


Arora V コンフィギャラブル機能ユニット (CFU)

ユーザーガイド

UG303-1.0J,2023-04-20

著作権について(2023)

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**高云、Gowin、及び GOWINSEMI 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の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かに拘わらず)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。

目次

目次	i
図一覧	iii
表一覧	iv
1 本マニュアルについて	1
1.1 マニュアル内容	1
1.2 関連ドキュメント	1
1.3 用語、略語	1
1.4 テクニカル・サポートとフィードバック	2
2 CFU の構造	3
2.1 CLS	4
2.1.1 CLS の動作モード	4
2.1.2 REG	4
2.2 CRU	5
3 CFU プリミティブ	6
3.1 LUT	6
3.1.1 LUT1	6
3.1.2 LUT2	8
3.1.3 LUT3	9
3.1.4 LUT4	11
3.1.5 Wide LUT	13
3.2 MUX	17
3.2.1 MUX2	17
3.2.2 MUX4	18
3.2.3 Wide MUX	20
3.3 ALU	23
3.4 FF	26
3.4.1 DFFSE	27
3.4.2 DFFRE	28
3.4.3 DFFPE	30
3.4.4 DFFCE	33

3.5 LATCH	35
3.5.1 DLCE	35
3.5.2 DLPE	37
3.6 SSRAM	39

図一覧

図 2-1 コンフィギャラブル機能ユニットの構造	3
図 2-2 CFU 内のレジスタの説明図	4
図 3-1 LUT1 のポート図	6
図 3-2 LUT2 のポート図	8
図 3-3 LUT3 のポート図	9
図 3-4 LUT4 のポート図	11
図 3-5 LUT5 のポート図	14
図 3-6 MUX2 のポート図	17
図 3-7 MUX4 のポート図	18
図 3-8 MUX8 のポート図	21
図 3-9 ALU のポート図	24
図 3-10 DFFSE のポート図	27
図 3-11 DFFRE のポート図	29
図 3-12 DFFPE のポート図	30
図 3-13 DFFCE のポート図	33
図 3-14 DLCE のポート図	36
図 3-15 DLPE のポート図	37

表一覧

表 1-1 用語、略語	1
表 2-1 CFU 内のレジスタ信号の説明	5
表 3-1 LUT1 のポートの説明	6
表 3-2 LUT1 のパラメータの説明	7
表 3-3 LUT1 の真理値表.....	7
表 3-4 LUT2 のポートの説明	8
表 3-5 LUT2 のパラメータの説明	8
表 3-6 LUT2 の真理値表.....	8
表 3-7 LUT3 のポートの説明	10
表 3-8 LUT3 のパラメータの説明	10
表 3-9 LUT3 の真理値表.....	10
表 3-10 LUT4 のポートの説明	11
表 3-11 LUT4 のパラメータの説明	12
表 3-12 LUT4 の真理値表.....	12
表 3-13 LUT5 のポートの説明	14
表 3-14 LUT5 のパラメータの説明	14
表 3-15 LUT5 の真理値表.....	15
表 3-16 MUX2 のポートの説明	17
表 3-17 MUX2 の真理値表	17
表 3-18 MUX4 のポートの説明	18
表 3-19 MUX4 の真理値表	19
表 3-20 MUX8 のポートの説明	21
表 3-21 MUX8 の真理値表	21
表 3-22 ALU の機能.....	23
表 3-23 ALU のポートの説明	24
表 3-24 ALU のパラメータの説明	24
表 3-25 FF プリミティブ	26
表 3-26 FF のタイプ.....	26
表 3-27 DFFSE のポートの説明	27

表 3-28 DFFSE のパラメータの説明	27
表 3-29 DFFRE のポートの説明	29
表 3-30 DFFRE のパラメータの説明	29
表 3-31 DFFPE のポートの説明	31
表 3-32 DFFPE のパラメータの説明	31
表 3-33 DFFCE のポートの説明	33
表 3-34 DFFCE のパラメータの説明	33
表 3-35 LATCH のタイプ	35
表 3-36 DLCE のポートの説明.....	36
表 3-37 DLCE のパラメータの説明	36
表 3-38 DLPE のポートの説明.....	38
表 3-39 DLPE のパラメータの説明.....	38

1 本マニュアルについて

1.1 マニュアル内容

このマニュアルは、主に Arora V FPGA 製品の CFU の構造、動作モード、およびプリミティブ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 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：

- GW5AT シリーズ FPGA 製品データシート([DS981](#))
- GW5A シリーズ FPGA 製品データシート([DS1103](#))
- GW5AST シリーズ FPGA 製品データシート([DS1104](#))
- Gowin ソフトウェア ユーザーガイド([SUG100](#))
- Arora V BSRAM & SSRAM ユーザーガイド([UG300](#))

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
ALU	Arithmetic Logic Unit	算術論理演算装置
BSRAM	Block Static Random Access Memory	ブロック SRAM
CFU	Configurable Function Unit	コンフィギュラブル機能ユニット
CLS	Configurable Logic Section	コンフィギュラブル論理セクション
CRU	Configurable Routing Unit	コンフィギュラブル配線ユニット
DFF	D Flip Flop	D フリップフロップ
DL	Data Latch	データラッチ

用語、略語	正式名称	意味
LUT	Look-up Table	ルックアップテーブル
MUX2	Multiplexer 2:1	2 : 1 マルチプレクサ
REG	Register	レジスタ
SSRAM	Shadow Static Random Access Memory	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。

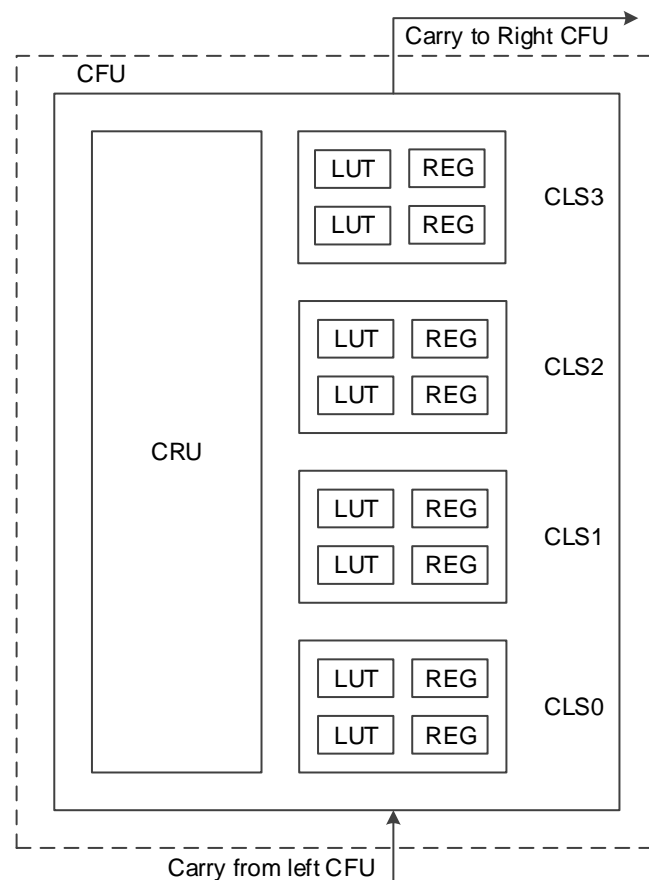
Web サイト : www.gowinsemi.com/ja

E-mail : support@gowinsemi.com

2CFU の構造

コンフィギュラブル機能ユニット(CFU)は、Gowin FPGA 製品のコアを構成する基本構成要素です。各基本構成要素は、4 つのコンフィギュラブル論理セクション(CLS)と対応するコンフィギュラブル配線ユニット(CRU)で構成されます。その中で、各 CLS には 2 つの 4 入力ルックアップテーブル(LUT)と 2 つのレジスタ(REG)が含まれます(図 2-1)。CFU 内の CLS は、アプリケーションシナリオに応じて、LUT、ALU、SRAM、および ROM として構成することができます。

図 2-1 コンフィギュラブル機能ユニットの構造



注記：

GW5AT デバイスは CLS3 の REG をサポートしており、CLS3 と CLS2 の CLK/CE/SR 信号は同じソースを共有します。

2.1 CLS

2.1.1 CLS の動作モード

CLS は、LUT モード、ALU モード、及びメモリモードをサポートします：

- LUT モード

各ルックアップテーブルは、1 つの 4 入力ルックアップテーブル (LUT4) として動作できます。さらに、CLS は、以下に示すように、LUT5/LUT6/LUT7/LUT8 などを実装できます。

- 1 つの CLS は、1 つの 5 入力ルックアップテーブル (LUT5) を形成できます。
- 2 つの CLS は、1 つの 6 入力ルックアップテーブル (LUT6) を形成できます。
- 4 つの CLS は、1 つの 7 入力ルックアップテーブル (LUT7) を形成できます。
- 8 つの CLS は、1 つの 8 入力ルックアップテーブル (LUT8) を形成できます。

- ALU モード

キャリーチェーンを利用することにより、LUT は ALU モードで動作して次の機能を実現することができます。

- 加算/減算
- 加算カウンタ及び減算カウンタを含むカウンタ
- 大なり比較、小なり比較、及び不等比較を含む比較器
- 乗算器

- メモリモード

このモードでは、1 つの CFU は 16 x 4 ビットの SRAM または ROM (ROM16) を形成できます。

2.1.2 REG

コンフィギュラブル論理セクション (CLS0~CLS3) にはそれぞれ、2 つのレジスタ (REG) が含まれています (図 2-2)。

図 2-2 CFU 内のレジスタの説明図

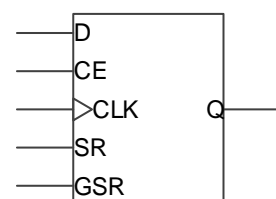


表 2-1 CFU 内のレジスタ信号の説明

信号名	I/O	説明
D	I	レジスタデータ入力 ^[1]
CE	I	アクティブ High またはアクティブ Low に構成できる CLK イネーブル信号 ^[2]
CLK	I	立ち上がりエッジトリガまたは立ち下がりエッジトリガに構成できるクロック信号 ²
SR	I	下記の機能に構成できるローカルセット/リセット入力 ^[2] : <ul style="list-style-type: none"> ● 同期リセット ● 同期セット ● 非同期リセット ● 非同期セット ● ローカルセット/リセットなし
GSR ^{[3],[4]}	I	下記の機能に構成できるグローバルセット/リセット ^[4] : <ul style="list-style-type: none"> ● 非同期リセット ● 非同期セット ● グローバルセット/リセットなし
Q	O	レジスタ出力

注記 :

- [1]信号 D のソースは、同じ CLS の LUT の出力または CRU の入力です。したがって、ルックアップテーブルが占有されている場合でも、レジスタは単独で使用できます。
- [2]CFU の CLS の CE/CLK/SR は、個別に構成できます(共線である CLS2/CLS3 を除く)。
- [3]Gowin FPGA 製品の内部では、GSR は CRU を経由することなく直接接続されています。
- [4]SR と GSR の両方が有効な場合、GSR が優先されます。

2.2 CRU

コンフィギュラブル配線ユニット(CRU)の主な機能は次のとおりです。

- 入力選択機能 : CFU の入力信号の入力ソース選択機能を提供します。
- 配線機能 : CFU の内部、CFU と CFU の間、および CFU と FPGA 内の他の機能ブロックの間など、CFU の入力と出力接続を実現します。

3 CFU プリミティブ

3.1 LUT

LUT には、LUT1、LUT2、LUT3、および LUT4 などがあります。これらの LUT は異なる入力ビット幅を持っています。

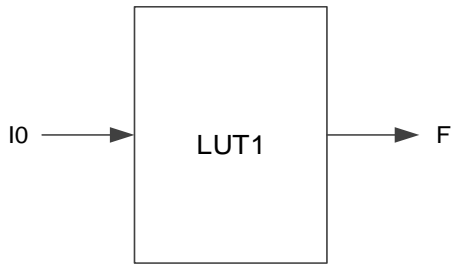
3.1.1 LUT1

プリミティブの紹介

LUT1(1-input Look-up Table)は最もシンプルな LUT で、通常バッファとインバーターの実現に使用されます。LUT1 は 1 入力ルックアップテーブルです。パラメータによって INIT に初期値を割り当てた後、入力したアドレスに応じて対応するデータを検索し、結果を出力します。

ポート図

図 3-1 LUT1 のポート図



ポートの説明

表 3-1 LUT1 のポートの説明

ポート	I/O	説明
I0	入力	データ入力信号
F	出力	データ出力信号

パラメータの説明

表 3-2 LUT1 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	2'h0~2'h3	2'h0	LUT1 の初期値

真理値表

表 3-3 LUT1 の真理値表

Input(I0)	Output(F)
0	INIT[0]
1	INIT[1]

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
LUT1 instName (
    .I0(I0),
    .F(F)
);
defparam instName.INIT=2'h1;
```

VHDL でのインスタンス化 :

```
COMPONENT LUT1
    GENERIC (INIT:bit_vector:=X"0");
    PORT(
        F:OUT std_logic;
        I0:IN std_logic
    );
END COMPONENT;

uut:LUT1
    GENERIC MAP(INIT=>X"0")
    PORT MAP (
        F=>F,
        I0=>I0
    );
```

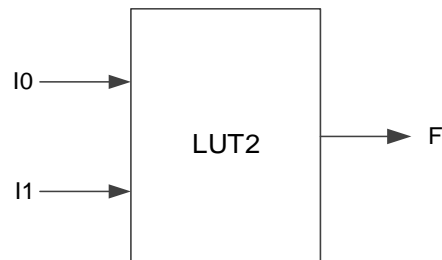
3.1.2 LUT2

プリミティブの紹介

LUT2(2-input Look-up Table) は 2 入力ルックアップテーブルです。パラメータによって INIT に初期値を割り当てた後、入力したアドレスに応じて対応するデータを検索し、結果を出力します。

ポート図

図 3-2 LUT2 のポート図



ポートの説明

表 3-4 LUT2 のポートの説明

ポート	I/O	説明
I0	入力	データ入力信号
I1	入力	データ入力信号
F	出力	データ出力信号

パラメータの説明

表 3-5 LUT2 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	4'h0~4'hf	4'h0	LUT2 の初期値

真理値表

表 3-6 LUT2 の真理値表

Input(I1)	Input(I0)	Output(F)
0	0	INIT[0]
0	1	INIT[1]
1	0	INIT[2]
1	1	INIT[3]

プリミティブのインスタンス化

Verilog でのインスタンス化：


```

LUT2 instName (
    .I0(I0),
    .I1(I1),
    .F(F)
);
defparam instName.INIT=4'h1;

```

VHDL でのインスタンス化 :

```

COMPONENT LUT2
    GENERIC (INIT:bit_vector:=X"0");
    PORT(
        F:OUT std_logic;
        I0:IN std_logic;
        I1:IN std_logic
    );
END COMPONENT;
uut:LUT2
    GENERIC MAP(INIT=>X"0")
    PORT MAP (
        F=>F,
        I0=>I0,
        I1=>I1
    );

```

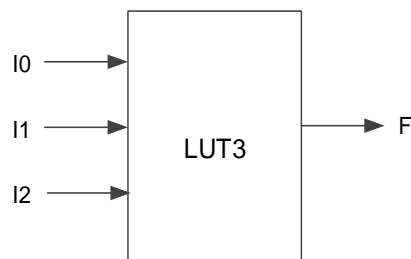
3.1.3 LUT3

プリミティブの紹介

LUT3(3-input Look-up Table) は 3 入力ルックアップテーブルです。パラメータによって INIT に初期値を割り当てた後、入力したアドレスに応じて対応するデータを検索し、結果を出力します。

ポート図

図 3-3 LUT3 のポート図



ポートの説明

表 3-7 LUT3 のポートの説明

ポート	I/O	説明
I0	入力	データ入力信号
I1	入力	データ入力信号
I2	入力	データ入力信号
F	出力	データ出力信号

パラメータの説明

表 3-8 LUT3 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	8'h00~8'hff	8'h00	LUT3 の初期値

真理値表

表 3-9 LUT3 の真理値表

Input(I2)	Input(I1)	Input(I0)	Output(F)
0	0	0	INIT[0]
0	0	1	INIT[1]
0	1	0	INIT[2]
0	1	1	INIT[3]
1	0	0	INIT[4]
1	0	1	INIT[5]
1	1	0	INIT[6]
1	1	1	INIT[7]

プリミティブのインスタンス化

Verilog でのインスタンス化：

```
LUT3 instName (
    .I0(I0),
    .I1(I1),
    .I2(I2),
    .F(F)
);
defparam instName.INIT=8'h10;
```

VHDL でのインスタンス化

```
COMPONENT LUT3
```

```

    GENERIC (INIT:bit_vector:=X"00");
    PORT(
        F:OUT std_logic;
        I0:IN std_logic;
        I1:IN std_logic;
        I2:IN std_logic
    );
END COMPONENT;
uut:LUT3
    GENERIC MAP(INIT=>X"00")
    PORT MAP (
        F=>F,
        I0=>I0,
        I1=>I1,
        I2=>I2
    );

```

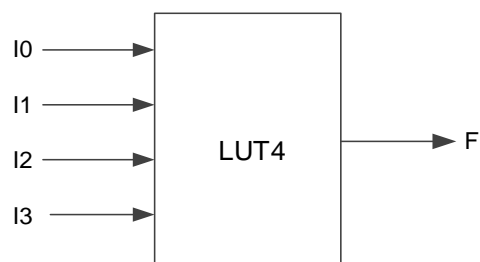
3.1.4 LUT4

プリミティブの紹介

LUT4(4-input Look-up Table) は 4 入力ルックアップテーブルです。パラメータによって INIT に初期値を割り当てた後、入力したアドレスに応じて対応するデータを検索し、結果を出力します。

ポート図

図 3-4 LUT4 のポート図



ポートの説明

表 3-10 LUT4 のポートの説明

ポート	I/O	説明
I0	入力	データ入力信号
I1	入力	データ入力信号
I2	入力	データ入力信号

ポート	I/O	説明
I3	入力	データ入力信号
F	出力	データ出力信号

パラメータの説明

表 3-11 LUT4 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	16'h0000~16'hffff	16'h0000	LUT4 の初期値

真理値表

表 3-12 LUT4 の真理値表

Input(I3)	Input(I2)	Input(I1)	Input(I0)	Output(F)
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]
1	1	0	1	INIT[13]
1	1	1	0	INIT[14]
1	1	1	1	INIT[15]

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
LUT4 instName (
    .I0(I0),
    .I1(I1),
    .I2(I2),
    .I3(I3),
    .F(F)
```

```

);
defparam instName.INIT=16'h1011;
VHDL でのインスタンス化 :
COMPONENT LUT4
    GENERIC (INIT:bit_vector:=X"0000");
    PORT(
        F:OUT std_logic;
        I0:IN std_logic;
        I1:IN std_logic;
        I2:IN std_logic;
        I3:IN std_logic
    );
END COMPONENT;
uut:LUT4
    GENERIC MAP(INIT=>X"0000")
    PORT MAP (
        F=>F,
        I0=>I0,
        I1=>I1,
        I2=>I2,
        I3=>I3
    );

```

3.1.5 Wide LUT

プリミティブの紹介

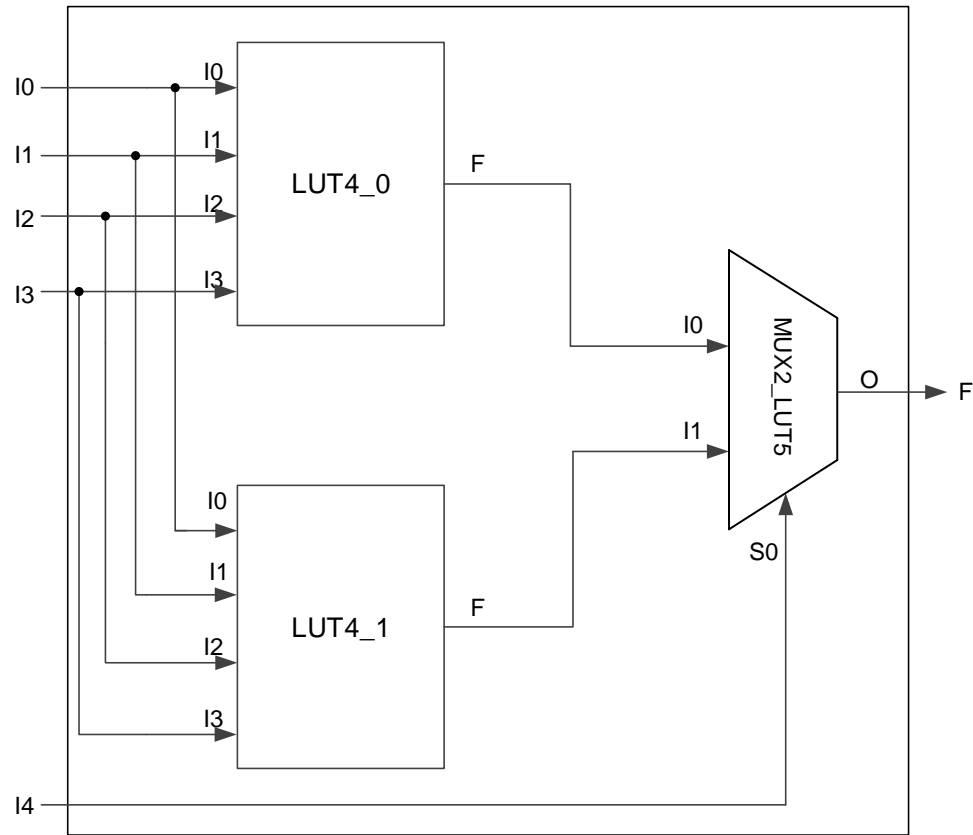
Wide LUT とは、LUT4 と MUX2 によって高次 LUT を形成することです。GOWIN FPGA は現在 MUX2_LUT5/ MUX2_LUT6/ MUX2_LUT7/ MUX2_LUT8 をサポートします。

高次 LUT は次のように構成されます。2 つの LUT4 と MUX2_LUT5 は LUT5、2 つの LUT5 と MUX2_LUT6 は LUT6、2 つの LUT6 と MUX2_LUT7 は LUT7、2 つの LUT7 と MUX2_LUT8 は LUT8 を形成します。

LUT5 を例に、Wide LUT の使用について紹介します。

ポート図

図 3-5 LUT5 のポート図



ポートの説明

表 3-13 LUT5 のポートの説明

ポート名	I/O	説明
I0	入力	データ入力信号
I1	入力	データ入力信号
I2	入力	データ入力信号
I3	入力	データ入力信号
I4	入力	データ入力信号
F	出力	データ出力信号

パラメータの説明

表 3-14 LUT5 のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	32'h00000~32'hffff	32'h00000	LUT5 の初期値

真理値表

表 3-15 LUT5 の真理値表

Input(I4)	Input(I3)	Input(I2)	Input(I1)	Input(I0)	Output(F)
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
LUT5 instName (  
    .I0(i0),  
    .I1(i1),  
    .I2(i2),  
    .I3(i3),  
    .I4(i4),  
    .F(f0)  
);  
defparam instName.INIT=32'h00000000;
```

VHDL でのインスタンス化 :

```
COMPONENT LUT5  
    PORT(  
        F:OUT std_logic;  
        I0:IN std_logic;  
        I1:IN std_logic;  
        I2:IN std_logic;  
        I3:IN std_logic;  
        I4:IN std_logic  
    );  
END COMPONENT;  
uut:LUT5  
    GENERIC MAP(INIT=>X"00000000")  
    PORT MAP (  
        F=>f0,  
        I0=>i0,  
        I1=>i1,  
        I2=>i2,  
        I3=>i3,  
        I4=>i4  
    );
```


3.2 MUX

MUX はマルチ入力を有するマルチプレクサで、チャンネル選択信号により 1 つのデータを選択して出力側に伝送します。GOWIN のプリミティブには、2 入力 1 出力と 4 入力 1 出力などのマルチプレクサがあります。

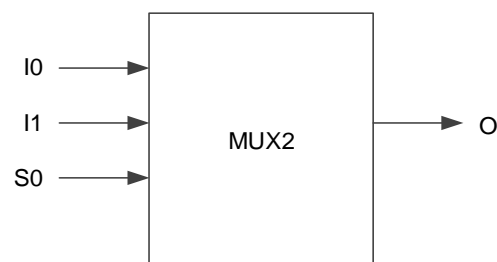
3.2.1 MUX2

プリミティブの紹介

MUX2(2-to-1 Multiplexer)は 2 入力 1 出力のマルチプレクサで、選択信号に従って、2 つの入力から 1 つを選択して出力します。

ポート図

図 3-6 MUX2 のポート図



ポートの説明

表 3-16 MUX2 のポートの説明

ポート	I/O	説明
I0	入力	データ入力信号
I1	入力	データ入力信号
S0	入力	データ選択信号
O	出力	データ出力信号

真理値表

表 3-17 MUX2 の真理値表

Input(S0)	Output(O)
0	I0
1	I1

プリミティブのインスタンス化

Verilog でのインスタンス化：

```

MUX2 instName (
    .I0(I0),
    .I1(I1),

```

```

        .S0(S0),
        .O(O)
    );

```

VHDL でのインスタンス化 :

```

COMPONENT MUX2
    PORT(
        O:OUT std_logic;
        I0:IN std_logic;
        I1:IN std_logic;
        S0:IN std_logic
    );
END COMPONENT;

uut:MUX2
    PORT MAP (
        O=>O,
        I0=>I0,
        I1=>I1,
        S0=>S0
    );

```

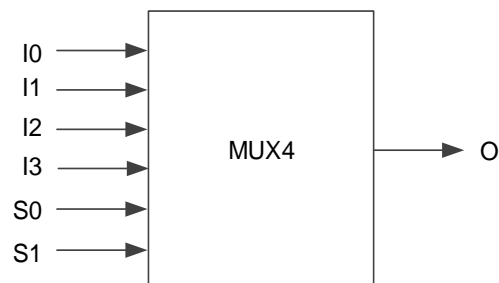
3.2.2 MUX4

プリミティブの紹介

MUX4(4-to-1 Multiplexer)は 4 入力 1 出力のマルチプレクサで、選択信号に従って、4 つの入力から 1 つを選択して出力します。

ポート図

図 3-7 MUX4 のポート図



ポートの説明

表 3-18 MUX4 のポートの説明

ポート	I/O	説明
-----	-----	----

ポート	I/O	説明
I0	入力	データ入力信号
I1	入力	データ入力信号
I2	入力	データ入力信号
I3	入力	データ入力信号
S0	入力	データ選択信号
S1	入力	データ選択信号
O	出力	データ出力信号

真理値表

表 3-19 MUX4 の真理値表

Input(S1)	Input(S0)	Output(O)
0	0	I0
0	1	I1
1	0	I2
1	1	I3

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```

MUX4 instName (
    .I0(I0),
    .I1(I1),
    .I2(I2),
    .I3(I3),
    .S0(S0),
    .S1(S1),
    .O(O)
);

```

VHDL でのインスタンス化 :

```

COMPONENT MUX4
    PORT(
        O:OUT std_logic;
        I0:IN std_logic;
        I1:IN std_logic;
        I2:IN std_logic;
        I3:IN std_logic;

```

```

        S0:IN std_logic;
        S1:IN std_logic

    );
END COMPONENT;
uut:MUX4
    PORT MAP (
        O=>O,
        I0=>I0,
        I1=>I1,
        I2=>I2,
        I3=>I3,
        S0=>S0,
        S1=>S1
    );

```

3.2.3 Wide MUX

プリミティブの紹介

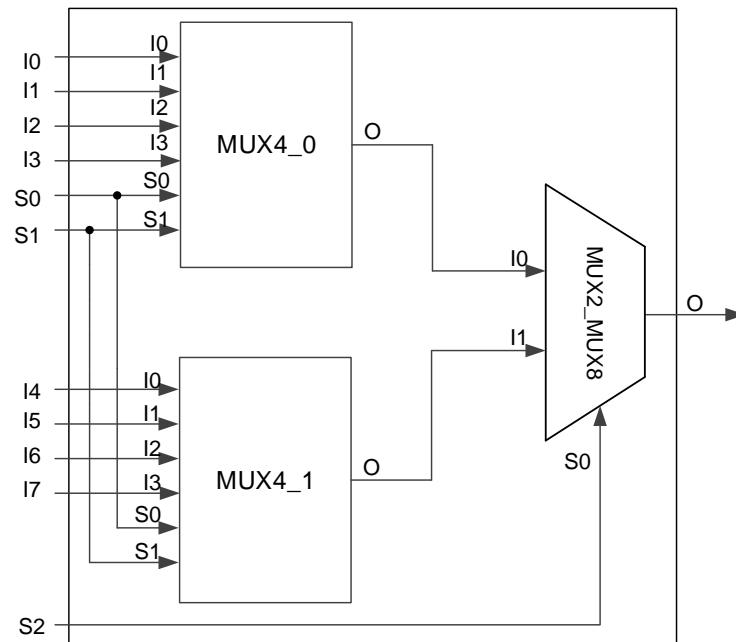
Wide LUT とは、MUX4 と MUX2 によって高次 MUX を形成することです。GOWIN FPGA は現在 MUX2_MUX8/ MUX2_MUX16/ MUX2_MUX32 をサポートします。

高次 MUX は次のように構成されます。2 つの MUX4 と MUX2_MUX8 は MUX8、2 つの MUX8 と MUX2_MUX16 は MUX16、2 つの MUX16 と MUX2_MUX32 は MUX32 を形成します。

MUX8 を例に、Wide MUX の使用について紹介します。

ポート図

図 3-8 MUX8 のポート図



ポートの説明

表 3-20 MUX8 のポートの説明

ポート	入力/出力	説明
I0	入力	データ入力信号
I1	入力	データ入力信号
I2	入力	データ入力信号
I3	入力	データ入力信号
I4	入力	データ入力信号
I5	入力	データ入力信号
I6	入力	データ入力信号
I7	入力	データ入力信号
S0	入力	データ選択信号
S1	入力	データ選択信号
S2	入力	データ選択信号
O	出力	データ出力信号

真理値表

表 3-21 MUX8 の真理値表

Input(S2)	Input(S1)	Input(S0)	Output(O)
0	0	0	I0
0	0	1	I1

Input(S2)	Input(S1)	Input(S0)	Output(O)
0	1	0	I2
0	1	1	I3
1	0	0	I4
1	0	1	I5
1	1	0	I6
1	1	1	I7

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```

MUX8 instName (
    .I0(i0),
    .I1(i1),
    .I2(i2),
    .I3(i3),
    .I4(i4),
    .I5(i5),
    .I6(i6),
    .I7(i7),
    .S0(s0),
    .S1(s1),
    .S2(s2),
    .O(o0)
);

```

VHDL でのインスタンス化 :

```

COMPONENT MUX8
    PORT(
        O:OUT std_logic;
        I0:IN std_logic;
        I1:IN std_logic;
        I2:IN std_logic;
        I3:IN std_logic;
        I4:IN std_logic;
        I5:IN std_logic;
        I6:IN std_logic;

```

```

        I7:IN std_logic;
        S0:IN std_logic;
        S1:IN std_logic;
        S2:IN std_logic

    );
END COMPONENT;
 uut:MUX8
    PORT MAP (
        O=>o0,
        I0=>I0,
        I1=>I1,
        I2=>I2,
        I3=>I3,
        I4=>I4,
        I5=>I5,
        I6=>I6,
        I7=>I7,
        S0=>S0,
        S1=>S1,
        S2=>S2
    );

```

3.3 ALU

プリミティブの紹介

ALU(2-input Arithmetic Logic Unit)は 2 入力算術論理演算装置で、ADD/SUB/ADDSUB などの機能を実現します(表 3-22)。

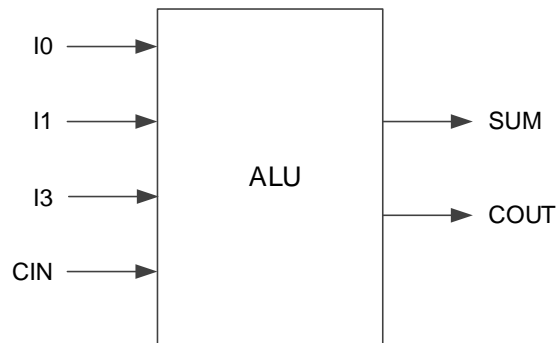
表 3-22 ALU の機能

項目	説明
ADD	加算
SUB	減算
ADDSUB	加算/減算。I3 の値に依存。 ● 1 : 加算 ● 0 : 減算
CUP	加算カウンタ
CDN	減算カウンタ
CUPCDN	加/減算カウンタ。I3 の値に依存。 ● 1 : 加算カウンタ

項目	説明
	● 0 : 減算カウンタ
GE	大なりイコール比較器
NE	不等比較器
LE	小なりイコール比較器

ポート図

図 3-9 ALU のポート図



注記：

GW5AT デバイスの CIN のソースは、前の ALU の COUT、または論理または定数です。

ポートの説明

表 3-23 ALU のポートの説明

ポート	Input/Output	説明
I0	入力	データ入力信号
I1	入力	データ入力信号
I3	入力	ADDSUBの加算/減算選択またはCUPCDNの加算/減算カウンタ選択に使用されるデータ選択信号
CIN	入力	データキャリー入力信号
COUT	出力	データキャリー出力信号
SUM	出力	データ出力信号

パラメータの説明

表 3-24 ALU のパラメータの説明

パラメータ	範囲	デフォルト	説明
ALU_MODE	0,1,2,3,4,5,6,7,8	0	Select the function of

パラメータ	範囲	デフォルト	説明
			arithmetic. ● 0: ADD ● 1: SUB ● 2: ADDSUB ● 3: NE ● 4: GE ● 5: LE ● 6: CUP ● 7: CDN ● 8: CUPCDN

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```

    ALU instName (
        .I0(I0),
        .I1(I1),
        .I3(I3),
        .CIN(CIN),
        .COUT(COUT),
        .SUM(SUM)
    );
    defparam instName.ALU_MODE=1;

```

VHDL でのインスタンス化 :

```

COMPONENT ALU
    GENERIC (ALU_MODE:integer:=0);
    PORT(
        COUT:OUT std_logic;
        SUM:OUT std_logic;
        I0:IN std_logic;
        I1:IN std_logic;
        I3:IN std_logic;
        CIN:IN std_logic
    );
END COMPONENT;

uut:ALU
    GENERIC MAP(ALU_MODE=>1)

```

```

PORT MAP (
    COUT=>COUT,
    SUM=>SUM,
    I0=>I0,
    I1=>I1,
    I3=>I3,
    CIN=>CIN
);

```

3.4 FF

フリップフロップは、シーケンシャル回路で一般的に使用される基本的なコンポーネントです。FPGA の内部シーケンシャルロジックは、FF 構造によって実現できます。一般的に使用される FF には、DFFSE、DFFRE、DFFPE、DFFCE などがあります。これらの FF は、リセットモードなどにおいて異なります。

FF に関するプリミティブは 4 つあります(表 3-25)。

表 3-25 FF プリミティブ

プリミティブ	説明
DFFSE	クロックイネーブルおよび同期セット付き D フリップフロップ
DFFRE	クロックイネーブルおよび同期リセット付き D フリップフロップ
DFFPE	クロックイネーブルおよび非同期セット付き D フリップフロップ
DFFCE	クロックイネーブルおよび非同期クリア付き D フリップフロップ

配置ルール

表 3-26 FF のタイプ

番号	タイプ 1	タイプ 2
1	DFFSE	DFFRE
2	DFFPE	DFFCE

- 同じタイプの DL の場合、同じ CLS の 2 つの FF に配置できます。データ入力ピン以外のすべての入力は共線でなければなりません。
- 異なるタイプの DFF の場合、の同じ番号の 2 つのタイプの DFF を同じ CLS の 2 つの FF に配置できます。データ入力ピン以外のすべての入力は共線でなければなりません。
- 同じ CLS の同じまたは異なる位置に DFF と ALU を制約することができます。
- 同じ CLS の同じまたは異なる位置に DFF と LUT を制約することがで

きます。

注記：

共線とは、同じ **net** ということです。インバータの前後の 2 つの **net** は共線ではなく、同じ **CLS** に配置できません。

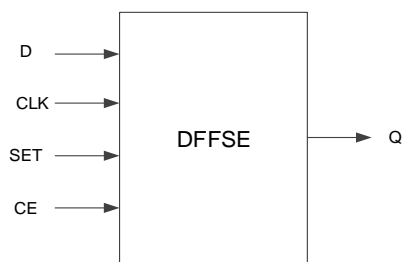
3.4.1 DFFSE

プリミティブの紹介

DFFSE(D Flip-Flop with Clock Enable and Synchronous Set)は立ち上がりエッジでトリガする D フリップフロップで、同期セットとクロックイネーブル機能を備えています。

ポート図

図 3-10 DFFSE のポート図



ポートの説明

表 3-27 DFFSE のポートの説明

ポート	I/O	説明
D	入力	データ入力信号
CLK	入力	クロック入力信号
SET	入力	同期セット信号、アクティブ High
CE	入力	クロックイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 3-28 DFFSE のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	1'b1	1'b1	DFFSE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化：

```
DFFSE instName (
    .D(D),
```

```

        .CLK(CLK),
        .SET(SET),
        .CE(CE),
        .Q(Q)
    );
    defparam instName.INIT=1'b1;
VHDL でのインスタンス化 :
    COMPONENT DFFSE
        GENERIC (INIT:bit:='1');
        PORT(
            Q:OUT std_logic;
            D:IN std_logic;
            CLK:IN std_logic;
            SET:IN std_logic;
            CE:IN std_logic
        );
    END COMPONENT;
    uut:DFFSE
        GENERIC MAP(INIT=>'1')
        PORT MAP (
            Q=>Q,
            D=>D,
            CLK=>CLK,
            SET=>SET,
            CE=>CE
        );

```

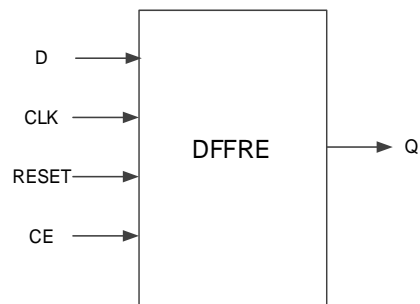
3.4.2 DFFRE

プリミティブの紹介

DFFRE(D Flip-Flop with Clock Enable and Synchronous Reset)は立ち上がりエッジでトリガする D フリップフロップで、同期リセットとクロックイネーブル機能を備えています。

ポート図

図 3-11 DFFRE のポート図



ポートの説明

表 3-29 DFFRE のポートの説明

ポート	I/O	説明
D	入力	データ入力信号
CLK	入力	クロック入力信号
RESET	入力	同期リセット信号、アクティブ High
CE	入力	クロックイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 3-30 DFFRE のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	1'b0	1'b0	DFFRE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```

DFFRE instName (
    .D(D),
    .CLK(CLK),
    .RESET(RESET),
    .CE(CE),
    .Q(Q)
);
defparam instName.INIT=1'b0;

```

VHDL でのインスタンス化 :

```

COMPONENT DFFRE
  GENERIC (INIT:bit:='0');
  PORT(
    Q:OUT std_logic;
    D:IN std_logic;
    CLK:IN std_logic;
    RESET:IN std_logic;
    CE:IN std_logic
  );
END COMPONENT;

uut:DFFRE
  GENERIC MAP(INIT=>'0')
  PORT MAP (
    Q=>Q,
    D=>D,
    CLK=>CLK,
    RESET=>RESET,
    CE=>CE
  );

```

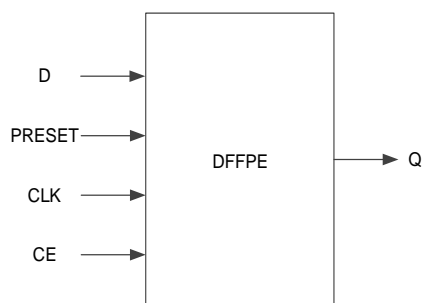
3.4.3 DFFPE

プリミティブの紹介

DFFPE(D Flip-Flop with Clock Enable and Asynchronous Preset)は立ち上がりエッジでトリガする D フリップフロップで、非同期セットとクロックイネーブル機能を備えています。

ポート図

図 3-12 DFFPE のポート図



ポートの説明

表 3-31 DFFPE のポートの説明

ポート	I/O	説明
D	入力	データ入力信号
CLK	入力	クロック入力信号
PRESET	入力	非同期セット信号、アクティブ High
CE	入力	クロックイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 3-32 DFFPE のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	1'b1	1'b1	DFFPE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
DFFPE instName (
    .D(D),
    .CLK(CLK),
    .PRESET(PRESET),
    .CE(CE),
    .Q(Q)
);
defparam instName.INIT=1'b1;
```

VHDL でのインスタンス化 :

```
COMPONENT DFFPE
    GENERIC (INIT:bit:= '1');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        CLK:IN std_logic;
        PRESET:IN std_logic;
        CE:IN std_logic
    );
END COMPONENT;
```

```
uut:DFFPE
  GENERIC MAP(INIT=>'1')
  PORT MAP (
    Q=>Q,
    D=>D,
    CLK=>CLK,
    PRESET=>PRESET,
    CE=>CE
  );
```

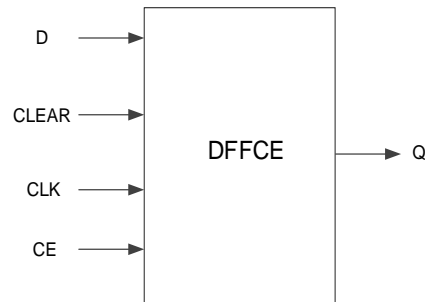

3.4.4 DFFCE

プリミティブの紹介

DFFCE(D Flip-Flop with Clock Enable and Asynchronous Clear)は立ち上がりエッジでトリガする D フリップフロップで、非同期クリアとクロックイネーブル機能を備えています。

ポート図

図 3-13 DFFCE のポート図



ポートの説明

表 3-33 DFFCE のポートの説明

ポート	I/O	説明
D	入力	データ入力信号
CLK	入力	クロック入力信号
CLEAR	入力	非同期クリア信号、アクティブ High
CE	入力	クロックイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 3-34 DFFCE のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	1'b0	1'b0	DFFCE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
DFFCE instName (
    .D(D),
    .CLK(CLK),
    .CLEAR(CLEAR),
    .CE(CE),
```

```
.Q(Q)
);
defparam instName.INIT=1'b0;
```

VHDL でのインスタンス化 :

```
COMPONENT DFFCE
    GENERIC (INIT:bit:='0');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        CLK:IN std_logic;
        CLEAR:IN std_logic;
        CE:IN std_logic
    );
END COMPONENT;
uut:DFFCE
    GENERIC MAP(INIT=>'0')
    PORT MAP (
        Q=>Q,
        D=>D,
        CLK=>CLK,
        CLEAR=>CLEAR,
        CE=>CE
    );
```

3.5 LATCH

ラッチは、1 ビットの情報を保持できる、レベルトリガの回路です。FF に関するプリミティブは 2 つあります(表 3-)。

表 3- LATCH プリミティブ

プリミティブ	説明
DLCE	非同期リセットとラッチイネーブル付きデータラッチ
DLPE	非同期プリセットとラッチイネーブル付きデータラッチ

配置ルール

表 3-35 LATCH のタイプ

番号	タイプ 1	タイプ 2
1	DLCE	DLPE

- 同じタイプの DL の場合、同じ CLS の 2 つの FF に配置できます。データ入力ピン以外のすべての入力は共線であればなりません。
- 異なるタイプの DL の場合、上表の同じ番号の 2 つのタイプを同じ CLS の 2 つの FF に配置できます。データ入力ピン以外のすべての入力は共線であればなりません。
- 同じ CLS の同じまたは異なる位置に DL と ALU を制約できます。
- 同じ CLS の同じまたは異なる位置に DL と LUT を制約できます。

注記：

共線とは、同じ net ということです。インバータの前後の 2 つの net は共線ではなく、同じ CLS に配置できません。

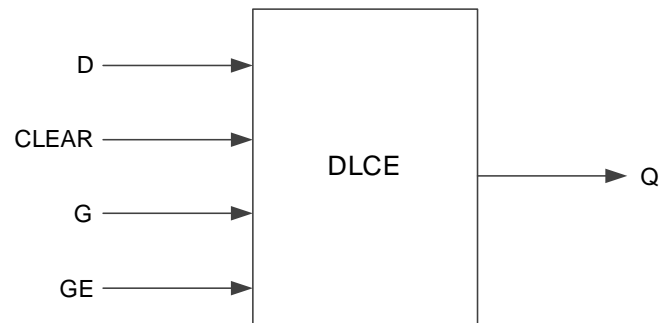
3.5.1 DLCE

プリミティブの紹介

DLCE(Data Latch with Asynchronous Clear and Latch Enable)はイネーブル制御と非同期クリア機能を備えたラッチで、制御信号 G はアクティブ High です。

ポート図

図 3-14 DLCE のポート図



ポートの説明

表 3-36 DLCE のポートの説明

ポート	I/O	説明
D	入力	データ入力信号
CLEAR	入力	非同期クリア信号、アクティブ High
G	入力	データ制御信号、アクティブ High
GE	入力	レベルのイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 3-37 DLCE のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	1'b0	1'b0	DLCE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```

DLCE instName (
    .D(D),
    .CLEAR(CLEAR),
    .G(G),
    .GE(GE),
    .Q(Q)
);
defparam instName.INIT=1'b0;
  
```

VHDL でのインスタンス化 :

```

COMPONENT DLCE
  GENERIC (INIT:bit:='0');
  PORT(
    Q:OUT std_logic;
    D:IN std_logic;
    G:IN std_logic;
    GE:IN std_logic;
    CLEAR:IN std_logic
  );
END COMPONENT;
uut:DLCE
  GENERIC MAP(INIT=>'0')
  PORT MAP (
    Q=>Q,
    D=>D,
    G=>G,
    GE=>GE,
    CLEAR=>CLEAR
  );

```

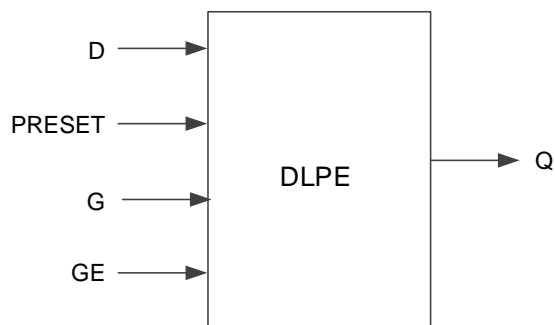
3.5.2 DLPE

プリミティブの紹介

DLPE(Data Latch with Asynchronous Preset and Latch Enable)はイネーブル制御とセット機能を備えたラッチで、制御信号 **G** はアクティブ High です。

ポート図

図 3-15 DLPE のポート図



ポートの説明

表 3-38 DLPE のポートの説明

ポート	I/O	説明
D	入力	データ入力信号
PRESET	入力	非同期セット信号、アクティブ High
G	入力	データ制御信号、アクティブ High
GE	入力	レベルのイネーブル信号
Q	出力	データ出力信号

パラメータの説明

表 3-39 DLPE のパラメータの説明

パラメータ	範囲	デフォルト	説明
INIT	1'b1	1'b1	DLPE の初期値

プリミティブのインスタンス化

Verilog でのインスタンス化 :

```
DLPE instName (
    .D(D),
    .PRESET(PRESET),
    .G(G),
    .GE(GE),
    .Q(Q)
);
defparam instName.INIT=1'b1;
```

VHDL でのインスタンス化 :

```
COMPONENT DLPE
    GENERIC (INIT:bit:='1');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        G:IN std_logic;
        GE:IN std_logic;
        PRESET:IN std_logic
    );
END COMPONENT;
```

```
uut:DLPE
    GENERIC MAP(INIT=>'1')
    PORT MAP (
        Q=>Q,
        D=>D,
        G=>G,
        GE=>GE
        PRESET =>PRESET
    );
```

3.6 SSRAM

SSRAM プリミティブについては、『Arora V BSRAM & SSRAM ユーザーガイド([UG300](#))』を参照してください。

