

Gowin コンフィギュラブル機能ユニット (CFU)

ユーザーガイド

著作権について（2022）

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、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。本文書における全ての情報は、予備的情報として取り扱われなければなりません。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
2016/05/17	1.05J	初版。
2016/07/15	1.06J	図面を更新。
2016/08/02	1.07J	GW2A シリーズ FPGA 製品へのサポートを追加。
2016/10/27	1.08J	GW2AR シリーズ FPGA 製品へのサポートを追加。
2020/12/17	1.09J	CFU の説明を更新。
2021/06/21	1.1J	<ul style="list-style-type: none">● GW1N-2B、GW1N-1P5、GW1N-1P5B、GW1NR-2B デバイスのサポートを追加。● FF および LATCH の INIT の値の範囲を更新。
2022/01/24	1.1.1J	コード例のフォーマットを微調整。

目次

目次.....	i
図一覧.....	iii
表一覧.....	v
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.....	5
2.2 CRU.....	6
3 CFU プリミティブ	7
3.1 LUT.....	7
3.1.1 LUT1.....	7
3.1.2 LUT2.....	8
3.1.3 LUT3.....	10
3.1.4 LUT4.....	12
3.1.5 Wide LUT	14
3.2 MUX.....	17
3.2.1 MUX2	18
3.2.2 MUX4	19
3.2.3 Wide MUX	21
3.3 ALU.....	24
3.4 FF	26
3.4.1 DFF.....	28
3.4.2 DFFE	29
3.4.3 DFFS	31
3.4.4 DFFSE.....	32

3.4.5 DFFR	34
3.4.6 DFFRE	35
3.4.7 DFFP	37
3.4.8 DFFPE	39
3.4.9 DFFC	40
3.4.10 DFFCE	42
3.4.11 DFFN	44
3.4.12 DFFNE	45
3.4.13 DFFNS	47
3.4.14 DFFNSE	48
3.4.15 DFFNR	50
3.4.16 DFFNRE	51
3.4.17 DFFNP	53
3.4.18 DFFNPE	55
3.4.19 DFFNC	56
3.4.20 DFFNCE	58
3.5 LATCH	59
3.5.1 DL	60
3.5.2 DLE	62
3.5.3 DLC	64
3.5.4 DLCE	65
3.5.5 DLP	67
3.5.6 DLPE	68
3.5.7 DLN	70
3.5.8 DLNE	72
3.5.9 DLNC	73
3.5.10 DLNCE	75
3.5.11 DLNP	76
3.5.12 DLNPE	78
3.6 SSRAM	80

図一覧

図 2-1 コンフィギャラブル機能ユニットの構造	4
図 2-2 CFU におけるレジスタの説明図.....	5
図 3-1 LUT1 のポート図.....	7
図 3-2 LUT2 のポート図.....	9
図 3-3 LUT3 のポート図.....	10
図 3-4 LUT4 のポート図.....	12
図 3-5 LUT5 のポート図.....	15
図 3-6 MUX2 のポート図	18
図 3-7 MUX4 のポート図	19
図 3-8 MUX8 のポート図	21
図 3-9 ALU ポートの説明図	24
図 3-10 DFF ポートの説明図	28
図 3-11 DFFE のポート図	29
図 3-12 DFFS ポートの説明図.....	31
図 3-13 DFFSE ポートの説明図	32
図 3-14 DFFR ポートの説明図.....	34
図 3-15 DFFRE ポートの説明図	36
図 3-16 DFFP ポートの説明図.....	37
図 3-17 DFFPE ポートの説明図	39
図 3-18 DFFC ポートの説明図.....	41
図 3-19 DFFCE ポートの説明図	42
図 3-20 DFFN ポートの説明図.....	44
図 3-21 DFFNE のポート図	45
図 3-22 DFFNS ポートの説明図	47
図 3-23 DFFNSE ポートの説明図.....	48
図 3-24 DFFNR ポートの説明図	50
図 3-25 DFFNRE ポートの説明図.....	52
図 3-26 DFFNP ポートの説明図	53
図 3-27 DFFNPE ポートの説明図.....	55

図 3-28 DFFNC ポートの説明図	56
図 3-29 DFFNCE ポートの説明図	58
図 3-30 DL ポートの説明図	61
図 3-31 DLE ポートの説明図	62
図 3-32 DLC ポートの説明図	64
図 3-33 DLCE ポートの説明図	65
図 3-34 DLP ポートの説明図	67
図 3-35 DLPE ポートの説明図	69
図 3-36 DLN ポートの説明図	70
図 3-37 DLNE ポートの説明図	72
図 3-38 DLNC ポートの説明図	73
図 3-39 DLNCE ポートの説明図	75
図 3-40 DLNP ポートの説明図	77
図 3-41 DLNPE ポートの説明図	78

表一覧

表 1-1 用語、略語	1
表 2-1 CFU におけるレジスタモジュール信号の説明	5
表 3-1 LUT1 のポート図.....	7
表 3-2 LUT1 のパラメータの説明	7
表 3-3 LUT1 の真理値表.....	8
表 3-4 LUT2 のポートの説明	9
表 3-5 LUT2 のパラメータの説明	9
表 3-6 LUT2 の真理値表.....	9
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	11
表 3-9 LUT3 の真理値表.....	11
表 3-10 LUT4 のポートの説明	12
表 3-11 LUT4 のパラメータの説明	13
表 3-12 LUT4 の真理値表.....	13
表 3-13 LUT5 のポートの説明	15
表 3-14 LUT5 のパラメータの説明	15
表 3-15 LUT5 の真理値表.....	16
表 3-16 MUX2 のポートの説明	18
表 3-17 MUX2 の真理値表	18
表 3-18 MUX4 のポートの説明	19
表 3-19 MUX4 の真理値表	20
表 3-20 MUX8 のポートの説明	22
表 3-21 MUX8 の真理値表	22
表 3-22 ALU の機能.....	24
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	25
表 3-25 FF プリミティブ	26
表 3-26 FF のタイプ.....	27
表 3-27 DFF のポートの説明	28

表 3-28 DFF のパラメータの説明	28
表 3-29 DFFE のポートの説明	30
表 3-30 DFFE のパラメータの説明	30
表 3-31 DFFS のポートの説明	31
表 3-32 DFFS のパラメータの説明	31
表 3-33 DFFSE のポートの説明	33
表 3-34 DFFSE のパラメータの説明	33
表 3-35 DFFR のポートの説明	34
表 3-36 DFFR のパラメータの説明	34
表 3-37 DFFRE のポートの説明	36
表 3-38 DFFRE のパラメータの説明	36
表 3-39 DFFP のポートの説明	37
表 3-40 DFFP のパラメータの説明	38
表 3-41 DFFPE のポートの説明	39
表 3-42 DFFPE のパラメータの説明	39
表 3-43 DFFC のポートの説明	41
表 3-44 DFFC のパラメータの説明	41
表 3-45 DFFCE のポートの説明	42
表 3-46 DFFCE のパラメータの説明	43
表 3-47 DFFN のポートの説明	44
表 3-48 DFFN のパラメータの説明	44
表 3-49 DFFNE のポートの説明	45
表 3-50 DFFNE のパラメータの説明	46
表 3-51 DFFNS のポートの説明	47
表 3-52 DFFNS のパラメータの説明	47
表 3-53 DFFNSE のポートの説明	48
表 3-54 DFFNSE のパラメータの説明	49
表 3-55 DFFNR のポートの説明	50
表 3-56 DFFNR のパラメータの説明	50
表 3-57 DFFNRE のポートの説明	52
表 3-58 DFFNRE のパラメータの説明	52
表 3-59 DFFNP のポートの説明	53
表 3-60 DFFNP のパラメータの説明	54
表 3-61 DFFNPE のポートの説明	55
表 3-62 DFFNPE のパラメータの説明	55
表 3-63 DFFNC のポートの説明	57
表 3-64 DFFNC のパラメータの説明	57

表 3-65 DFFNCE のポートの説明.....	58
表 3-66 DFFNCE のパラメータの説明	58
表 3-67 LATCH プリミティブ	60
表 3-68 LATCH のタイプ	60
表 3-69 DL のポートの説明.....	61
表 3-70 DL のパラメータの説明	61
表 3-71 DLE のポートの説明	62
表 3-72 DLE のパラメータの説明	63
表 3-73 DLC のポートの説明	64
表 3-74 DLC のパラメータの説明.....	64
表 3-75 DLCE のポートの説明.....	66
表 3-76 DLCE のパラメータの説明	66
表 3-77 DLP のポートの説明	67
表 3-78 DLP のパラメータの説明	67
表 3-79 DLPE のポートの説明.....	69
表 3-80 DLPE のパラメータの説明.....	69
表 3-81 DLN のポートの説明	70
表 3-82 DLN のパラメータの説明.....	71
表 3-83 DLNE のポートの説明.....	72
表 3-84 DLNE のパラメータの説明	72
表 3-85 DLNC のポートの説明	73
表 3-86 DLNC のパラメータの説明	74
表 3-87 DLNCE のポートの説明.....	75
表 3-88 DLNCE のパラメータの説明.....	75
表 3-89 DLNP のポートの説明.....	77
表 3-90 DLNP のパラメータの説明	77
表 3-91 DLNPE のポートの説明	78
表 3-92 DLNPE のパラメータの説明	79

1 本マニュアルについて

1.1 マニュアル内容

このマニュアルでは、主に 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 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：

- GW2A シリーズ FPGA 製品データシート ([DS102](#))
- GW1N シリーズ FPGA 製品データシート([DS100](#))
- GW2AR シリーズ FPGA 製品データシート([DS226](#))
- Gowin BSRAM & SSRAM ユーザーガイド ([UG285](#))

1.3 用語、略語

本マニュアルで使用される用語、略語、及びその意味については、表 1-1 を参照してください。

表 1-1 用語、略語

用語、略語	正式名称	意味
CFU	Configurable Function Unit	コンフィギュラブル機能ユニット
CLU	Configurable Logic Unit	コンフィギュラブル論理ユニット
LUT	Look-up Table	ルックアップテーブル
CRU	Configurable Routing Unit	コンフィギュラブル配線ユニット
CLS	Configurable Logic Section	コンフィギュラブル論理セクション
SSRAM	Shadow Static Random Access Memory	分散 SRAM
BSRAM	Block Static Random Access Memory	ブロック SRAM
REG	Register	レジスタ

用語、略語	正式名称	意味
MUX2	Multiplexer 2:1	マルチプレクサ 2 : 1
ALU	Arithmetic Logic Unit	演算論理ユニット
DFF	D Flip Flop	D フリップフロップ
DL	Data Latch	データラッチ

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。

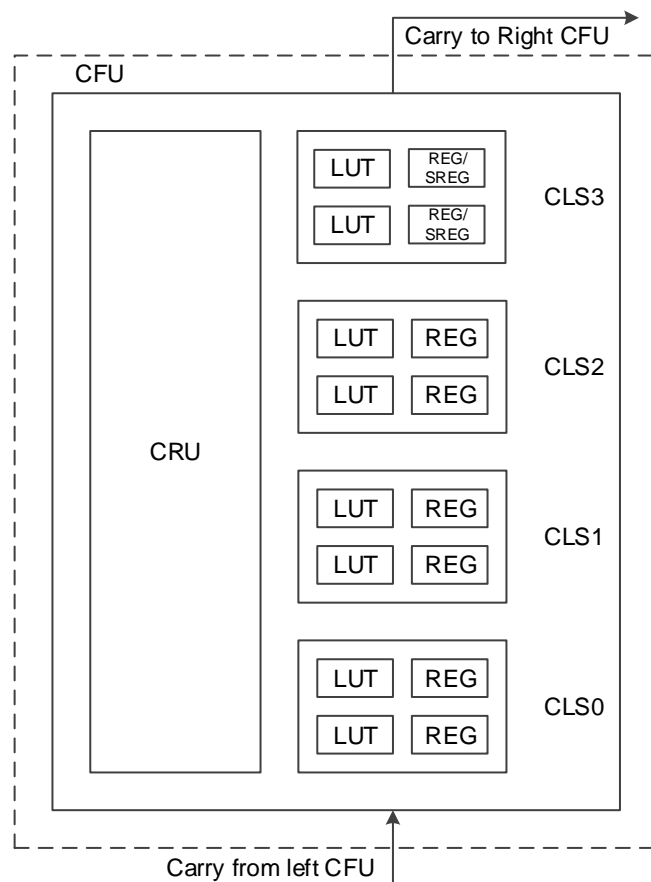
Web サイト : www.gowinsemi.com/ja

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) とコンフィギュラブル論理ユニット (CLU) は、Gowin の FPGA 製品の基本的な構成要素です。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、3 つの CLS にはそれぞれ 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、もう 1 つの CLS には 2 つの 4 入力 LUT のみが含まれます(図 2-1)。CLU 内の CLS は、スタティック RAM として構成することはできませんが、ベーシック・ルックアップテーブル、演算ロジック、及び ROM として構成することはできます。CFU 内の CLS は、アプリケーションシナリオに応じて、ベーシック・ルックアップテーブル、演算ロジックユニット、スタティック RAM、および ROM の 4 つの動作モードに構成できます。このマニュアルでは、CFU について説明します。

図 2-1 コンフィギャラブル機能ユニットの構造



注記：

- SREG を実装するには専用のパッチが必要です。詳細は、Gowin のテクニカル・サポートにお問い合わせください。
- 現在、GW1N-2、GW1N-1P5、GW1N-2B、GW1N-1P5B、GW1NR-2、GW1NR-2B デバイスのみが 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、ベーシック・ルックアップテーブルモード、演算ロジックモード、及びメモリモードをサポートします：

- ベーシック・ルックアップテーブルモード

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、CLU は、以下に示すように、LUT5/LUT6/LUT7/LUT8 などを実装できます。

- 1 つのコンフィギャラブル論理セクションは、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 つのコンフィギャラブル論理セクションは、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 つのコンフィギャラブル論理セクションは、1 つの 7 入力ルックアップテーブル (LUT7) を形成できます。

アップテーブル(LUT7)を形成できます。

- 8つのコンフィギャラブル論理セクションは、1つの8入力ルックアップテーブル(LUT8)を形成できます。

● 演算ロジックモード

キャリーチェーンを利用することにより、LUTは演算ロジック (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ビットのスタティックSRAM (SSRAM) またはROM (ROM16) を形成できます。

2.1.2 REG

コンフィギャラブル論理セクション (CLS0~CLS2)にはそれぞれ、図 2-2 に示す通り、2つのレジスタ(REG)が含まれています。

図 2-2 CFU におけるレジスタの説明図

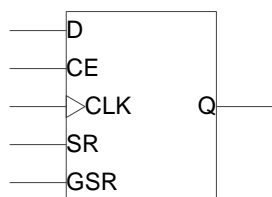


表 2-1 CFU におけるレジスタモジュール信号の説明

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 に設定できる 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設定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 は独立しています。
- [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

コンフィギャラブル配線ユニット (CRU) の主な機能は、下記のとおりです：

- 入力選択機能：CFU の入力信号の入力ソースを選択します。
- 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で CFU の入力と出力接続を実現します。

3CFU プリミティブ

3.1 LUT

LUT には、LUT1、LUT2、LUT3、および LUT4 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。

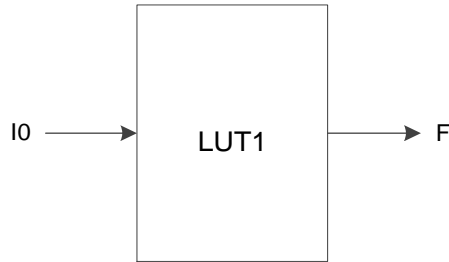
3.1.1 LUT1

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 のポート図

ポート	I/O	説明
I0	入力	データ入力
F	出力	データ出力

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
    );
```

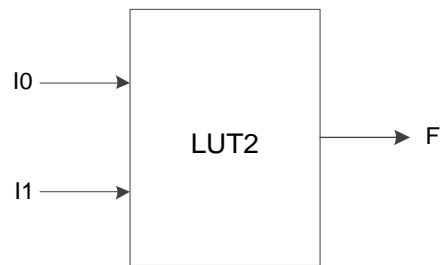
3.1.2 LUT2

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
I1	入力	データ入力
F	出力	データ出力

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
```

```
    );
```

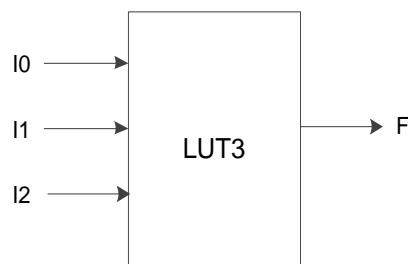
3.1.3 LUT3

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
I1	入力	データ入力

ポート	I/O	説明
I2	入力	データ入力
F	出力	データ出力

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
    );

```

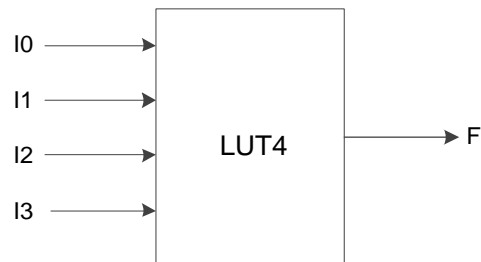
3.1.4 LUT4

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
I1	入力	データ入力
I2	入力	データ入力
I3	入力	データ入力
F	出力	データ出力

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

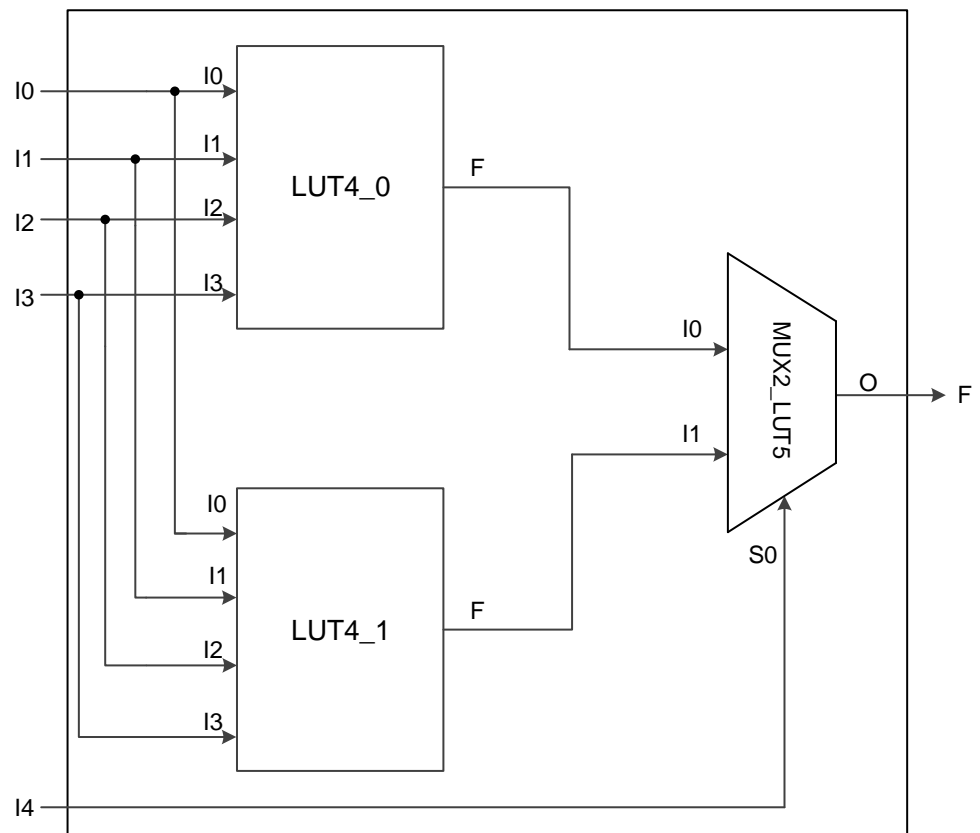
Wide LUT とは、LUT4 と MUX2 によって高レベル LUT を形成することです。GOWIN FPGA は現在 MUX2_LUT5/MUX2_LUT6/MUX2_LUT7/MUX2_LUT8 をサポートします。

高レベル 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
I1	入力	データ入力
I2	入力	データ入力
I3	入力	データ入力
I4	入力	データ入力
F	出力	データ出力

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。

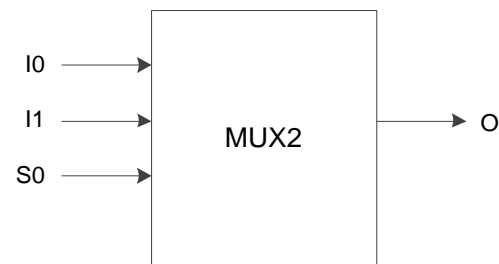
3.2.1 MUX2

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
I1	入力	データ入力
S0	入力	データ選択信号
O	出力	データ出力

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
    );

```

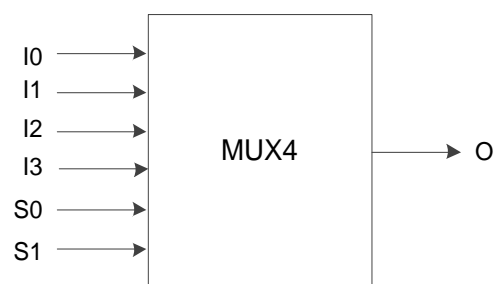
3.2.2 MUX4

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
I0	入力	データ入力
I1	入力	データ入力
I2	入力	データ入力

ポート	I/O	説明
I3	入力	データ入力
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

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

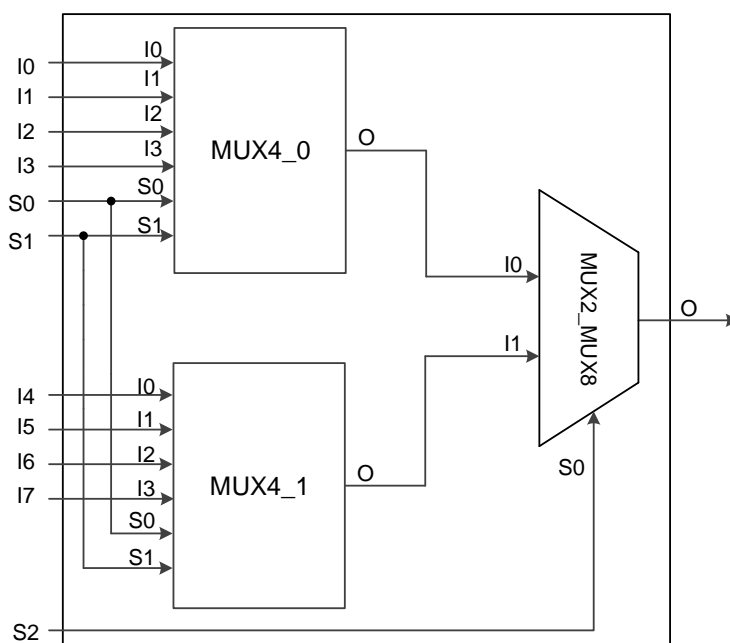
Wide LUT とは、MUX4 と MUX2 によって高レベル MUX を形成することです。GOWIN FPGA は現在 MUX2_MUX8/ MUX2_MUX16/ MUX2_MUX32 をサポートします。

高レベル 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
I1	入力	データ入力
I2	入力	データ入力
I3	入力	データ入力
I4	入力	データ入力
I5	入力	データ入力
I6	入力	データ入力
I7	入力	データ入力
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

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
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

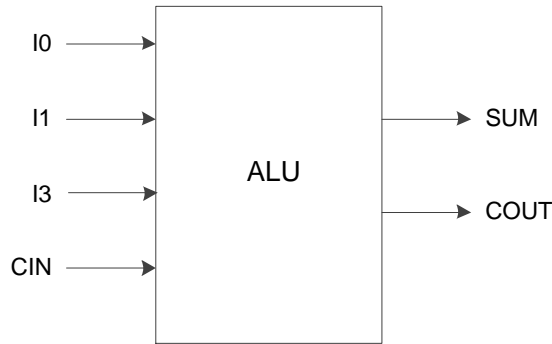
ALU(2-input Arithmetic Logic Unit)は 2 入力演算論理ユニットで、ADD/SUB/ADDSUB などの機能を実現します (表 3-22)。

表 3-22 ALU の機能

項目	説明
ADD	加算
SUB	減算
ADDSUB	加算/減算。I3 が 1 の場合は加算、I3 が 1 の 0 場合は減算です。
CUP	加算カウンタ
CDN	減算カウンタ
CUPCDN	加算/減算カウンタ。I3 が 1 の場合は加算カウンタ、I3 が 1 の 0 場合は減算カウンタです。
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
MULT	乗算器

ポート図

図 3-9 ALU ポートの説明図



ポートの説明

表 3-23 ALU のポートの説明

ポート	Input/Output	説明
-----	--------------	----

ポート	Input/Output	説明
I0	入力	データ入力
I1	入力	データ入力
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

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	0,1,2,3,4,5,6,7,8,9	0	Select the function of arithmetic. 0:ADD; 1:SUB; 2:ADDSUB; 3:NE; 4:GE; 5:LE; 6:CUP; 7:CDN; 8:CUPCDN; 9:MULT

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、タイミング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 の内部タイミングロジックは、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、DFF、DFFE、DFFS、DFFSE などがあります。これらの FF は、リセットモードやトリガー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 20 個あり、表 3-25 に示すとおりです。

表 3-25 FF プリミティブ

プリミティブ	説明
DFF	D フリップフロップ
DFFE	クロックイネーブル付き D フリップフロップ
DFFS	同期セット付き D フリップフロップ
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
DFFR	同期リセット付き 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

プリミティブ	説明
DFFP	非同期セット付き 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
DFFC	非同期クリア付き 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
DFFN	立ち下がりエッジ D フリップフロップ
DFFNE	立ち下がりクロックイネーブル付き D フリップフロップ
DFFNS	立ち下がりエッジ同期セット付き D フリップフロップ
DFFNSE	立ち下がりエッジクロックイネーブル付き、同期セット D フリップフロップ
DFFNR	立ち下がりエッジ同期リセット付き D フリップフロップ
DFFNRE	立ち下がりエッジ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
DFFNP	立ち下がりエッジ非同期リセット付き D フリップフロップ
DFFNPE	立ち下がりエッジ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
DFFNC	立ち下がりエッジ非同期クリア付き D フリップフロップ
DFFNCE	立ち下がりエッジ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	DFFS	DFFR
2	DFFSE	DFFRE
3	DFFP	DFFC
4	DFFPE	DFFCE
5	DFFNS	DFFNR
6	DFFNSE	DFFNRE
7	DFFNP	DFFNC
8	DFFNPE	DFFNCE

- 同じタイプの DFF 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、表 3-26 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。

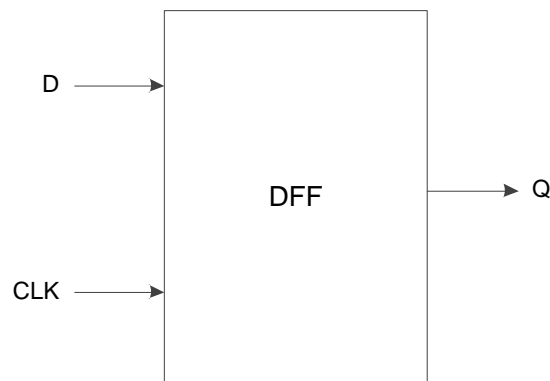
3.4.1 DFF

プリミティブの紹介

DFF(D Flip-Flop)は最もシンプルでよく使われるフリップフロップで、通常は信号サンプリングと処理に使用され、立ち上がりエッジでトリガする **D** フリップフロップです。

ポート図

図 3-10 DFF ポートの説明図



ポートの説明

表 3-27 DFF のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力
Q	出力	データ出力

パラメータの説明

表 3-28 DFF 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	DFF 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：

```

DFF instName (
    .D(D),
    .CLK(CLK),
    .Q(Q)
  )
  
```

```

);
defparam instName.INIT=1'b0;
VHDLでのインスタンス化：
COMPONENT DFF
    GENERIC (INIT:bit:= '0');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        CLK:IN std_logic

    );
END COMPONENT;
uut:DFF
    GENERIC MAP(INIT=>'0')
    PORT MAP (
        Q=>Q,
        D=>D,
        CLK=>CLK
    );

```

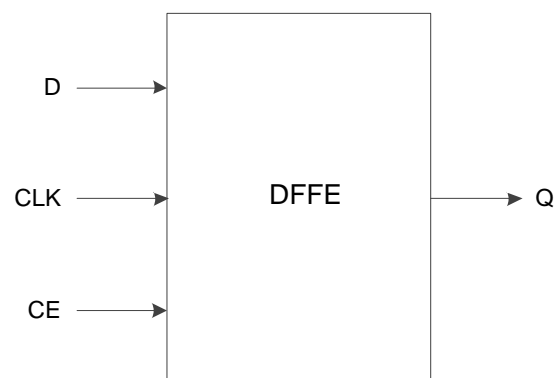
3.4.2 DFFE

プリミティブの紹介

DFFE(D Flip-Flop with Clock Enable)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、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 DFFE のポート図



ポートの説明

表 3-29 DFFE のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力
CE	入力	クロックイネーブル信号
Q	出力	データ出力

パラメータの説明

表 3-30 DFFE 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	DFFE 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 :

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

VHDL でのインスタンス化 :

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

 uut:DFFE
    GENERIC MAP(INIT=>'0')
    PORT MAP (
```



```

        Q=>Q,
        D=>D,
        CLK=>CLK,
        CE=>CE
    );

```

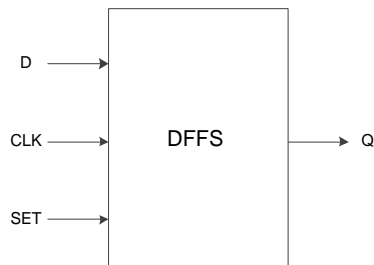
3.4.3 DFFS

プリミティブの紹介

DFFS(D Flip-Flop with 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、同期セット機能を備えています。

ポート図

図 3-12 DFFS ポートの説明図



ポートの説明

表 3-31 DFFS のポートの説明

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

パラメータの説明

表 3-32 DFFS 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	DFFS 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 :

```

DFFS instName (
    .D(D),

```

```

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

```

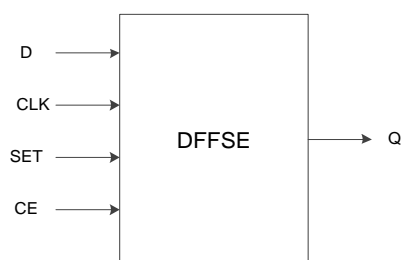
3.4.4 DFFSE

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 3-13 DFFSE ポートの説明図



ポートの説明

表 3-33 DFFSE のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力
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

パラメータの説明

表 3-34 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
  );

```

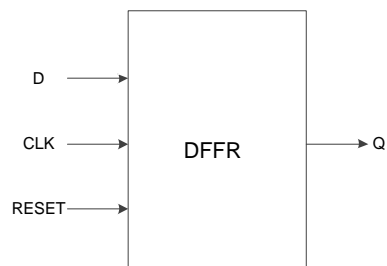
3.4.5 DFFR

プリミティブの紹介

DFFR(D Flip-Flop with 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、同期リセット機能を備えています。

ポート図

図 3-14 DFFR ポートの説明図



ポートの説明

表 3-35 DFFR のポートの説明

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

パラメータの説明

表 3-36 DFFR 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	DFFR 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 :

```
DFFR instName (
    .D(D),
    .CLK(CLK),
    .RESET(RESET),
    .Q(q)
);
defparam instName.INIT=1'b0;
```

VHDL でのインスタンス化 :

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

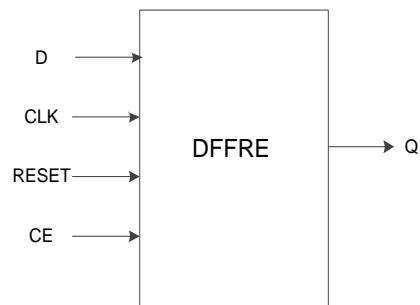
3.4.6 DFFRE

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 3-15 DFFRE ポートの説明図



ポートの説明

表 3-37 DFFRE のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力
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

パラメータの説明

表 3-38 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
    );

```

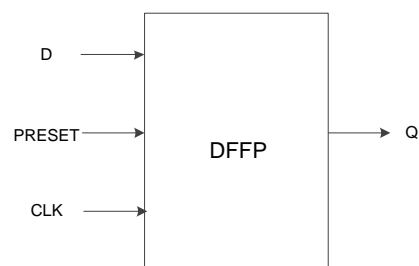
3.4.7 DFFP

プリミティブの紹介

DFFP(D Flip-Flop with 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、非同期セット機能を備えています。

ポート図

図 3-16 DFFP ポートの説明図



ポートの説明

表 3-39 DFFP のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力

ポート	I/O	説明
PRESET	入力	非同期セット信号、アクティブ High
Q	出力	データ出力

パラメータの説明

表 3-40 DFFP 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	DFFP 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 :

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

VHDL でのインスタンス化 :

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

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


);

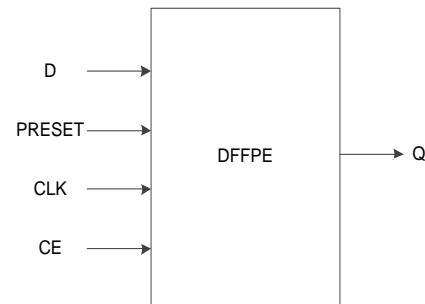
3.4.8 DFFPE

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 3-17 DFFPE ポートの説明図



ポートの説明

表 3-41 DFFPE のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力
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

パラメータの説明

表 3-42 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
        );

```

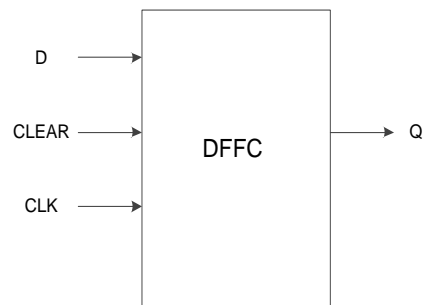
3.4.9 DFFC

プリミティブの紹介

DFFC(D Flip-Flop with 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、非同期クリア機能を備えています。

ポート図

図 3-18 DFFC ポートの説明図



ポートの説明

表 3-43 DFFC のポートの説明

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

パラメータの説明

表 3-44 DFFC 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	DFFC 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 :

```

DFFC instName (
    .D(D),
    .CLK(CLK),
    .CLEAR(CLEAR),
    .Q(Q)
);
defparam instName.INIT=1'b0;

```

VHDL でのインスタンス化 :

```

COMPONENT DFFC
    GENERIC (INIT:bit='0');
    PORT(

```

```

        Q:OUT std_logic;
        D:IN std_logic;
        CLK:IN std_logic;
        CLEAR:IN std_logic

    );
END COMPONENT;
 uut:DFFCE
    GENERIC MAP(INIT=>'0')
    PORT MAP (
        Q=>Q,
        D=>D,
        CLK=>CLK,
        CLEAR=>CLEAR
    );

```

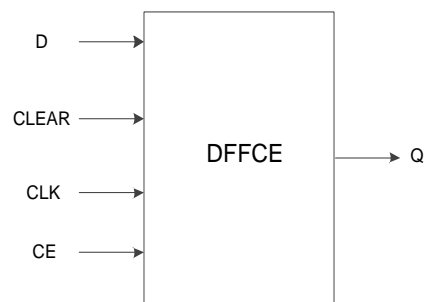
3.4.10 DFFCE

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 3-19 DFFCE ポートの説明図



ポートの説明

表 3-45 DFFCE のポートの説明

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

ポート	I/O	説明
Q	出力	データ出力

パラメータの説明

表 3-46 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

);

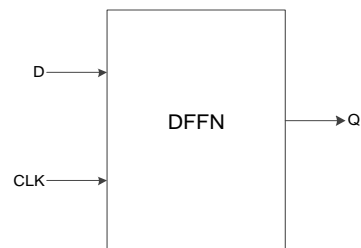
3.4.11 DFFN

プリミティブの紹介

DFFN(D Flip-Flop with Negative-Edge Clock)立ち下がりエッジでトリガする D フリップフロップです。

ポート図

図 3-20 DFFN ポートの説明図



ポートの説明

表 3-47 DFFN のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力
Q	出力	データ出力

パラメータの説明

表 3-48 DFFN 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	DFFN 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 :

```
DFFN instName (
    .D(D),
    .CLK(CLK),
    .Q(Q)
);
defparam instName.INIT=1'b0;
```

VHDL でのインスタンス化 :

```

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

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

```

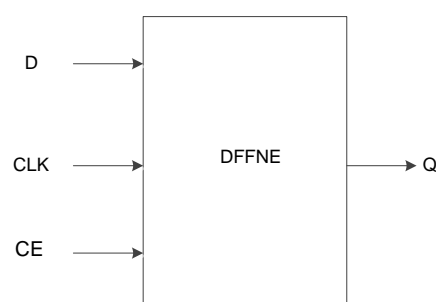
3.4.12 DFFNE

プリミティブの紹介

DFFNE(D Flip-Flop with Negative-Edge Clock and Clock Enable)は立ち下がりエッジでトリガする D フリップフロップで、クロックイネーブル機能を備えています。

ポート図

図 3-21 DFFNE のポート図



ポートの説明

表 3-49 DFFNE のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力

ポート	I/O	説明
CE	入力	クロックイネーブル信号
Q	出力	データ出力

パラメータの説明

表 3-50 DFFNE 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	DFFNE 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 :

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

VHDL でのインスタンス化 :

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

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

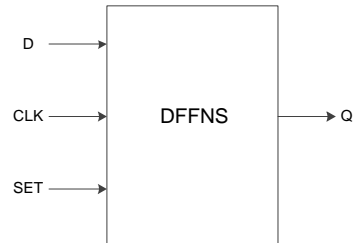

3.4.13 DFFNS

プリミティブの紹介

DFFNS(D Flip-Flop with Negative-Edge Clock and Synchronous Set)は立ち下がりエッジでトリガする D フリップフロップで、同期セット機能を備えています。

ポート図

図 3-22 DFFNS ポートの説明図



ポートの説明

表 3-51 DFFNS のポートの説明

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

パラメータの説明

表 3-52 DFFNS 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	DFFNS 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 :

```

DFFNS instName (
    .D(D),
    .CLK(CLK),
    .SET(SET),
    .Q(Q)
);
defparam instName.INIT=1'b1;

```

VHDL でのインスタンス化 :

```

COMPONENT DFFNS
    GENERIC (INIT:bit:='1');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        CLK:IN std_logic;
        SET:IN std_logic
    );
END COMPONENT;

uut:DFFNS
    GENERIC MAP(INIT=>'1')
    PORT MAP (
        Q=>Q,
        D=>D,
        CLK=>CLK,
        SET=>SET
    );

```

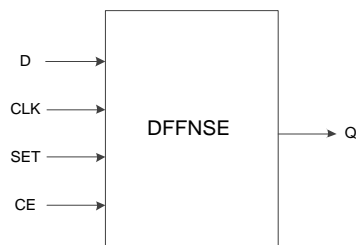
3.4.14 DFFNSE

プリミティブの紹介

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

ポート図

図 3-23 DFFNSE ポートの説明図



ポートの説明

表 3-53 DFFNSE のポートの説明

ポート	I/O	説明
D	入力	データ入力

ポート	I/O	説明
CLK	入力	クロック入力
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

パラメータの説明

表 3-54 DFFNSE 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	DFFNSE 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 :

```
DFFNSE 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 DFFNSE
    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:DFFNSE
    GENERIC MAP(INIT=>'1')
    PORT MAP (
```

```

        Q=>Q,
        D=>D,
        CLK=>CLK,
        SET=>SET,
        CE=>CE
    );

```

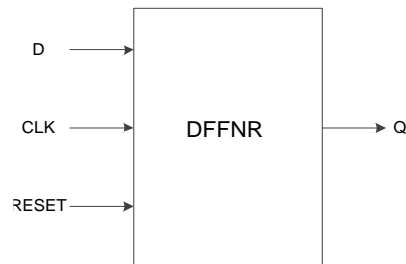
3.4.15 DFFNR

プリミティブの紹介

DFFNR(D Flip-Flop with Negative-Edge Clock and Synchronous Reset) は立ち下がりエッジでトリガする D フリップフロップで、同期リセット機能を備えています。

ポート図

図 3-24 DFFNR ポートの説明図



ポートの説明

表 3-55 DFFNR のポートの説明

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

パラメータの説明

表 3-56 DFFNR 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	DFFNR 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：

```
DFFNR instName (
```

```

        .D(D),
        .CLK(CLK),
        .RESET(RESET),
        .Q(Q)
    );
    defparam instName.INIT=1'b0;
VHDL でのインスタンス化 :
    COMPONENT DFFNR
        GENERIC (INIT:bit:= '0');
        PORT(
            Q:OUT std_logic;
            D:IN std_logic;
            CLK:IN std_logic;
            RESET:IN std_logic
        );
    END COMPONENT;
    uut:DFFNR
        GENERIC MAP(INIT=>'0')
        PORT MAP (
            Q=>Q,
            D=>D,
            CLK=>CLK,
            RESET=>RESET
        );

```

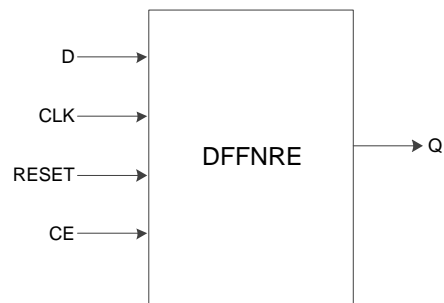
3.4.16 DFFNRE

プリミティブの紹介

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

ポート図

図 3-25 DFFNRE ポートの説明図



ポートの説明

表 3-57 DFFNRE のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力
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

パラメータの説明

表 3-58 DFFNRE 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	DFFNRE 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 :

```

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

```

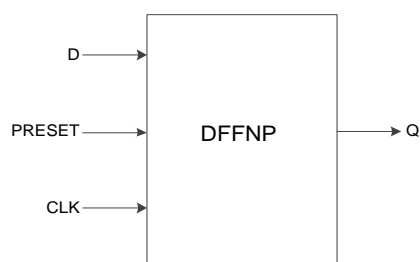
3.4.17 DFFNP

プリミティブの紹介

DFFNP(D Flip-Flop with Negative-Edge Clock and Asynchronous Preset)は立ち下がりエッジでトリガする D フリップフロップで、非同期セット機能を備えています。

ポート図

図 3-26 DFFNP ポートの説明図



ポートの説明

表 3-59 DFFNP のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力

PRESET	入力	非同期セット信号、アクティブ High
Q	出力	データ出力

パラメータの説明

表 3-60 DFFNP 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	DFFNP 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 :

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

VHDL でのインスタンス化 :

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

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

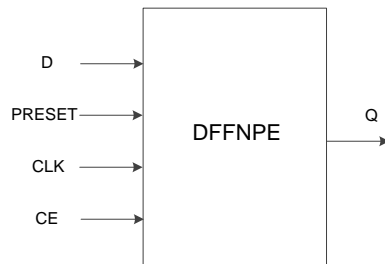

3.4.18 DFFNPE

プリミティブの紹介

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

ポート図

図 3-27 DFFNPE ポートの説明図



ポートの説明

表 3-61 DFFNPE のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力
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

パラメータの説明

表 3-62 DFFNPE 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	DFFNPE 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：

```

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

```

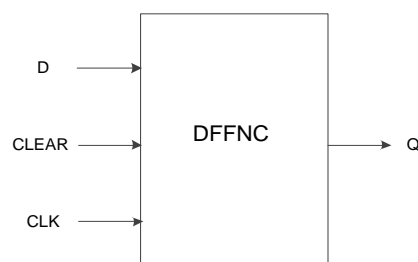
3.4.19 DFFNC

プリミティブの紹介

DFFNC(D Flip-Flop with Negative-Edge Clock and Asynchronous Clear) は立ち下がリエッジでトリガする D フリップフロップで、非同期クリア機能を備えています。

ポート図

図 3-28 DFFNC ポートの説明図



ポートの説明

表 3-63 DFFNC のポートの説明

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

パラメータの説明

表 3-64 DFFNC 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	DFFNC 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 :

```
DFFNC instName (
    .D(D),
    .CLK(CLK),
    .CLEAR(CLEAR),
    .Q(Q)
);
defparam instName.INIT=1'b0;
```

VHDL でのインスタンス化 :

```
COMPONENT DFFNC
    GENERIC (INIT:bit:= '0');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        CLK:IN std_logic;
        CLEAR:IN std_logic
    );
END COMPONENT;

uut:DFFNC
    GENERIC MAP(INIT=>'0')
    PORT MAP (
```

```

        Q=>Q,
        D=>D,
        CLK=>CLK,
        CLEAR=>CLEAR
    );

```

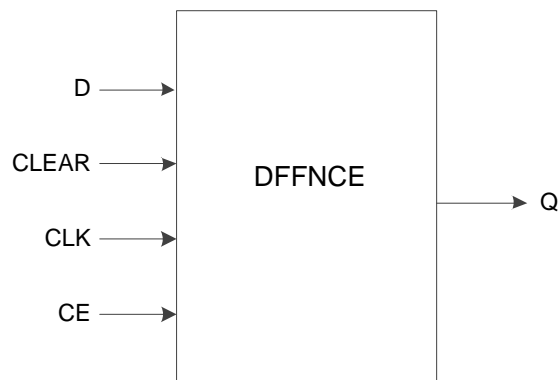
3.4.20 DFFNCE

プリミティブの紹介

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

ポート図

図 3-29 DFFNCE ポートの説明図



ポートの説明

表 3-65 DFFNCE のポートの説明

ポート	I/O	説明
D	入力	データ入力
CLK	入力	クロック入力
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

パラメータの説明

表 3-66 DFFNCE 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	DFFNCE 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 :

```
DFFNCE 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 DFFNCE
    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:DFFNCE
    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。**LATCH**に関するプリミティブは12個あり、表 3-67 に示すとおりです。

表 3-67 LATCH プリミティブ

プリミティブ	説明
DL	データラッチ
DLE	ラッチイネーブル付きデータラッチ
DLC	非同期リセット付きデータラッチ
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
DLP	非同期プリセット付きデータラッチ
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
DLN	アクティブ Low のデータラッチ
DLNE	ラッチイネーブル付きアクティブ Low のデータラッチ
DLNC	非同期リセット付きアクティブ Low のデータラッチ
DLNCE	非同期リセットとラッチイネーブル付きアクティブ Low のデータラッチ
DLNP	非同期プリセット付きアクティブ Low のデータラッチ
DLNPE	非同期プリセットとラッチイネーブル付きアクティブ Low のデータラッチ

配置ルール

表 3-68 LATCH のタイプ

番号	タイプ 1	タイプ 2
1	DLC	DLP
2	DLCE	DLPE
3	DLNC	DLNP
4	DLNCE	DLNPE

- 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共線ではありません。
- 異なるタイプの DL の場合、表 3-68 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共線ではありません。
- 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。

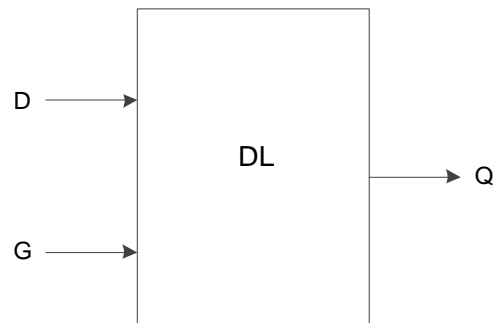
3.5.1 DL

プリミティブの紹介

DL(Data Latch)はそのうち最もシンプルでよく使われるラッチで、制御信号 G はアクティブ High です。

ポート図

図 3-30 DL ポートの説明図



ポートの説明

表 3-69 DL のポートの説明

ポート	I/O	説明
D	入力	データ入力
G	入力	データ制御信号、アクティブ High
Q	出力	データ出力

パラメータの説明

表 3-70 DL 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	DL 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 :

```

DL instName (
    .D(D),
    .G(G),
    .Q(Q)
);
defparam instName.INIT=1'b0;

```

VHDL でのインスタンス化 :

```

COMPONENT DL
    GENERIC (INIT:bit:= '0');
    PORT(
        Q:OUT std_logic;

```

```

        D:IN std_logic;
        G:IN std_logic
    );
END COMPONENT;
uut:DL
    GENERIC MAP(INIT=>'0')
    PORT MAP (
        Q=>Q,
        D=>D,
        G=>G
    );

```

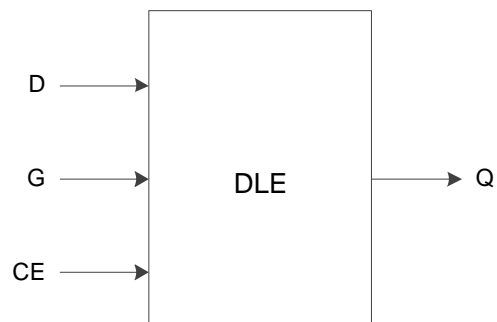
3.5.2 DLE

プリミティブの紹介

DLE(Data Latch with Latch Enable)はイネーブル制御を備えたラッチで、制御信号 **G** はアクティブ High です。

ポート図

図 3-31 DLE ポートの説明図



ポートの説明

表 3-71 DLE のポートの説明

ポート	I/O	説明
D	入力	データ入力
G	入力	データ制御信号、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

パラメータの説明

表 3-72 DLE 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	DLE 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 :

```
DLE instName (
    .D(D),
    .G(G),
    .CE(CE),
    .Q(Q)
);
defparam instName.INIT=1'b0;
```

VHDL でのインスタンス化 :

```
COMPONENT DLE
    GENERIC (INIT:bit:= '0');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        G:IN std_logic;
        CE:IN std_logic
    );
END COMPONENT;

uut:DLE
    GENERIC MAP(INIT=>'0')
    PORT MAP (
        Q=>Q,
        D=>D,
        G=>G,
        CE=>CE
    );
```

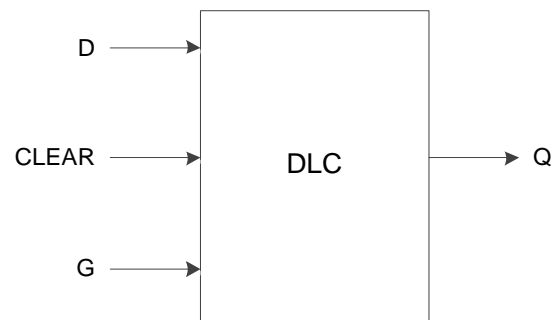
3.5.3 DLC

プリミティブの紹介

DLC(Data Latch with Asynchronous Clear)は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 3-32 DLC ポートの説明図



ポートの説明

表 3-73 DLC のポートの説明

ポート	I/O	説明
D	入力	データ入力
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
Q	出力	データ出力

パラメータの説明

表 3-74 DLC 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	DLC 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：

```

DLC instName (
    .D(D),
    .G(G),
    .CLEAR(CLEAR),
    .Q(Q)
);
  
```

```
defparam instName.INIT=1'b0;
```

VHDL でのインスタンス化 :

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

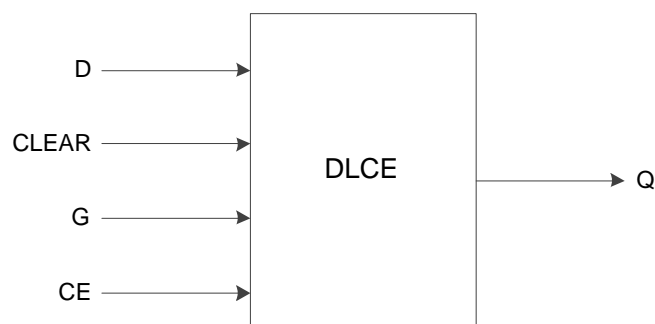
3.5.4 DLCE

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 3-33 DLCE ポートの説明図



ポートの説明

表 3-75 DLCE のポートの説明

ポート	I/O	説明
D	入力	データ入力
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
CE	入力	クロックイネーブル信号
Q	出力	データ出力

パラメータの説明

表 3-76 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),
    .CE(CE),
    .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;
        CE:IN std_logic;
        CLEAR:IN std_logic
    );
END COMPONENT;
```

```

uut:DLCE
  GENERIC MAP(INIT=>'0')
  PORT MAP (
    Q=>Q,
    D=>D,
    G=>G,
    CE=>CE,
    CLEAR=>CLEAR
  );

```

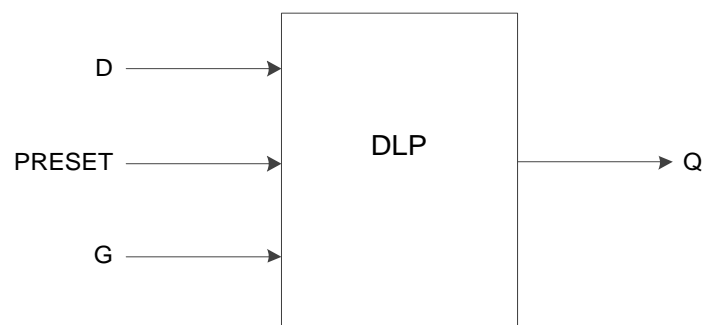
3.5.5 DLP

プリミティブの紹介

DLP(Data Latch with Asynchronous Preset)は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 3-34 DLP ポートの説明図



ポートの説明

表 3-77 DLP のポートの説明

ポート	I/O	説明
D	入力	データ入力
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
Q	出力	データ出力

パラメータの説明

表 3-78 DLP 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	DLP 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 :

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

VHDL でのインスタンス化 :

```
COMPONENT DLP
    GENERIC (INIT:bit:='1');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        G:IN std_logic;
        PRESET:IN std_logic
    );
END COMPONENT;
uut:DLP
    GENERIC MAP(INIT=>'1')
    PORT MAP (
        Q=>Q,
        D=>D,
        G=>G,
        PRESET => PRESET
    );
```

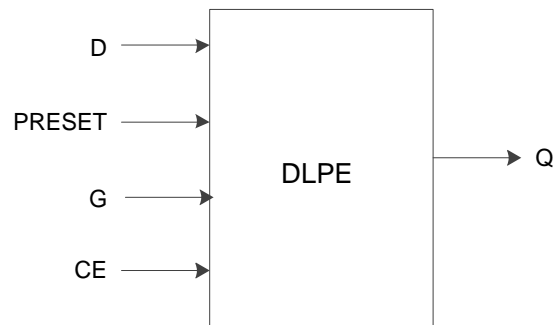
3.5.6 DLPE

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 3-35 DLPE ポートの説明図



ポートの説明

表 3-79 DLPE のポートの説明

ポート	I/O	説明
D	入力	データ入力
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
CE	入力	クロックイネーブル信号
Q	出力	データ出力

パラメータの説明

表 3-80 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),
    .CE(CE),
    .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;
    CE:IN std_logic;
    PRESET:IN std_logic
  );
END COMPONENT;

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

```

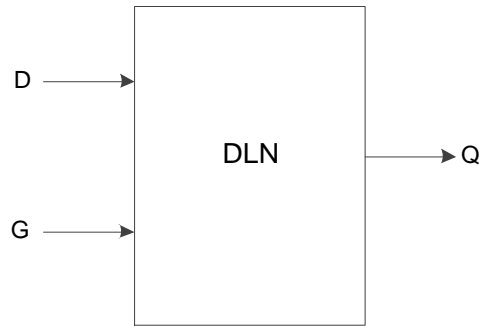
3.5.7 DLN

プリミティブの紹介

DLN(Data Latch with Inverted Gate)は制御信号がアクティブ Low のラッチです。

ポート図

図 3-36 DLN ポートの説明図



ポートの説明

表 3-81 DLN のポートの説明

ポート	I/O	説明
-----	-----	----

D	入力	データ入力
G	入力	データ制御信号、アクティブ Low
Q	出力	データ出力

パラメータの説明

表 3-82 DLN 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	DLN 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 :

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

VHDL でのインスタンス化 :

```
COMPONENT DLN
    GENERIC (INIT:bit:= '0');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        G:IN std_logic
    );
END COMPONENT;

uut:DLN
    GENERIC MAP(INIT=>'0')
    PORT MAP (
        Q=>Q,
        D=>D,
        G=>G
    );
```

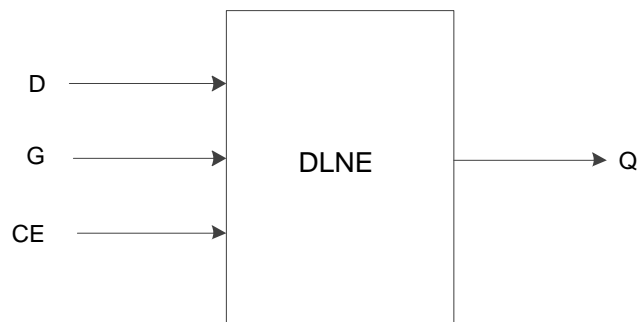
3.5.8 DLNE

プリミティブの紹介

DLNE(Data Latch with Latch Enable and Inverted Gate)はイネーブル制御を備えたラッチで、制御信号 **G** はアクティブ **Low** です。

ポート図

図 3-37 DLNE ポートの説明図



ポートの説明

表 3-83 DLNE のポートの説明

ポート	I/O	説明
D	入力	データ入力
G	入力	データ制御信号、アクティブ Low
CE	入力	クロックイネーブル信号
Q	出力	データ出力

パラメータの説明

表 3-84 DLNE 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	DLNE 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：

```

DLNE instName (
    .D(D),
    .G(G),
    .CE(CE),
    .Q(Q)
);
defparam instName.INIT=1'b0;
  
```

VHDL でのインスタンス化 :

```

COMPONENT DLNE
  GENERIC (INIT:bit:='0');
  PORT(
    Q:OUT std_logic;
    D:IN std_logic;
    G:IN std_logic;
    CE:IN std_logic
  );
END COMPONENT;

uut:DLNE
  GENERIC MAP(INIT=>'0')
  PORT MAP (
    Q=>Q,
    D=>D,
    G=>G,
    CE => CE
  );

```

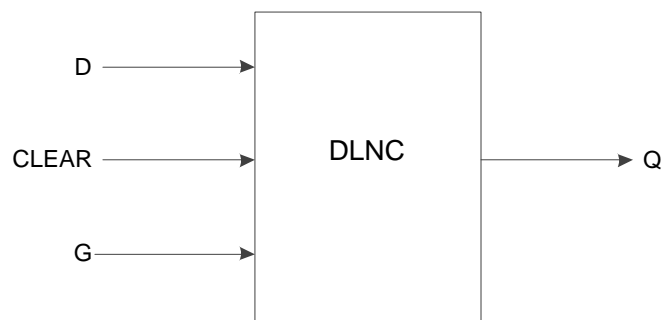
3.5.9 DLNC

プリミティブの紹介

DLNC(Data Latch with Asynchronous Clear and Inverted Gate)は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 **Low** です。

ポート図

図 3-38 DLNC ポートの説明図



ポートの説明

表 3-85 DLNC のポートの説明

ポート	I/O	説明
-----	-----	----

ポート	I/O	説明
D	入力	データ入力
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 Low
Q	出力	データ出力

パラメータの説明

表 3-86 DLNC 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	DLNC 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 :

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

VHDL でのインスタンス化 :

```
COMPONENT DLNC
    GENERIC (INIT:bit:= '0');
    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        G:IN std_logic;
        CLEAR:IN std_logic
    );
END COMPONENT;

uut:DLNC
    GENERIC MAP(INIT=>'0')
    PORT MAP (
        Q=>Q,
        D=>D,
```

```

        G=>G,
        CLEAR => CLEAR
    );

```

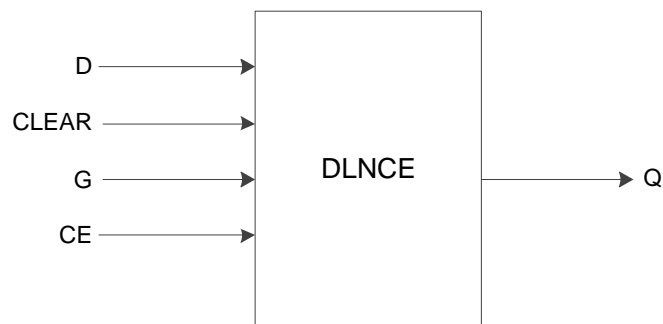
3.5.10 DLNCE

プリミティブの紹介

DLNCE(Data Latch with Asynchronous Clear, Latch Enable, and Inverted Gate)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 **Low** です。

ポート図

図 3-39 DLNCE ポートの説明図



ポートの説明

表 3-87 DLNCE のポートの説明

ポート	I/O	説明
D	入力	データ入力
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 Low
CE	入力	クロックイネーブル信号
Q	出力	データ出力

パラメータの説明

表 3-88 DLNCE 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	DLNCE 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：

```

DLNCE instName (
    .D(D),

```

```

        .CLEAR(CLEAR),
        .G(G),
        .CE(CE),
        .Q(Q)
    );
    defparam instName.INIT=1'b0;
VHDL でのインスタンス化 :
    COMPONENT DLNCE
        GENERIC (INIT:bit:= '0');
        PORT(
            Q:OUT std_logic;
            D:IN std_logic;
            G:IN std_logic;
            CE:IN std_logic;
            CLEAR:IN std_logic
        );
    END COMPONENT;
    uut:DLNCE
        GENERIC MAP(INIT=>'0'
        )
        PORT MAP (
            Q=>Q,
            D=>D,
            G=>G,
            CE=>CE,
            CLEAR=>CLEAR
        );

```

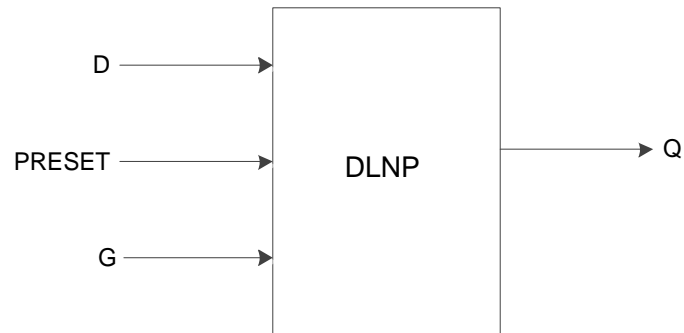
3.5.11 DLNP

プリミティブの紹介

DLNP(Data Latch with Asynchronous Preset and Inverted Gate)はセット機能を備えたラッチで、制御信号 **G** はアクティブ **Low** です。

ポート図

図 3-40 DLNP ポートの説明図



ポートの説明

表 3-89 DLNP のポートの説明

ポート	I/O	説明
D	入力	データ入力
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 Low
Q	出力	データ出力

パラメータの説明

表 3-90 DLNP 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	DLNP 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 :

```

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

```

VHDL でのインスタンス化 :

```

COMPONENT DLNP
    GENERIC (INIT:bit:= '1');

```

```

    PORT(
        Q:OUT std_logic;
        D:IN std_logic;
        G:IN std_logic;
        PRESET:IN std_logic

    );
END COMPONENT;
uut:DLNP
    GENERIC MAP(INIT=>'1')
    PORT MAP (
        Q=>Q,
        D=>D,
        G=>G,
        PRESET => PRESET
    );

```

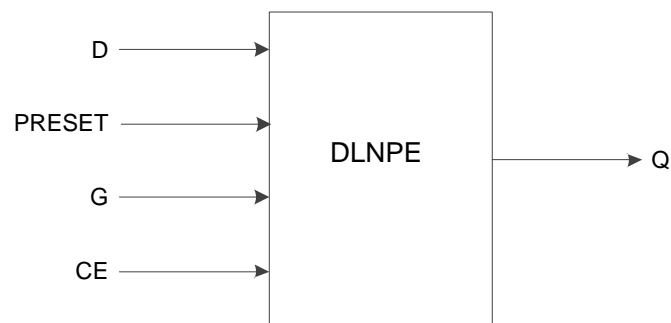
3.5.12 DLNPE

プリミティブの紹介

DLNPE(Data Latch with Asynchronous Preset,Latch Enable and Inverted Gate)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 **Low** です。

ポート図

図 3-41 DLNPE ポートの説明図



ポートの説明

表 3-91 DLNPE のポートの説明

ポート	I/O	説明
D	入力	データ入力
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 Low

ポート	I/O	説明
CE	入力	クロックイネーブル信号
Q	出力	データ出力

パラメータの説明

表 3-92 DLNPE 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	DLNPE 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 :

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

VHDL でのインスタンス化 :

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

uut:DLNPE
    GENERIC MAP(INIT=>'1')
    PORT MAP (
        Q=>Q,
        D=>D,
```

```
        G=>G,  
        CE=>CE,  
        PRESET => PRESET  
    );
```

3.6 SSRAM

SSRAM プリミティブについては、『Gowin BSRAM & SSRAM ユーザーガイド ([UG285](#))』を参照してください。

