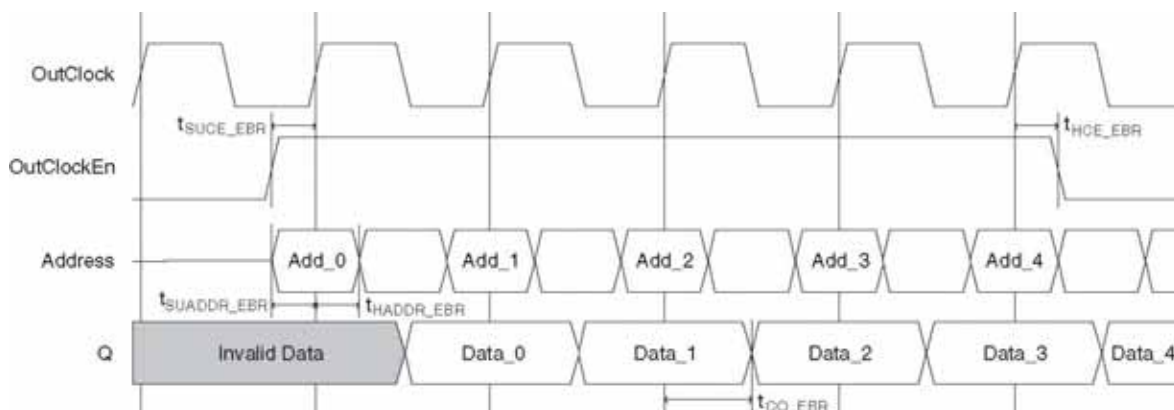
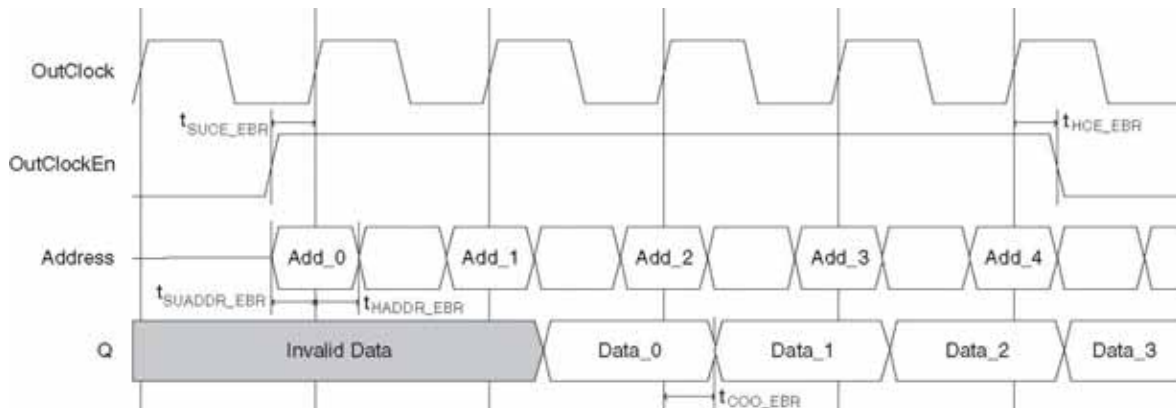


図11-20 ROMタイミング波形 ～ 出力レジスタあり



ファーストイン・ファーストアウト(FIFO, FIFO_DC) ～ EBRベース

ハードウェアには、シングルポート(RAM_DQ)、擬似デュアルポート(RAM_DP)、及び(真の)デュアルポート(RAM_DP_TRUE) RAMに構成することができる組み込みブロックRAM (EBR)があります。ユーザは擬似デュアルポートRAMを用いて、ispLEVERデザイン・ソフトウェアのIPexpressポイントツールを活用して、FIFOとFIFO_DCを構築できます。

これらFIFOは出力レジスタあり(パイプライン)、又はなし(非パイプライン)としてそれぞれ構成することができます。パイプライン・モードでは、ユーザにはRdEn信号で出力レジスタをイネーブルする付加的なオプションがあります。以下のセクションでFIFOの動作について議論します。

ファーストイン・ファーストアウト(FIFO) メモリ

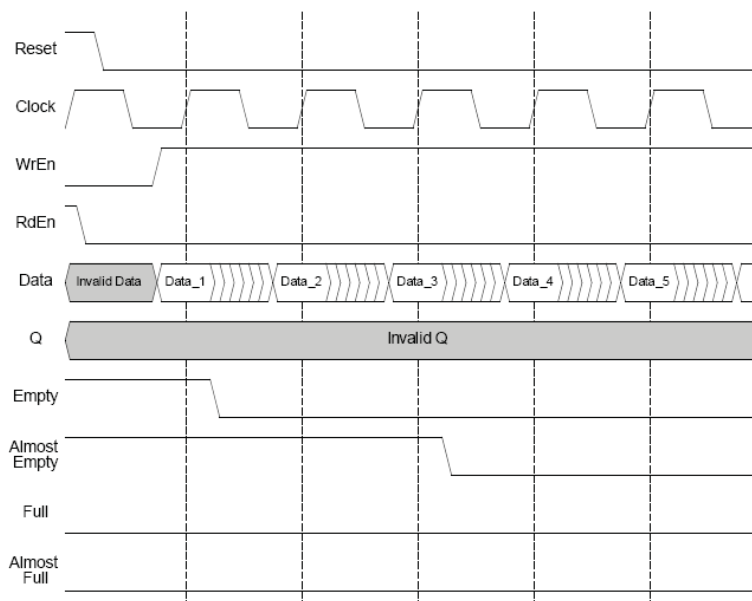
FIFO (または単一クロックFIFO) はエミュレートFIFOです。アドレスとフラグのロジック回路はRAM周りのFPGAファブリックに実装されます。

FIFOで利用できるポートは以下の通りです。

- Reset
- Clock
- WrEn
- RdEn
- Data
- Q
- Full フラグ
- Almost Full フラグ
- Empty フラグ
- Almost Empty フラグ

まず最初に、非パイプラインの(出力レジスタのない) FIFOについて議論します。図11-21は、いつ空(empty)の状態、またいつデータが書き始められるかというFIFO動作を示します。

図11-21 出力レジスタのないFIFO、データライト・サイクルの初め

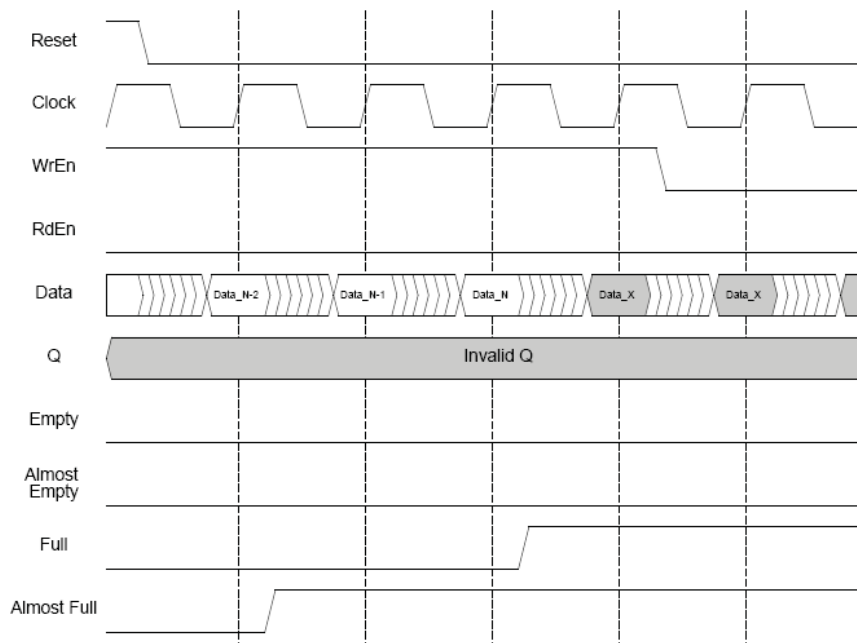


FIFOに書き始めるためにはWrEn信号はHighでなければなりません。初めはEmptyとAlmost EmptyフラグがHighで、またFullとAlmost FullはLowです。

最初のデータがFIFOに書かれると、FIFOはもう空ではないのでEmptyフラグはネゲートされ(Lowになり)ます。この図では、Almost Emptyのフラグ設定が3 (アドレス位置の3)であると想定しています。したがって、3番目のアドレス位置が満たされると、Almost Emptyフラグはネゲートされます。

今、FIFOを満たすために書き続けると仮定します。FIFOが満たされると、Almost FullとFullの両フラグがアサートされます。図11-22はこれらのフラグの動きを示します。この図ではFIFOの深さが 'N' であると想定しています。

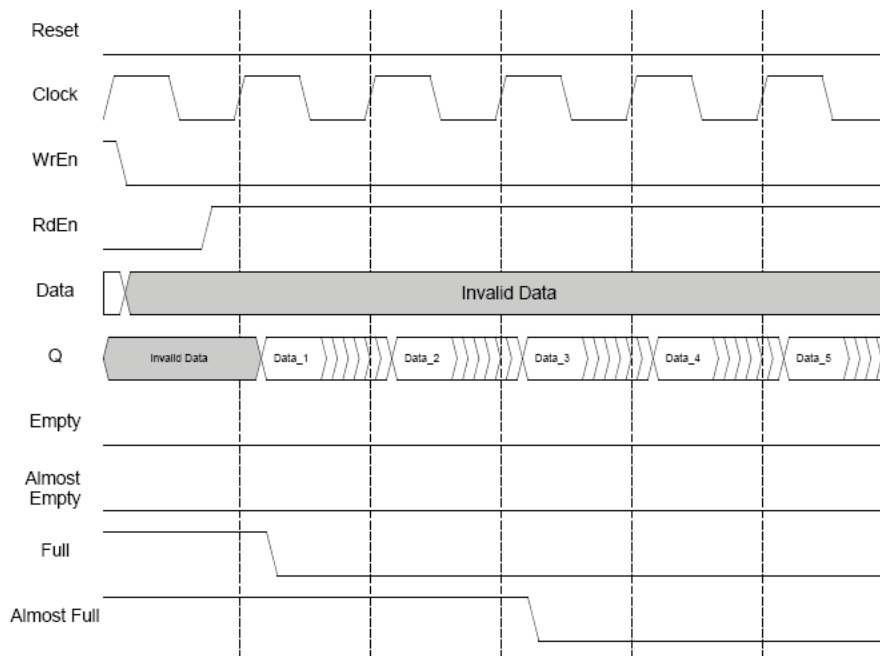
図11-22 出力レジスタのないFIFO、データライト・サイクルの終わり



この場合、Almost FullフラグはFIFOが満たされる2つ前の位置にあります。FIFOのN-2位置に書かれるとAlmost Fullフラグはアサートされ、そして最終ワードが書かれるとFullフラグがアサートされます。FIFOがフルであるため(FullフラグがHigh)、Data_Xデータ入力は書かれません。

次にFIFOの内容が読み出されときの波形を見ます。図11-23はリードサイクルの始まりを示します。RdEnがHighになり、データリードが始まります。示されているようにFullとAlmost Fullフラグはネゲートされます。

図11-23 出力レジスタのないFIFO、データリード・サイクルの初め



同様に、データが読み出されてFIFOが空になると、Almost EmptyとEmptyフラグがアサートされます。

図11-24 出力レジスタのないFIFO、データリード・サイクルの終わり

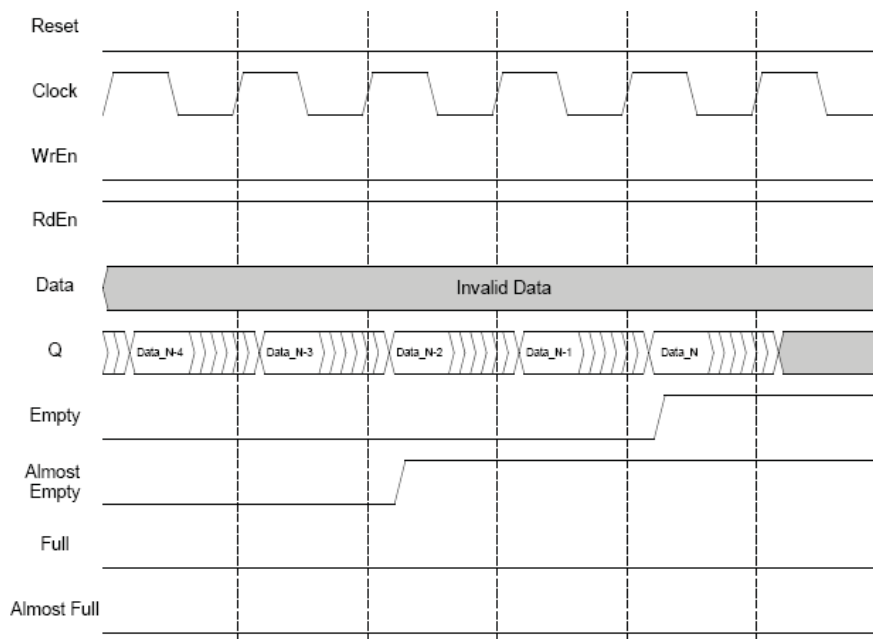


図11-21～11-24は出力レジスタのないFIFO（非パイプラインFIFO）の振舞いを示します。パイプライン・レジスタを有効にすると、出力データは1クロック周期遅れます。また、出力レジスタをRdEn信号によってイネーブルする付加的なオプションがあります。

図11-25から11-28は出力レジスタありFIFOとRdEnによる出力レジスタのイネーブルがあるFIFOの類似波形を示します。留意されるべきは、フラグが出力レジスタなしのFIFOと同様のタイミングでアサートされ、またネゲートされることです。1クロック周期遅れるのはデータ出力 'Q' のみです。

図11-25 出力レジスタのあるFIFO、データライト・サイクルの初め

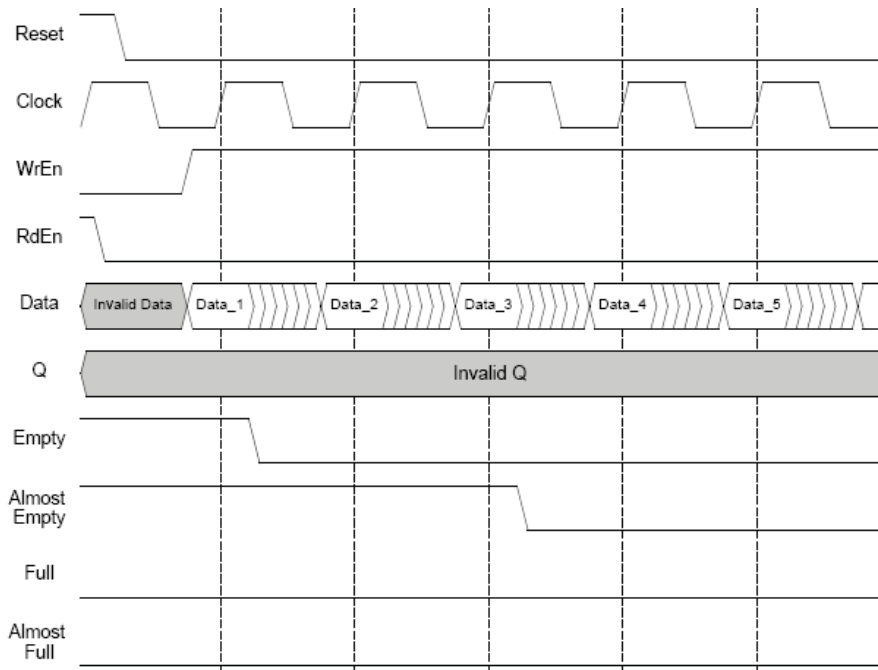


図11-26 出力レジスタのあるFIFO、データライト・サイクルの終わり

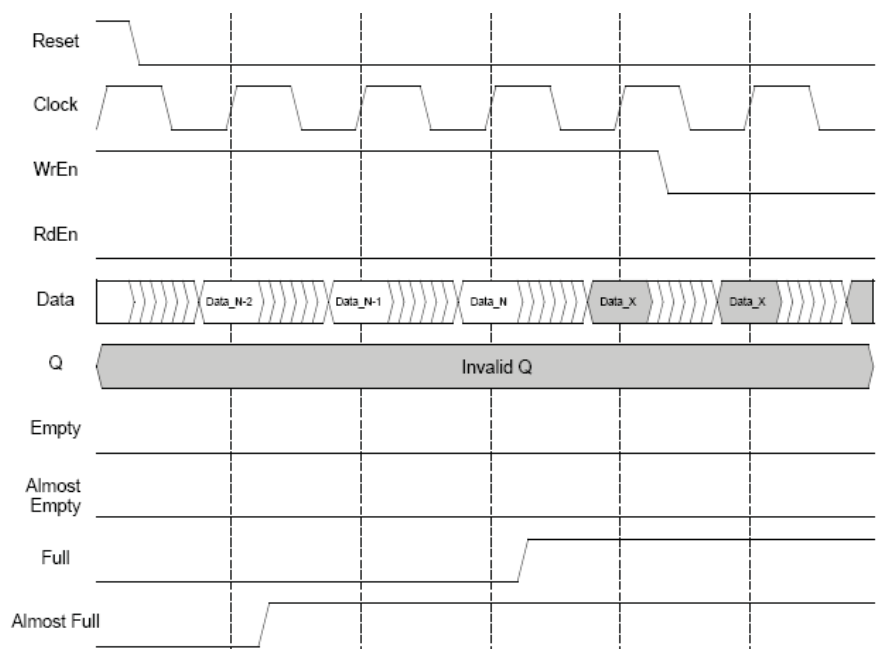


図11-27 出力レジスタのあるFIFO、データリード・サイクルの初め

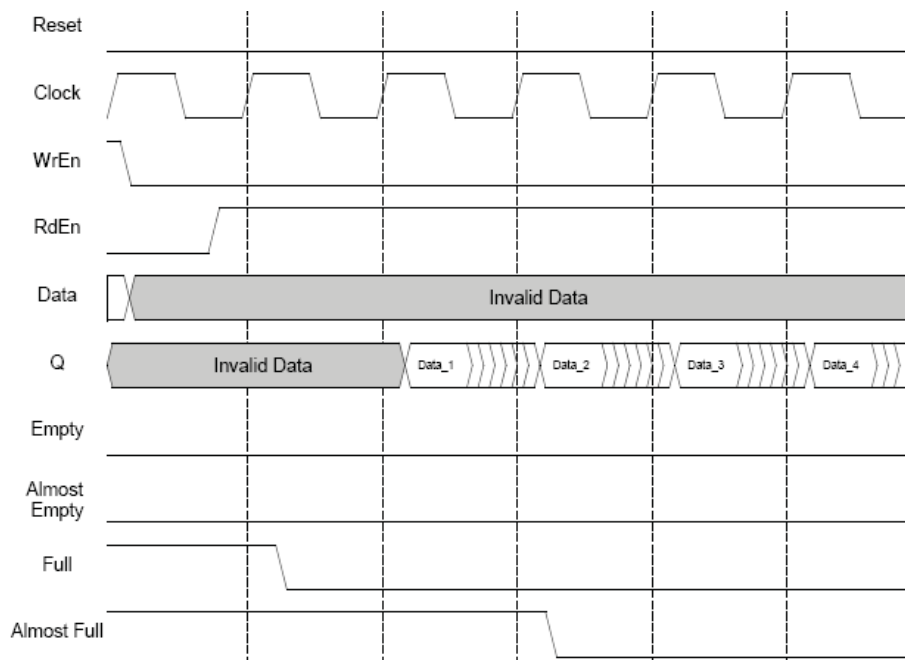
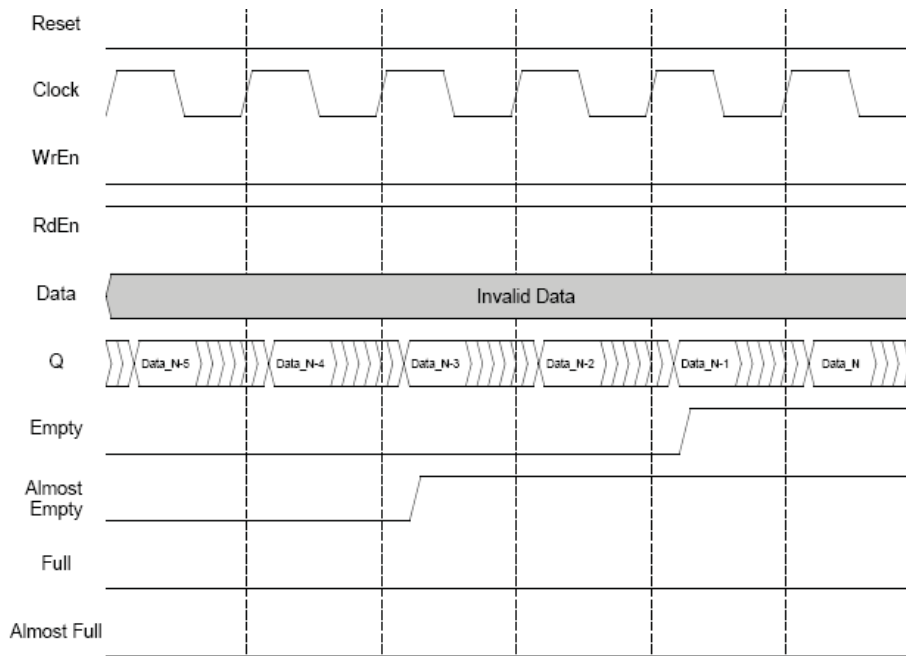
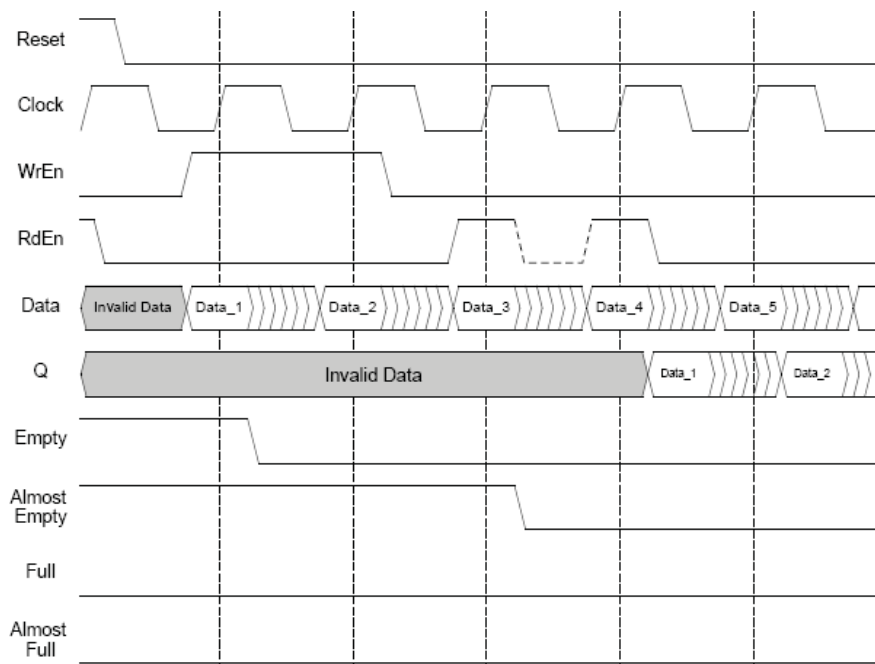


図11-28 出力レジスタのあるFIFO、データリード・サイクルの終わり



そして最後に、RdEnで出力レジスタをイネーブルするオプションを選択すると、(非パイプラインのFIFOと比べて)データ出力をさらに1クロック周期遅らせます。また、RdEnもそのクロック周期の間Highであるべきで、さもなければデータはRdEnがTrueになる時の余分なクロック周期がかかります。

図11-29 出力レジスタと、出力レジスタにRdEnのあるFIFO



デュアルクロック・ファーストイン・ファーストアウト(FIFO_DC) メモリ

FIFO_DC（またはデュアルクロックFIFO）もまたエミュレートFIFOです。FIFOと同様に、アドレスとフラグのロジック回路はRAM周りのFPGAファブリックに実装されます。FIFO_DCで利用できるポートは以下の通りです。

- Reset
- RPRreset
- WrClock
- RdClock
- WrEn
- RdEn
- Data
- Q
- Full フラグ
- Almost Full フラグ
- Empty フラグ
- Almost Empty フラグ

FIFO_DCフラグ

エミュレートFIFOとしてのFIFO_DCは、フラグがブロックRAM周りのFPGAロジックで実装される必要があります。2本のクロックのために、フラグがリードクロックからライトクロックに、また同様に逆方向にそれぞれクロックドメインをまたぐ必要があります。これはフラグのアサートやネゲートの際にレイテンシ(遅延)を加えます。レイテンシは(アサートかネゲートのどちらか)一方の場合だけ避けることができ、或いはこれら場合の間で分散することもできます。

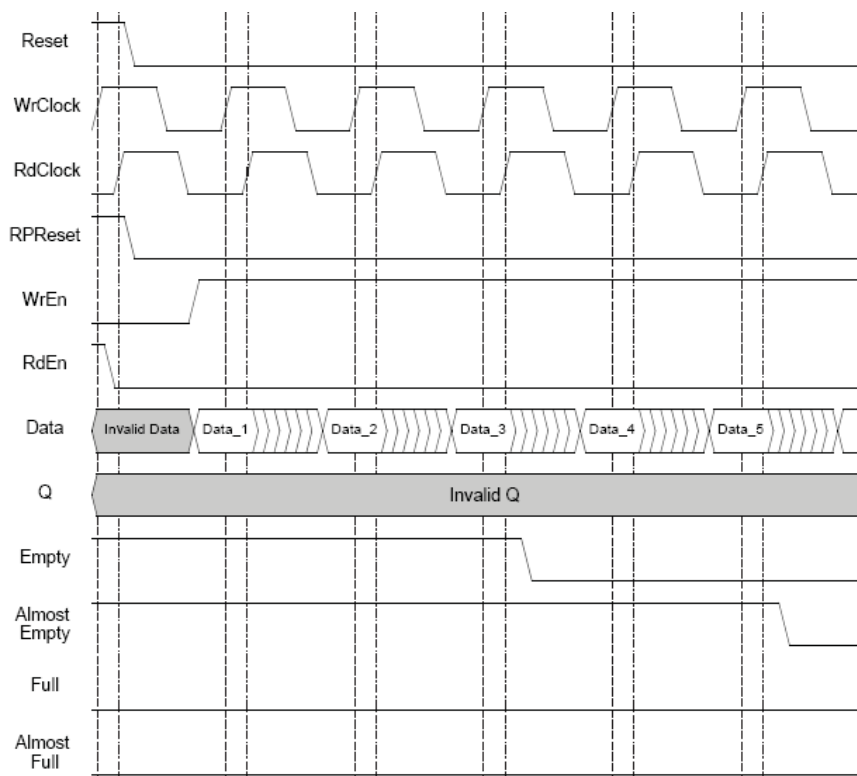
現行のエミュレートFIFO_DCでは、これらフラグのアサート時にレイテンシが全くなく、ラティスはこれが何よりも重要であると考えています。これらのフラグが真になる必要があるとき、レイテンシが全くありません。しかしながら、上述したように2クロックドメインで動作するフラグロジックの設計のために、ネゲートの際にはレイテンシがあります。

今、FIFO_DCを満たすために書き始めると仮定します。書き込み動作はWrClockとWrEnによって制御されます。しかしながら、EmptyとAlmost Emptyフラグのネゲートには付加的なRdClock周期がかかります。

他方、FullとAlmost FullのネゲートはFIFO_DCからのデータの読み出しが必要です。データを読んだ後、フラグが出力されるのに付加的なWrClock周期がかかります。

以上を理解した上で、出力レジスタなしFIFO_DCの波形を見ましょう。図11-30は、いつ空(empty)の状態か、またいつデータが書き始められるかというFIFO_DC動作を示します。

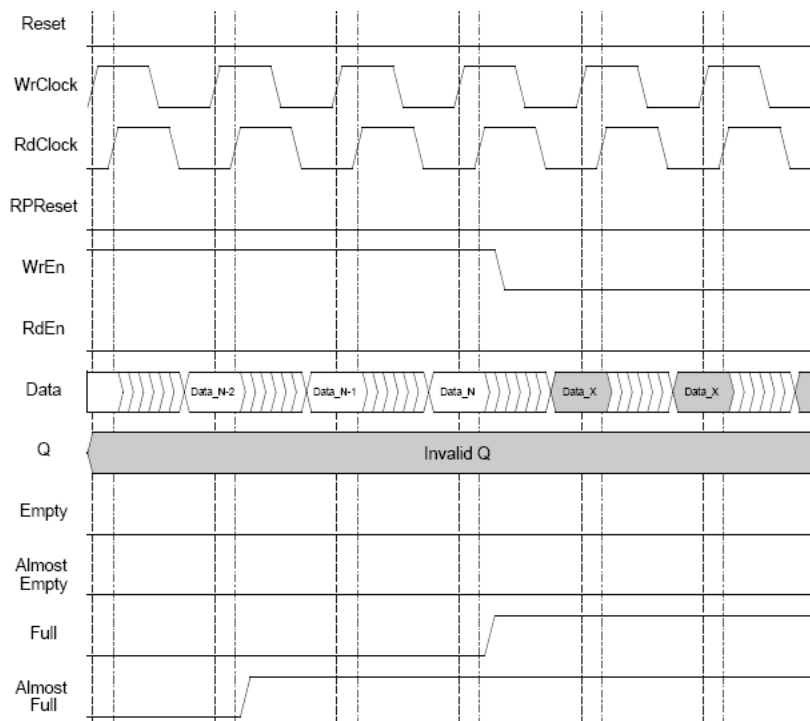
図11-30 出力レジスタのないFIFO_DC、データライト・サイクルの初め



FIFO_DCに書き始めるためにはWrEn信号はHighでなければなりません。初めはEmptyとAlmost EmptyフラグはHighで、またFullとAlmost FullはLowです。

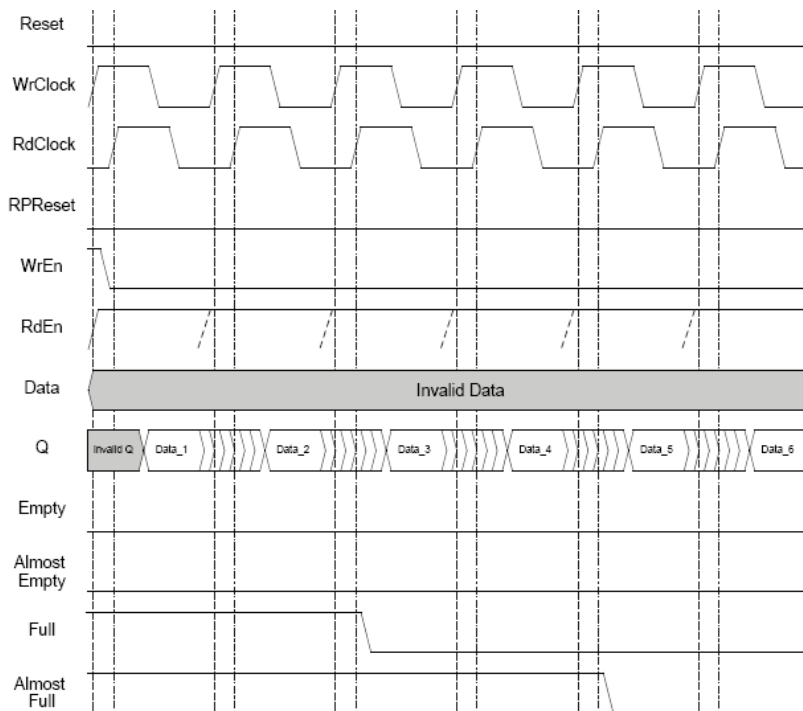
最初のデータがFIFOに書かれると、FIFO_DCはもう空ではないのでEmptyフラグはネゲートされ(Lowになります)。この図では、Almost Emptyのフラグ設定が3 (アドレス位置の3)であると想定しています。したがって、3番目のアドレス位置が満たされるとAlmost Emptyフラグはネゲートされます。

今、FIFO_DCを満たすために書き続けると仮定します。FIFO_DCが満たされると、Almost FullとFullの両フラグがアサートされます。図11-31はこれらフラグの動作を示します。この図ではFIFO_DCの深さが 'N' であると想定しています。

図11-31 出力レジスタのないFIFO_DC、データライト・サイクルの終わり

この場合、Almost FullフラグはFIFO_DCが満たされる2つ前の位置にあります。FIFO_DCのN-2位置に書かれるとAlmost Fullフラグはアサートされ、そして最終ワードが書かれるとFullフラグがアサートされます。

FIFO_DCがフルであるため(FullフラグがHigh)、Data_Xデータ入力は書かれません。これらのフラグはすぐにアサートされ、そのレイテンシが全くないことに留意してください。

図11-32 出力レジスタのないFIFO_DC、データリード・サイクルの初め

次にFIFO_DCの内容が読み出されときの波形を見ます。図11-32はリードサイクルの始まりを示します。RdEnがHighになり、データリードが始まります。示すようにAlmost FullとFullの両フラグがネゲートされます。この場合、ネゲートが2クロック周期遅れることに注意してください。

同様に、データが読み出されてFIFO_DCが空になると、Almost EmptyとEmptyフラグがアサートされます。

図11-33 出力レジスタのないFIFO_DC、データリード・サイクルの終わり

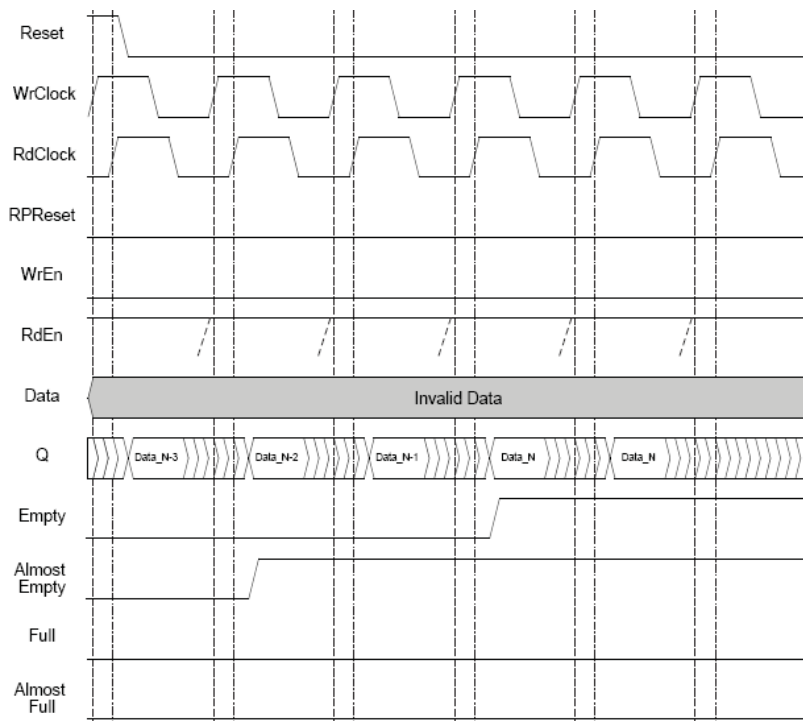


図11-33は出力レジスタのないFIFO_DC(非パイプラインFIFO_DC)の振舞いを示します。パイプライン・レジスタを有効にすると、出力データは1クロック周期遅れます。また、出力レジスタをRdEn信号によってイネーブルする付加的なオプションがあります。

図11-34から11-37は出力レジスタありFIFO_DCと、RdEnによる出力レジスタのイネーブルがないFIFO_DCの類似波形を示します。留意されるべきは、フラグが出力レジスタなしのFIFO_DCと同様のタイミングでアサートされ、またネゲートされることです。1クロック周期遅れるのはデータ出力 'Q' のみです。

図11-34 出力レジスタのあるFIFO_DC、データライト・サイクルの初め

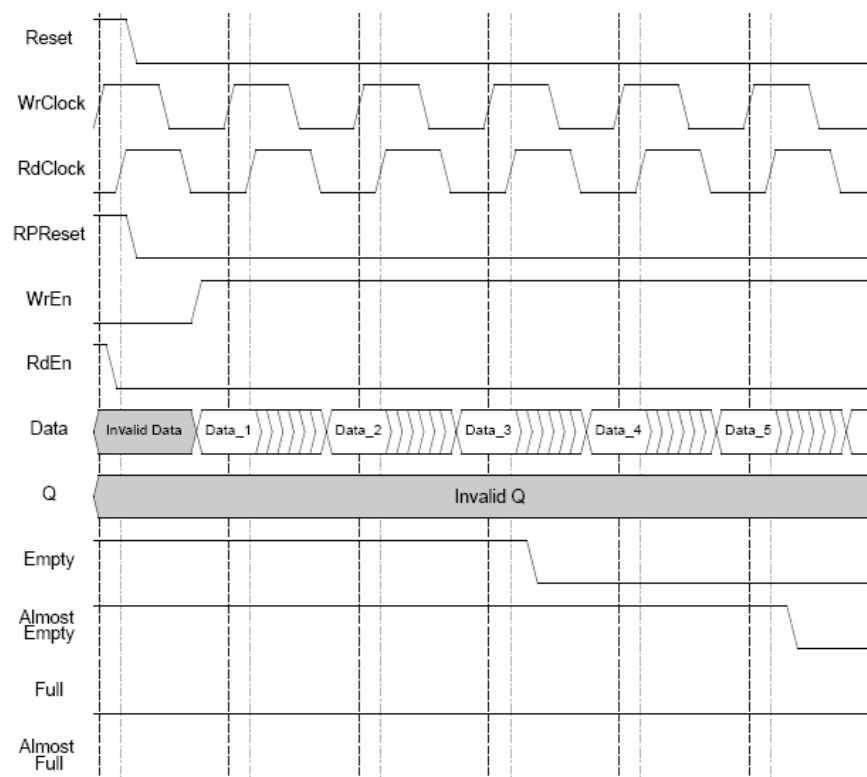


図11-35 出力レジスタのあるFIFO_DC、データライト・サイクルの終わり

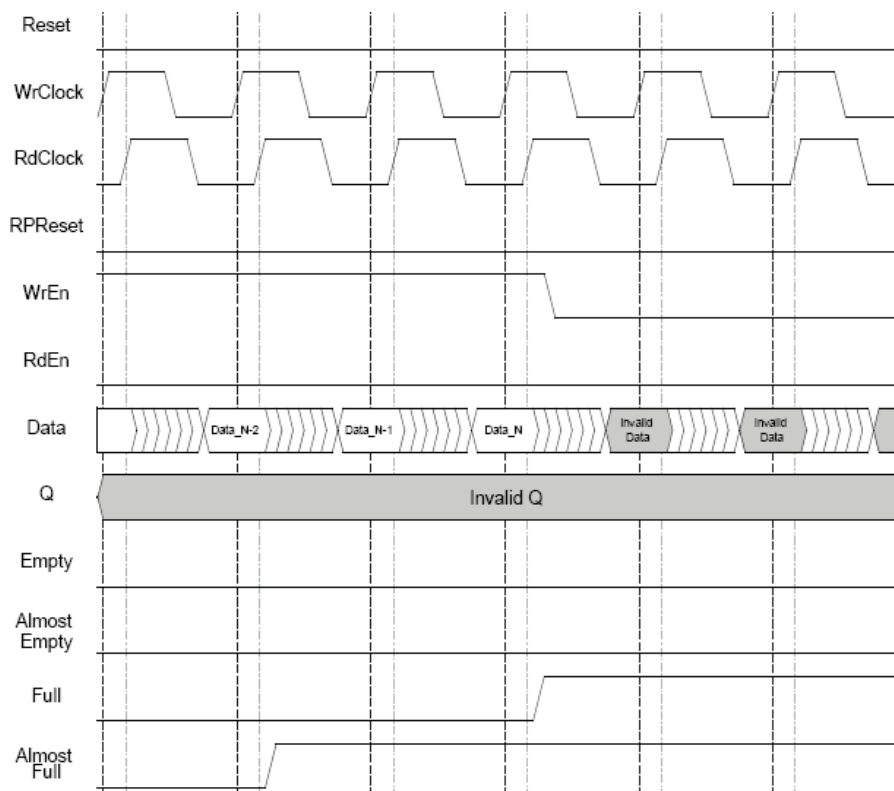


図11-36 出力レジスタのあるFIFO_DC、データリード・サイクルの初め

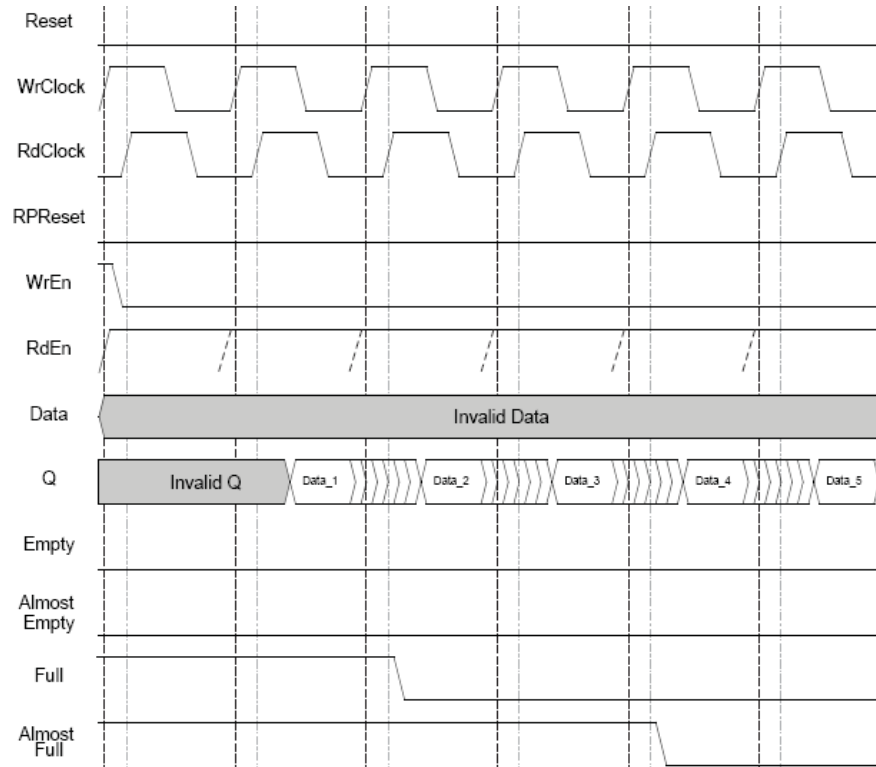
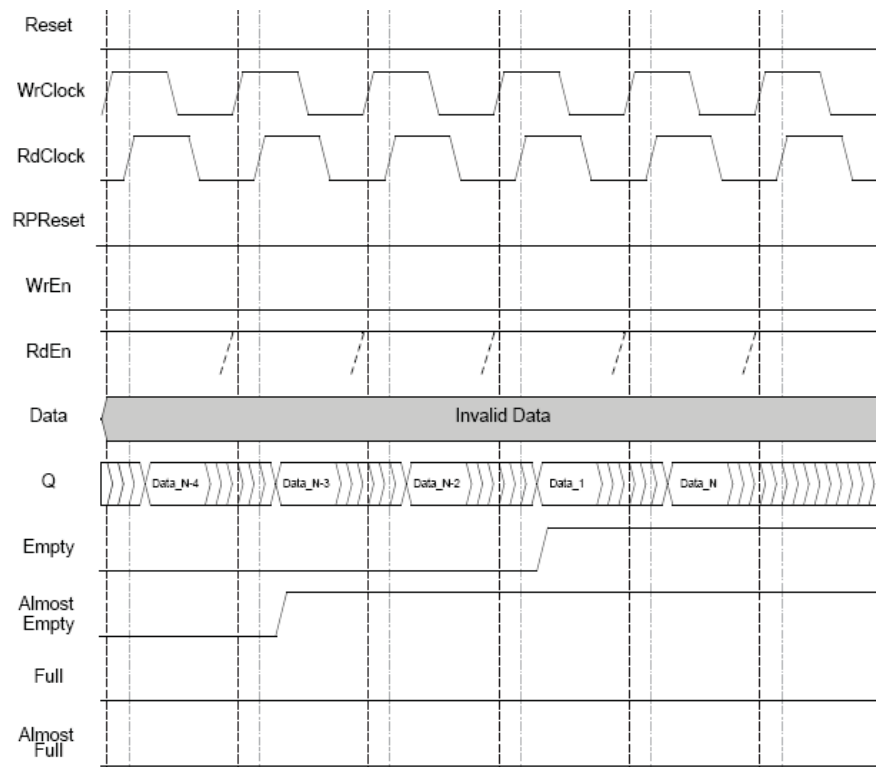


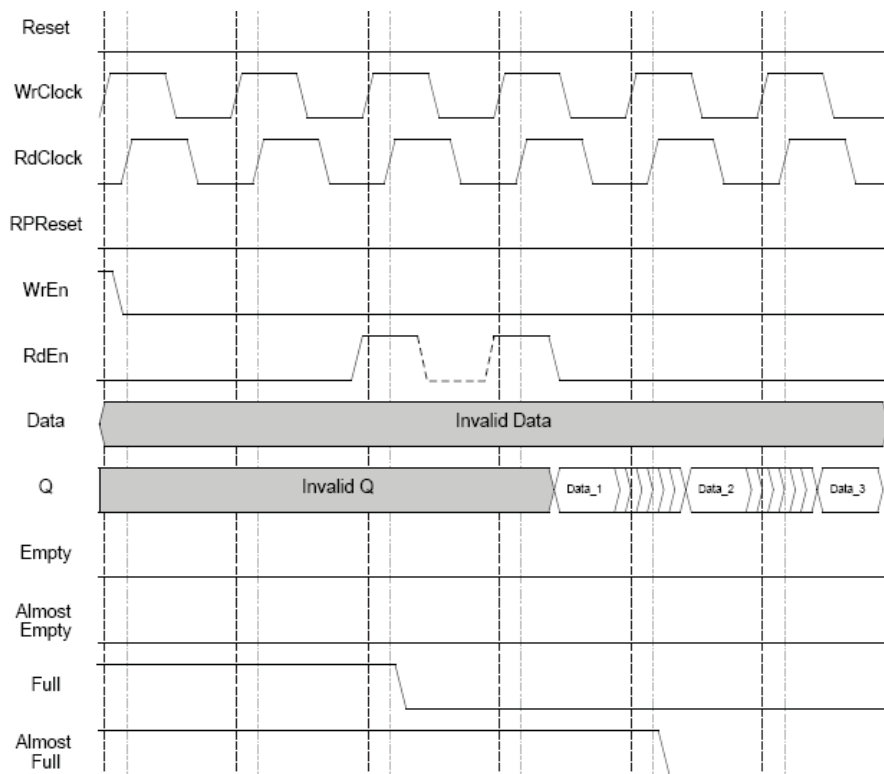
図11-37 出力レジスタのあるFIFO_DC、データリード・サイクルの終わり



そして最後に、RdEnで出力レジスタをイネーブルするオプションを選択すると、(非パイプラインの

FIFO_DCと比べて)データ出力をさらに1クロック周期遅れます。また、RdEnもそのクロック周期の間Highであるべきで、さもなければデータはRdEnが真になる時の付加的なクロック周期がかかります。

図11-38 出力レジスタと、出力レジスタにRdEnのあるFIFO_DC

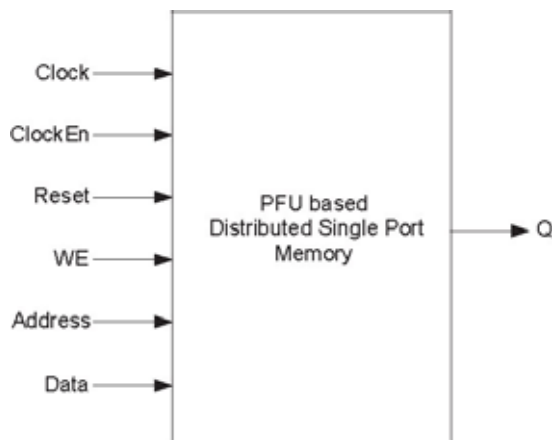


分散シングルポートRAM (Distributed_SPRAM) ; PFUベース

PFUベースの分散シングルポートRAMは、PFUで利用する4入力LUT(ルックアップ・テーブル)を用いることで生成されます。より大きい分散メモリサイズを作成するためにこれらのLUTを連結することができます。

図11-39はIPexpressによって生成される分散シングルポートRAMモジュールを示します。

図11-39 IPexpress によって生成された分散シングルポートRAMモジュール



生成されたモジュールはPFU内にある4入力LUTを利用します。Clock、ClockEn、およびResetのような追加

ロジックは、PFU内にあるリソースを利用することによって生成されます。

リードクロック(RdClock)やリードクロック・イネーブル(RdClockEn)などのポートはハードウェア・プリミティブでは利用できません。IPexpress内のコンフィグレーションで、ユーザが出力レジスタをイネーブルすると、これに対応したモジュールが生成されます。

メモリの各ポートとその定義は表11-14にまとめられています。この表のリストはIPexpressで生成されたモジュールとプリミティブに対応します。

表11-14 PFUベース分散シングルポートRAMのポート定義

生成されたモジュールの ポート名	PFUプリミティブの ポート名	記述	アクティブなステート
Clock	CK	クロック	立ち上がりエッジ
ClockEn	-	クロック・イネーブル	Highアクティブ
Reset	-	リセット	Highアクティブ
WE	WRE	ライト・イネーブル	Highアクティブ
Address	AD[3:0]	アドレス	—
Data	DI[1:0]	データ入力	—
Q	DO[1:0]	データ出力	—

Clock Enable (ClockEn)などのポートはハードウェア・プリミティブにはありません。これらは、ユーザが出力レジスタをイネーブルしする設定を選択すると、IPexpressによって生成されます。

分散シングルポートRAM (Distributed_SPRAM)の出力レジスタをイネーブルするかどうかのオプションがあります。これらに対する内部タイミング波形を、それぞれ図11-40から図11-41に示します。

図11-40 PFUベース分散シングルポートRAMタイミング波形 ～ 出力レジスタなし

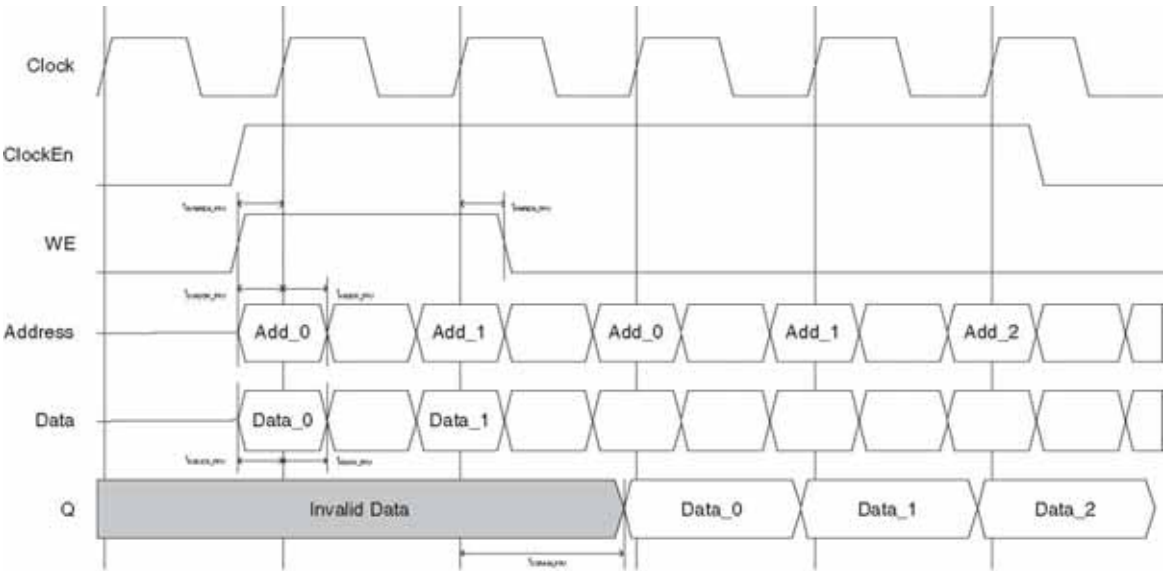
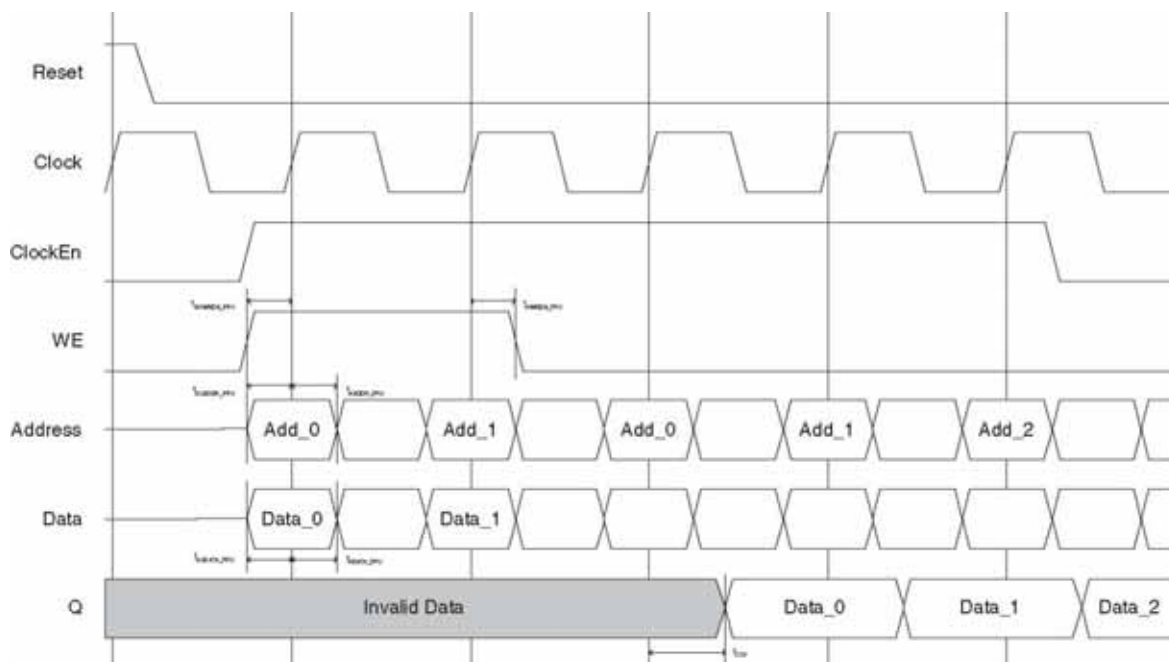


図 11-41 PFUベース分散シングルポートRAMタイミング波形 ～ 出力レジスタあり

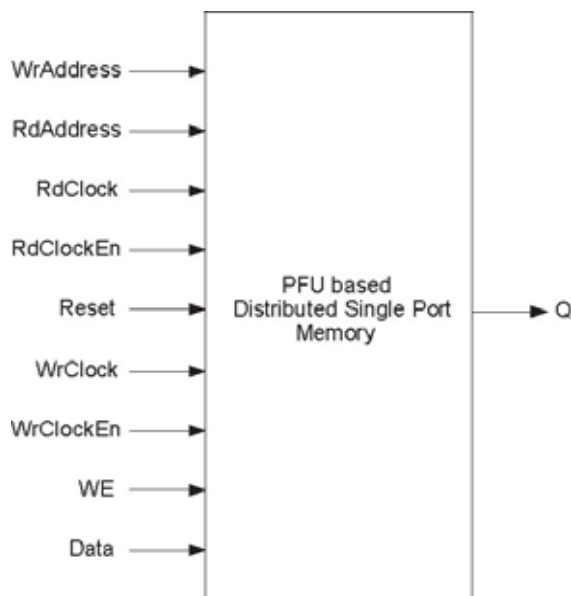


分散デュアルポートRAM (Distributed_DPRAM) ～ PFUベース

また、PFUベースの分散デュアルポートRAMは、PFUで利用する4入力LUTを用いることで生成されます。より大きい分散メモリ・サイズを作成するためにこれらのLUTを連結することができます。

図11-42はIPexpressによって生成される分散シングルポートRAMモジュールを示します。

図11-42 IPexpressによって生成された分散デュアルポートRAMモジュール



生成されたモジュールはPFU内にある4入力LUTを利用します。Clocks、Clock Enables、およびResetのための追加ロジックは、PFUにあるリソースを利用することによって生成されます。

リードクロック(RdClock)やリード・クロック・イネーブル(RdClockEn)などのポートはハードウェア・プリミティブでは利用できません。モジュール・マネージャ内のコンフィギュレーションで、ユーザが出力レジスタをイネーブルすると、これに対応したモジュールが生成されます。

メモリの種々ポートとそれらの定義は表11-15に示されています。表はIPexpressによって生成されたモジュールとプリミティブの対応するポートを記載します。

表11-15 PFUベースの分散デュアルポートRAMのポート定義

生成モジュールのポート名	PFUプリミティブのポート名	記述	アクティブなステート
WrAddress	WAD[3:0]	ライトアドレス	—
RdAddress	RAD[3:0]	リードアドレス	—
RdClock	—	リードクロック	立ち上がりエッジ
RdClockEn	—	リードクロック・イネーブル	Highアクティブ
WrClock	WCK	ライトクロック	立ち上がりエッジ
WrClockEn	—	ライトクロック・イネーブル	Highアクティブ
WE	WRE	ライトイネーブル	Highアクティブ
Data	DI[1:0]	データ入力	—
Q	RDO[1:0]	データ出力	—

分散デュアルポートRAM (Distributed DPRAM)の出力レジスタをイネーブルするかどうかのオプションがあります。これらに対する内部タイミング波形を、それぞれ図11-43から図11-44に示します。

図11-43 PFUベース分散デュアルポートRAMタイミング波形 ~ 出力レジスタなし

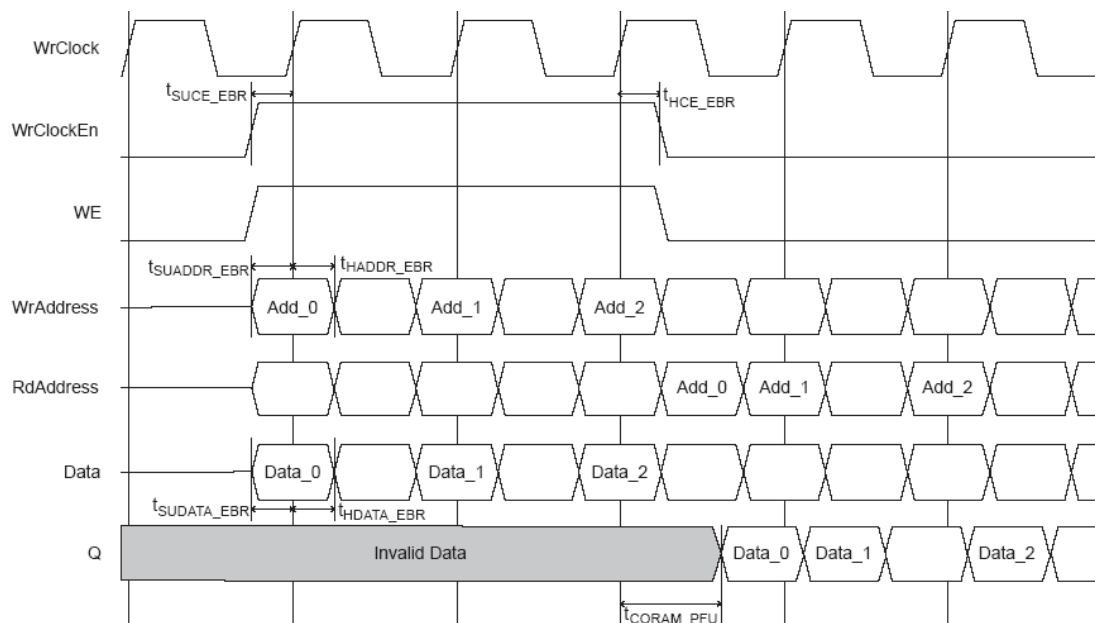
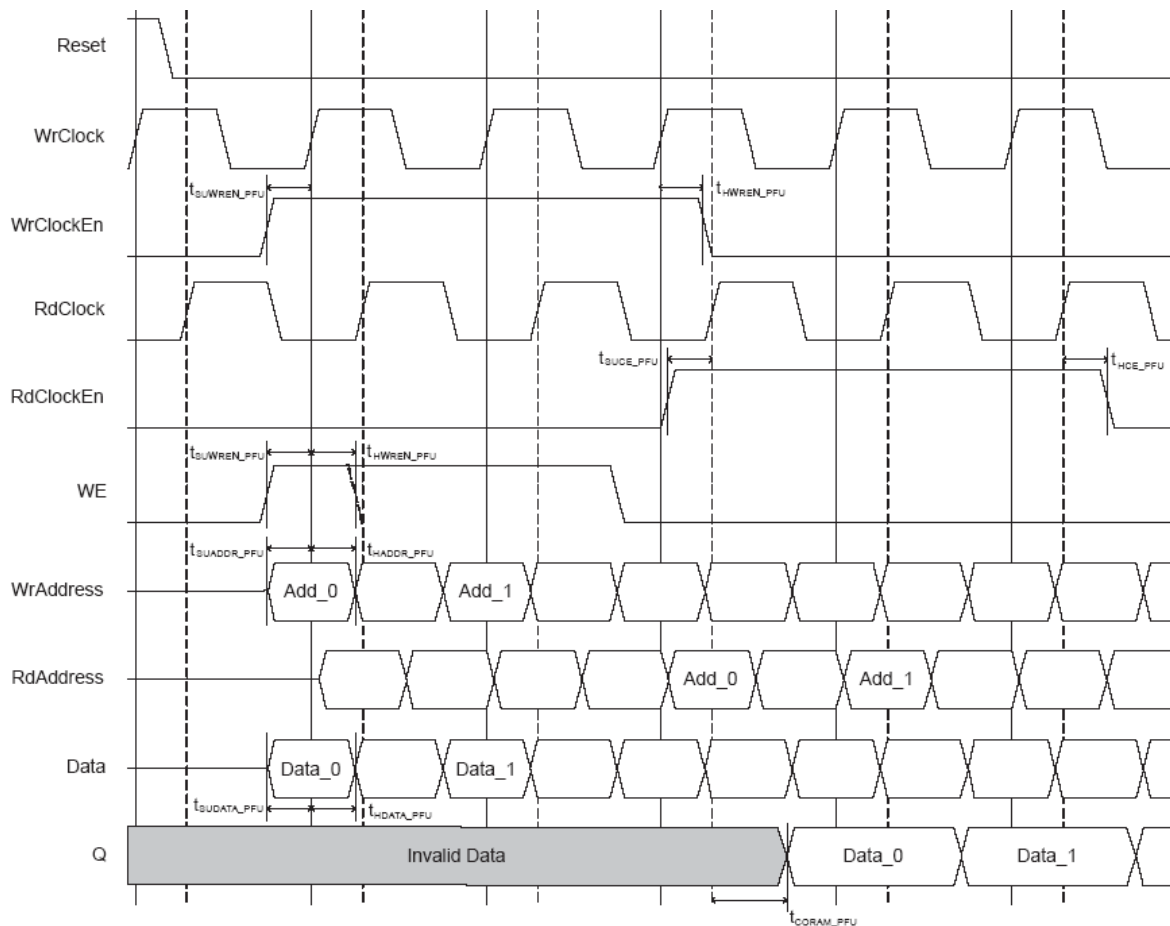


図11-44 PFUベース分散デュアルポートRAMタイミング波形 ～ 出力レジスタあり

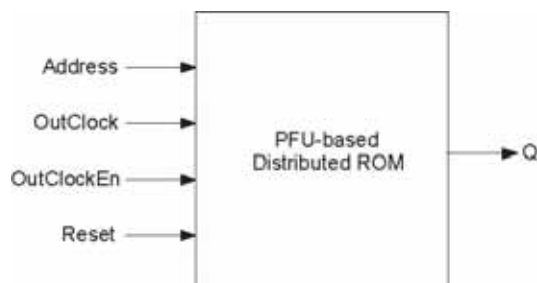


分散ROM (Distributed_ROM) ～ PFUベース

また、PFUベースの分散ROMは、PFUにある4入力LUTを用いることで作成されます。より大きい分散メモリ・サイズを作成するためにこれらのLUTを連結することができます。

図11-45はIPexpressによって生成される分散シングルポートRAMモジュールを示します。

図11-45 IPexpressによって生成された分散ROM



生成されるモジュールはPFUにある4入力LUTを利用します。ClockやResetなどの追加ロジックはPFUにあるリソースを活用することで生成できます。

Out Clock (OutClock)やOut Clock Enable (OutClockEn)などのポートはハードウェア・プリミティブで利用

できません。IPexpress内のコンフィグレーションで、ユーザが出力レジスタをイネーブルすると、これに対応したモジュールが生成されます。

メモリの種々ポートとそれらの定義は表11-16に示されています。表はIPexpressによって生成されたモジュールとプリミティブの対応するポートを記載します。

表11-16 PFUベースの分散ROMポート定義

生成されたモジュールの ポート名	PFUプリミティブ のポート名	記述	アクティブなステート
Address	AD[3:0]	アドレス	—
OutClock	—	出力クロック	立ち上がりエッジ
OutClockEn	—	出力クロック・イネーブル	Highアクティブ
Reset	—	リセット	Highアクティブ
Q	DO	データ出力	—

分散ROM (Distributed_ROM)の出力レジスタをイネーブルするかどうかのオプションに対する内部タイミング波形を、それぞれ図11-46から図11-47に示します。

図11-46 PFUベース分散ROMタイミング波形 ～ 出力レジスタなし

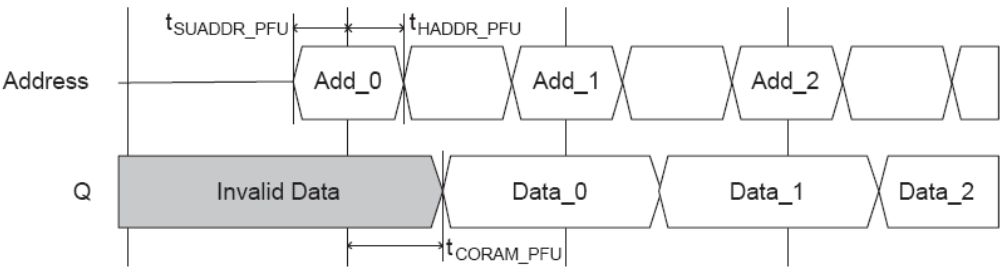
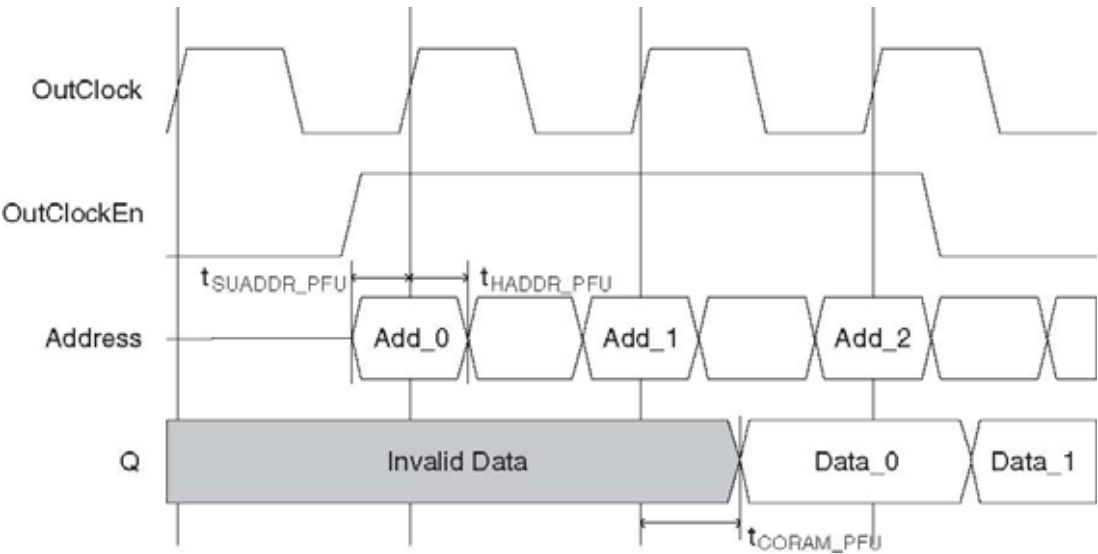


図11-47 PFUベース分散ROMタイミング波形 ～ 出力レジスタあり



メモリの初期化

EBRベースのROMおよびRAMモードで、メモリ・アレイ内の各ビットのパワーオン後の状態を指定することが可能です。メモリ・アレイにおける各ビットは0か1かどちらかの値を持つことができます。

初期化ファイル形式

初期化ファイルはASCIIファイルで、ユーザはどのテキストエディタでも使用できます。IPexpressは3タイプのメモリフォーマットをサポートします。

- バイナリファイル
- 16進(Hex)ファイル
- アドレス付き16進 (Addressed Hex)

メモリ初期化ファイルのファイル名は*.mem (<file_name>.mem)です。各列は、特定のメモリ領域に格納される値について表現します。キャラクタ数(または、コラム数)は各アドレス(または、メモリ・モジュールの幅)のビット数を表わします。

初期化ファイルは、主にROMを構成するために用いられます。RAMモードのEBRもまた、メモリ内容をプリロードするためにオプションとしてこの初期化ファイルを用いることができます。

バイナリ・ファイル

ファイルは、本質的には0と1のテキスト・ファイルです。列はワード数を、コラムがメモリ幅を示します。

メモリ・サイズ 20x32

```
00100000010000000010000001000000
00000001000000010000000100000001
000000100000000100000001000000010
000000110000000110000001100000011
00000100000001000000010000000100
00000101000001010000010100000101
00000110000001100000011000000110
00000111000001110000011100000111
00001000010010000000100001001000
00001001010010010000100101001001
00001010010010100000101001001010
00001011010010110000101101001011
00001100000011000000110000001100
00001101001011010000110100101101
00001110001111100000111000111110
00001111001111110000111100111111
00010000000100000001000000010000
00010001000100010001000100010001
00010010000100100001001000010010
00010011000100110001001100010011
```

16進ファイル

16進ファイルは本質的には、類似の列・コラムでアレンジされた16進キャラクタのテキストファイルです。ファイル内の列の数はアドレスの位置の数と同じで、各列がメモリ位置の内容を示しています。

メモリ・サイズ 8x16

```
A001
0B03
1004
CE06
0007
040A
```

0017
02A4

アドレス付き16進

アドレス付き16進はアドレスとデータによる行から成ります。各行はアドレスから始まり、コロン、そしていろいろなデータが続きます。memfileの形式は “アドレス: データ データ データ データ…” で、アドレスとデータは16進数です。

-A0 : 03 F3 3E 4F
-B2 : 3B 9F

最初の行はアドレスA0に03を置き、アドレスA1にF3、アドレスA2に3E、およびアドレスA3に4Fを置きます。第二行はアドレスB2に3Bを、アドレスB3に9Fを置きます。

アドレスとデータの値に制限は全くありません。値の範囲はaddr_widthとdata_width値に基づいて自動的にチェックされます。アドレスかデータ値に誤りがあれば、エラーメッセージがプリント（表示）されます。ユーザはすべてのアドレスの位置でデータを指定する必要はありません。あるアドレスで、データが指定されない場合、その位置のデータは0に初期化されます。IPexpressは、論理合成フローとシミュレーション・フローで共にメモリの初期化を可能にします。

テクニカル・サポート支援

ホットライン: 1-800-LATTICE (North America)
+1-408-826-6002 (Outside North America)

e-mail: techsupport@latticesemi.com

インターネット: www.latticesemi.com

変更履歴（日本語版）

Rev.#	日付	変更箇所
1.3J	Apr. 2007	既発行
1.8J	Jan.6, 2008	1, 表11-1、ECP2M50 の誤りを訂正 2, 図11-4 更新 3, (p.11-5) 下から4つめの節に "If an EBR is ..." 記述 4 行追加 4, 表11-5、AddressA, AddressB のポート名更新 5, (p.11-22:33) First In First Out (FIFO, FIFO_DC) – EBR Based ・記述全面更新、旧図11-21/旧表11-14:17 削除 6, 図11-43/-44 更新 << 共通（複数）>> Read-Before-Write モードに関する記述の削除 ・表11-4 Write Mode の値と表下の記述 ・旧図11-8/-9 削除 ・表11-7 Write Mode の値と表下の記述 ・旧図11-15/-16 削除 ・(p.11-36) Appendix WRITEMODE 内記述 << rev1.8 英語版オリジナルとの違い、など >> ・(p.1) sysDSPに関する最終文章を除外 ・その他全般的な文章・用語の校正、不明瞭な図の差し替え

付録A.アトリビュート定義

DATA_WIDTH

データ幅はRAMとFIFO要素に関連しています。DATA_WIDTHアトリビュートは各ワードのビット数を定義します。それは各メモリモジュールで、RAMサイズ表で定義されるように値を取ります。

REGMODE

REGMODEアトリビュートが、メモリ内のパイプライン化をイネーブルするのに用いられます。このアトリビュートはRAMとFIFO要素に関連します。REGMODEアトリビュートはNOREGかOUTREGモードパラメータを取り、出力パイプラインレジスタをディセーブル、またはイネーブルします。

RESETMODE

RESETMODEアトリビュートで、ユーザはメモリ内のリセット方法を選択します。このアトリビュートはブロックRAM要素に関連しています。RESETMODEは2つのパラメータを取り、SYNCとASYNCSYNCは、メモリリセットがクロックと同期することを意味します。ASYNCSYNCは、メモリリセットがクロックに非同期であることを意味します。

CSDECODE

CSDECODEアトリビュートは、ブロックRAM要素に関連します。EBRプリミティブで利用できるポートのCSは、メモリが複数のEBRブロックを連結される必要があるときに役に立ちます。複数のEBRブロックが連結されるとき、CS信号はアドレスのMSBとなります。CSは3ビットバスなので、容易に8つのメモリを連結することができます。CSDECODEは以下のパラメータを取ります; "000", "001", "010", "011", "100", "101", "110", そして "111"です。CSDECODE値はCS[2:0]のデコード値を決定します。擬似デュアルポートRAM用に、CSDECODE_Wは書き込み時のチップセレクト・デコード、CSDECODE_Rが読み出し時のチップセレクト・デコードです。CSDECODE_AとCSDECODE_Bは真のデュアルポートRAM要素に用いられて、AとBポートにそれぞれ関連します。

WRITEMODE

WRITEMODEアトリビュートはブロックRAM要素に関連しています。それはNORMAL、およびWRITETHROUGHモードのパラメータを取ります。

NORMALモードで、書き込み動作の間は出力データを変えないか、または更新しません。本モードは全てのデータ幅でサポートされます。

WRITETHROUGHモードで、ライトサイクルの間、入力データで出力データを更新します。本モードは全てのデータ幅でサポートされます。

WRITEMODE_AとWRITEMODE_BはデュアルポートRAM要素に用いられて、真のデュアルポートRAMの場合にはAとBポートに関連します。

全ての（真のデュアルポート・モジュールの）モードで、一方のポートからのリードアクセスと、他方のポートからのライトアクセスを同じアドレスに対して行うことは勧められません。この場合読み出しデータは不定になり得ます。また、同じアドレスに同時ライトアクセスも勧められません（一方からHighを書き、他方からLowを書いた場合など、そのアドレスにストアされるデータは不定になります）。

この状況が起これる場合は、これを検出するために以下のような制御ロジックを実装することを推奨します。

- 1, リードデータが無効な可能性があることを示すステータス信号を設ける、或いは
- 2, 両ポートからの（同一アドレスへの）同時アクセスを防ぐ制御ロジックを実装する

GSR

GSRアトリビュートは、RAM要素のためにグローバルなセット/リセットをイネーブルするか、またはディセーブルするために用いられます。