



Gowin 可配置功能单元(CFU) 用户指南

UG288-1.1.1,2022-01-24

版权所有 © 2022 广东高云半导体科技股份有限公司

GOWIN高云、Gowin以及高云均为广东高云半导体科技股份有限公司注册商标，本手册中提到的其他任何商标，其所有权利属其拥有者所有。未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本档内容的部分或全部，并不得以任何形式传播。

免责声明

本档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，高云半导体保留修改档中任何内容的权利，恕不另行通知。高云半导体不承诺对这些档进行适时的更新。

版本信息

日期	版本	说明
2016/05/17	1.05	初始版本。
2016/07/15	1.06	标准化插图。
2016/08/02	1.07	适用 GW2A 系列 FPGA 产品。
2016/10/27	1.08	适用 GW2AR 系列 FPGA 产品。
2020/12/17	1.09	更新 CFU 介绍内容。
2021/06/21	1.1	<ul style="list-style-type: none">● 增加器件 GW1N-2B、GW1N-1P5、GW1N-1P5B、GW1NR-2B 支持；● 更新了 FF 和 LATCH 的 INIT 取值范围。
2022/01/24	1.1.1	示例代码格式微调。

目录

目录	i
图目录	iii
表目录	v
1 关于本手册	1
1.1 手册内容	1
1.2 相关文档	1
1.3 术语、缩略语	1
1.4 技术支持与反馈	2
2 CFU 结构	3
2.1 CLS	4
2.1.1 CLS 模式	4
2.1.2 REG	4
2.2 CRU	5
3 CFU 原语	6
3.1 LUT	6
3.1.1 LUT1	6
3.1.2 LUT2	7
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 DFF	27
3.4.2 DFFE	29
3.4.3 DFFS	30
3.4.4 DFFSE	32

3.4.5 DFFR 33

3.4.6 DFFRE 35

3.4.7 DFFP 37

3.4.8 DFFPE 38

3.4.9 DFFC 40

3.4.10 DFFCE 41

3.4.11 DFFN..... 43

3.4.12 DFFNE 44

3.4.13 DFFNS 46

3.4.14 DFFNSE 47

3.4.15 DFFNR..... 49

3.4.16 DFFNRE 50

3.4.17 DFFNP 52

3.4.18 DFFNPE 54

3.4.19 DFFNC..... 55

3.4.20 DFFNCE 57

3.5 LATCH 59

3.5.1 DL 60

3.5.2 DLE 61

3.5.3 DLC..... 63

3.5.4 DLCE 64

3.5.5 DLP 66

3.5.6 DLPE 67

3.5.7 DLN..... 69

3.5.8 DLNE 71

3.5.9 DLNC 72

3.5.10 DLNCE..... 74

3.5.11 DLNP..... 75

3.5.12 DLNPE 77

3.6 SSRAM 79

图目录

图 2-1 可配置功能单元结构示意图	3
图 2-2 CFU 中的寄存器示意图	4
图 3-1 LUT1 端口示意图	6
图 3-2 LUT2 端口示意图	8
图 3-3 LUT3 端口示意图	9
图 3-4 LUT4 端口示意图	11
图 3-5 LUT5 端口示意图	14
图 3-6 MUX2 端口示意图	17
图 3-7 MUX4 端口示意图	18
图 3-8 MUX8 端口示意图	21
图 3-9 ALU 端口示意图	24
图 3-10 DFF 端口示意图	27
图 3-11 DFFE 端口示意图	29
图 3-12 DFFS 端口示意图	30
图 3-13 DFFSE 端口示意图	32
图 3-14 DFFR 端口示意图	34
图 3-15 DFFRE 端口示意图	35
图 3-16 DFFP 端口示意图	37
图 3-17 DFFPE 端口示意图	38
图 3-18 DFFC 端口示意图	40
图 3-19 DFFCE 端口示意图	41
图 3-20 DFFN 端口示意图	43
图 3-21 DFFNE 端口示意图	44
图 3-22 DFFNS 端口示意图	46
图 3-23 DFFNSE 端口示意图	47
图 3-24 DFFNR 端口示意图	49
图 3-25 DFFNRE 端口示意图	51
图 3-26 DFFNP 端口示意图	52
图 3-27 DFFNPE 端口示意图	54

图 3-28 DFFNC 端口示意图	55
图 3-29 DFFNCE 端口示意图	57
图 3-30 DL 端口示意图	60
图 3-31 DLE 端口示意图	61
图 3-32 DLC 端口示意图	63
图 3-33 DLCE 端口示意图	64
图 3-34 DLP 端口示意图	66
图 3-35 DLPE 端口示意图	68
图 3-36 DLN 端口示意图	69
图 3-37 DLNE 端口示意图	71
图 3-38 DLNC 端口示意图	72
图 3-39 DLNCE 端口示意图	74
图 3-40 DLNP 端口示意图	76
图 3-41 DLNPE 端口示意图	77

表目录

表 1-1 术语、缩略语	1
表 2-1 CFU 中寄存器模块信号说明	4
表 3-1 LUT1 端口介绍	6
表 3-2 LUT1 参数介绍	6
表 3-3 LUT1 真值表	7
表 3-4 LUT2 端口介绍	8
表 3-5 LUT2 参数介绍	8
表 3-6 LUT2 真值表	8
表 3-7 LUT3 端口介绍	9
表 3-8 LUT3 参数介绍	10
表 3-9 LUT3 真值表	10
表 3-10 LUT4 端口介绍	11
表 3-11 LUT4 参数介绍	12
表 3-12 LUT4 真值表	12
表 3-13 LUT5 端口介绍	14
表 3-14 LUT5 参数介绍	14
表 3-15 LUT5 真值表	15
表 3-16 MUX2 端口介绍	17
表 3-17 MUX2 真值表	17
表 3-18 MUX4 端口介绍	18
表 3-19 MUX4 真值表	19
表 3-20 MUX8 端口介绍	21
表 3-21 MUX8 真值表	21
表 3-22 ALU 功能	23
表 3-23 ALU 端口介绍	24
表 3-24 ALU 参数介绍	24
表 3-25 与 FF 相关的原语	26
表 3-26 FF 类型	26
表 3-27 DFF 端口介绍	27

表 3-28 DFF 参数介绍	28
表 3-29 DFFE 端口介绍	29
表 3-30 DFFE 参数介绍	29
表 3-31 DFFS 端口介绍	31
表 3-32 DFFS 参数介绍	31
表 3-33 DFFSE 端口介绍	32
表 3-34 DFFSE 参数介绍	32
表 3-35 DFFR 端口介绍	34
表 3-36 DFFR 参数介绍	34
表 3-37 DFFRE 端口介绍	35
表 3-38 DFFRE 参数介绍	36
表 3-39 DFFP 端口介绍	37
表 3-40 DFFP 参数介绍	37
表 3-41 DFFPE 端口介绍	38
表 3-42 DFFPE 参数介绍	39
表 3-43 DFFC 端口介绍	40
表 3-44 DFFC 参数介绍	40
表 3-45 DFFCE 端口介绍	42
表 3-46 DFFCE 参数介绍	42
表 3-47 DFFN 端口介绍	43
表 3-48 DFFN 参数介绍	43
表 3-49 DFFNE 端口介绍	45
表 3-50 DFFNE 参数介绍	45
表 3-51 DFFNS 端口介绍	46
表 3-52 DFFNS 参数介绍	46
表 3-53 DFFNSE 端口介绍	48
表 3-54 DFFNSE 参数介绍	48
表 3-55 DFFNR 端口介绍	49
表 3-56 DFFNR 参数介绍	49
表 3-57 DFFNRE 端口介绍	51
表 3-58 DFFNRE 参数介绍	51
表 3-59 DFFNP 端口介绍	52
表 3-60 DFFNP 参数介绍	53
表 3-61 DFFNPE 端口介绍	54
表 3-62 DFFNPE 参数介绍	54
表 3-63 DFFNC 端口介绍	56
表 3-64 DFFNC 参数介绍	56

表 3-65 DFFNCE 端口介绍	57
表 3-66 DFFNCE 参数介绍	57
表 3-67 与 LATCH 相关的原语	59
表 3-68 LATCH 类型	59
表 3-69 DL 端口介绍	60
表 3-70 DL 参数介绍	60
表 3-71 DLE 端口介绍	61
表 3-72 DLE 参数介绍	62
表 3-73 DLC 端口介绍	63
表 3-74 DLC 参数介绍	63
表 3-75 DLCE 端口介绍	65
表 3-76 DLCE 参数介绍	65
表 3-77 DLP 端口介绍	66
表 3-78 DLP 参数介绍	66
表 3-79 DLPE 端口介绍	68
表 3-80 DLPE 参数介绍	68
表 3-81 DLN 端口介绍	69
表 3-82 DLN 参数介绍	70
表 3-83 DLNE 端口介绍	71
表 3-84 DLNE 参数介绍	71
表 3-85 DLNC 端口介绍	73
表 3-86 DLNC 参数介绍	73
表 3-87 DLNCE 端口介绍	74
表 3-88 DLNCE 参数介绍	74
表 3-89 DLNP 端口介绍	76
表 3-90 DLNP 参数介绍	76
表 3-91 DLNPE 端口介绍	77
表 3-92 DLNPE 参数介绍	78

1 关于本手册

1.1 手册内容

Gowin 可配置功能单元(CFU)手册主要描述了可配置功能单元的结构、工作模式和原语。

1.2 相关文档

通过登录高云®半导体网站 www.gowinsemi.com.cn 可以下载、查看相关文档：

- [DS102, GW2A 系列 FPGA 产品数据手册](#)
- [DS100, GW1N 系列 FPGA 产品数据手册](#)
- [DS226, GW2AR 系列 FPGA 产品数据手册](#)
- [UG285, Gowin 存储器\(BSRAM & SSRAM\)用户指南](#)

1.3 术语、缩略语

表 1-1 中列出了本手册中出现的相关术语、缩略语及相关释义。

表 1-1 术语、缩略语

术语、缩略语	全称	含义
CFU	Configurable Function Unit	可配置功能单元
CLU	Configurable Logic Unit	可配置逻辑单元
LUT	Look-up Table	查找表
CRU	Configurable Routing Unit	可配置布线单元
CLS	Configurable Logic Section	可配置逻辑块
SSRAM	Shadow Static Random Access Memory	分布式静态随机存储器
BSRAM	Block Static Random Access Memory	块状静态随机存储器
REG	Register	寄存器
MUX2	Multiplexer 2:1	2 选 1 选择器
ALU	Arithmetic Logic Unit	算术逻辑单元
DFF	D Flip Flop	D 触发器

术语、缩略语	全称	含义
DL	Data Latch	数据锁存器

1.4 技术支持与反馈

高云半导体提供全方位技术支持，在使用过程中如有任何疑问或建议，可直接与公司联系：

网址：www.gowinsemi.com.cn

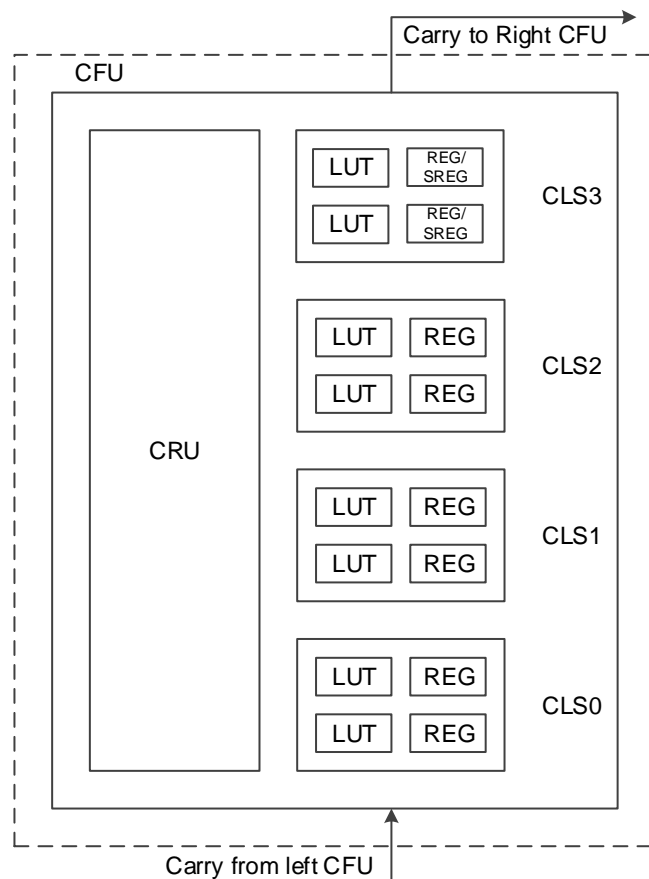
E-mail：support@gowinsemi.com

Tel: +86 755 8262 0391

2 CFU 结构

可配置功能单元(CFU)和可配置逻辑单元(CLU)是构成高云半导体 FPGA 产品内核的两种基本单元,每个基本单元可由四个可配置逻辑块(CLS)以及相应的可配置布线单元(CRU)组成,其中三个可配置逻辑块各包含两个四输入查找表(LUT)和两个寄存器(REG),另外一个可配置逻辑块只包含两个四输入查找表,如图 2-1 所示。CLU 中的可配置逻辑块不能配置为静态随机存储器,可配置为基本查找表、算术逻辑单元和只读存储器。CFU 中的可配置逻辑块可根据应用场景配置成基本查找表、算术逻辑单元、静态随机存储器和只读存储器四种工作模式。本手册以 CFU 来说明 CFU 和 CLU 的基本情况。

图 2-1 可配置功能单元结构示意图



注!

- SREG 需要特殊的软件支持。如有需要，请联系高云半导体技术支持或当地办事处。
- 目前，仅 GW1N-2、GW1N-1P5、GW1N-2B、GW1N-1P5B、GW1NR-2、GW1NR-2B 器件支持 CLS3 的 REG，且 CLS3 与 CLS2 的 CLK/CE/SR 信号同源。

2.1 CLS

2.1.1 CLS 模式

CLS 支持基本查找表、算术逻辑和存储器模式：

- 基本查找表模式

每个查找表可以被配置为一个 4 输入查找表(LUT4)，可配置逻辑单元可实现高阶查找表功能：

- 一个可配置逻辑块可配置成一个 5 输入查找表(LUT5)。
- 两个可配置逻辑块可配置成一个 6 输入查找表(LUT6)。
- 四个可配置逻辑块可配置成一个 7 输入查找表(LUT7)。
- 八个可配置逻辑块可配置成一个 8 输入查找表(LUT8)。

- 算术逻辑模式

结合进位链，查找表可配置成算术逻辑模式(ALU)，用作实现以下功能：

- 加法/减法运算
- 计数器，包括加计数器和减计数器。
- 比较器，包括大于比较、小于比较和不相等比较。
- 乘法器

- 存储器模式

在此模式下，一个可配置功能单元可构成 16 x 4 位的静态随机存储器 (SRAM) 或只读存储器 (ROM16)。

2.1.2 REG

可配置逻辑块(CLS0~CLS2)各含两个寄存器(REG)，如图 2-2 所示。

图 2-2 CFU 中的寄存器示意图

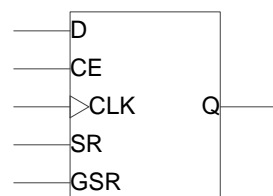


表 2-1 CFU 中寄存器模块信号说明

信号名	I/O	描述
D	I	寄存器数据输入 ^[1]
CE	I	CLK 使能信号，可配置为高电平使能或低电平使能 ^[2]

信号名	I/O	描述
CLK	I	时钟信号，可配置为上升沿触发或下降沿触发 ^[2]
SR	I	本地置复位输入，可配置为如下功能 ^[2] ： <ul style="list-style-type: none"> ● 同步复位 ● 同步置位 ● 异步复位 ● 异步置位 ● 无本地置复位
GSR ^{[3],[4]}	I	全局复置位，可配置为如下功能 ^[4] ： <ul style="list-style-type: none"> ● 异步复位 ● 异步置位 ● 无全局复置位
Q	O	寄存器输出

注！

- [1]信号 D 的来源可以选择同一可配置逻辑块中任一查找表的输出，也可以选择来自于 CRU 的输入。因此在查找表被占用的情况下，寄存器仍可以单独使用。
- [2]CFU 中可配置逻辑块的 CE/CLK/SR 均可独立配置选择。
- [3]在高云半导体 FPGA 产品内部，GSR 通过直连线连接，不通过 CRU。
- [4]SR 与 GSR 同时有效时 GSR 有较高的优先级。

2.2 CRU

布线资源单元 CRU 的功能主要包括两个方面：

- 输入选择功能：为 CFU 的输入信号提供输入源选择。
- 布线资源功能：为 CFU 的输入/输出信号提供连接关系，包括 CFU 内部连接、CFU 之间连接以及 CFU 和 FPGA 内部其他功能模块之间的连接。

3 CFU 原语

3.1 LUT

输入查找表 LUT，常用的 LUT 结构有 LUT1、LUT2、LUT3、LUT4，其区别在于查找表输入位宽的不同。

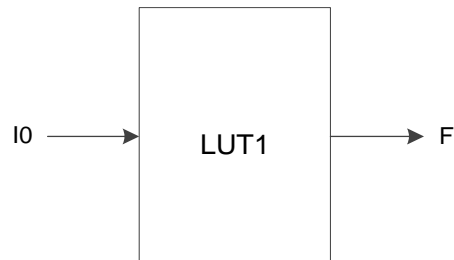
3.1.1 LUT1

原语介绍

LUT1(1-input Look-up Table)是其中最简单的一种，常用于实现缓冲器和反相器。LUT1 为 1 输入的查找表，通过 parameter 给 INIT 赋初值后，根据输入的地址查找对应的数据并输出结果。

端口示意图

图 3-1 LUT1 端口示意图



端口介绍

表 3-1 LUT1 端口介绍

端口	I/O	描述
I0	Input	数据输入信号
F	Output	数据输出信号

参数介绍

表 3-2 LUT1 参数介绍

参数	范围	默认	描述
----	----	----	----

INIT	2'h0~2'h3	2'h0	LUT1 初始值
------	-----------	------	----------

真值表

表 3-3 LUT1 真值表

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

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );
```

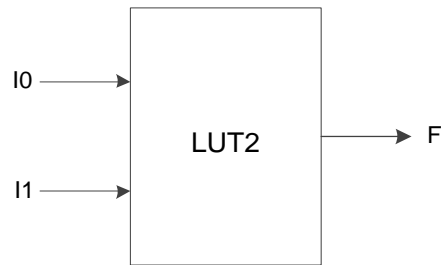
3.1.2 LUT2

原语介绍

LUT2(2-input Look-up Table)为 2 输入的查找表,通过 parameter 给 INIT 赋初值后,根据输入的地址查找对应的数据并输出结果。

端口示意图

图 3-2 LUT2 端口示意图



端口介绍

表 3-4 LUT2 端口介绍

端口	I/O	描述
I0	Input	数据输入信号
I1	Input	数据输入信号
F	Output	数据输出信号

参数介绍

表 3-5 LUT2 参数介绍

参数	范围	默认	描述
INIT	4'h0~4'hf	4'h0	LUT2 初始值

真值表

表 3-6 LUT2 真值表

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

原语例化

Verilog 例化:

```
LUT2 instName (
    .I0(I0),
    .I1(I1),
    .F(F)
);
```

```
defparam instName.INIT=4'h1;
```

Vhdl 例化:

```
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
  );
```

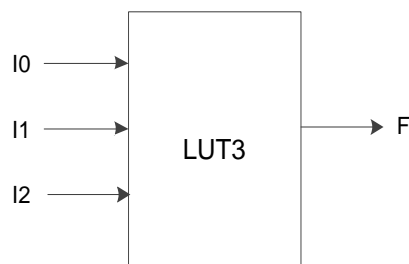
3.1.3 LUT3

原语介绍

LUT3(3-input Look-up Table)为 3 输入的查找表,通过 parameter 给 INIT 赋初值后,根据输入的地址查找对应的数据并输出结果。

端口示意图

图 3-3 LUT3 端口示意图



端口介绍

表 3-7 LUT3 端口介绍

端口	I/O	描述
I0	Input	数据输入信号
I1	Input	数据输入信号
I2	Input	数据输入信号

端口	I/O	描述
F	Output	数据输出信号

参数介绍

表 3-8 LUT3 参数介绍

参数	范围	默认	描述
INIT	8'h00~8'hff	8'h00	LUT3 初始值

真值表

表 3-9 LUT3 真值表

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]

原语例化

Verilog 例化:

```
LUT3 instName (
    .I0(I0),
    .I1(I1),
    .I2(I2),
    .F(F)
);
```

```
defparam instName.INIT=8'h10;
```

Vhdl 例化

```
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
    );

```

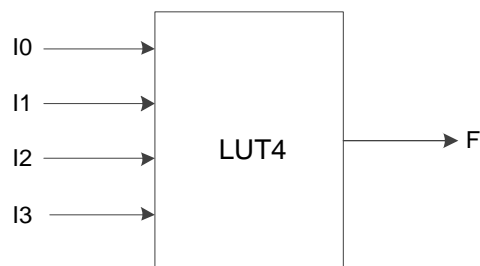
3.1.4 LUT4

原语介绍

LUT4(4-input Look-up Table)为4输入的查找表,通过 parameter 给 INIT 赋初值后,根据输入的地址查找对应的数据并输出结果。

端口示意图

图 3-4 LUT4 端口示意图



端口介绍

表 3-10 LUT4 端口介绍

端口	I/O	描述
I0	Input	数据输入信号
I1	Input	数据输入信号
I2	Input	数据输入信号
I3	Input	数据输入信号
F	Output	数据输出信号

参数介绍

表 3-11 LUT4 参数介绍

参数	范围	默认	描述
INIT	16'h0000~16'hffff	16'h0000	LUT4 初始值

真值表

表 3-12 LUT4 真值表

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]

原语例化

Verilog 例化:

```

LUT4 instName (
    .I0(I0),
    .I1(I1),
    .I2(I2),
    .I3(I3),
    .F(F)
);
defparam instName.INIT=16'h1011;

```

Vhdl 例化:

```

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

原语介绍

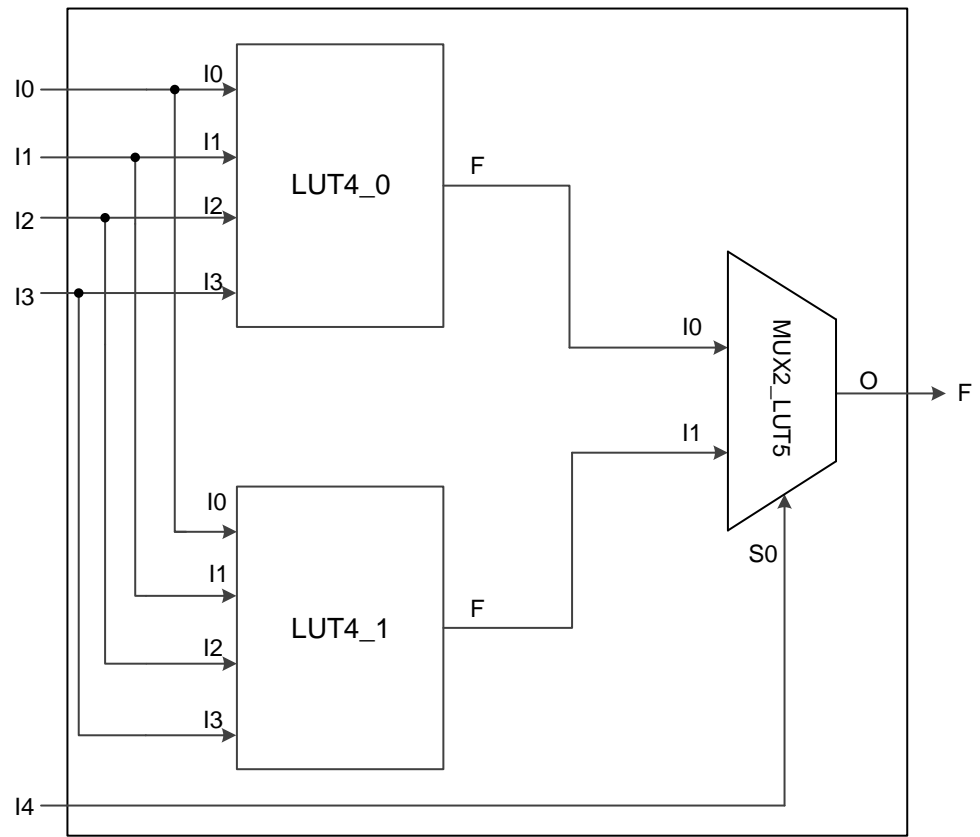
Wide LUT 是通过 LUT4 和 MUX2 构造高阶 LUT，高云 FPGA 目前支持的构造高阶 LUT 的 MUX2 有 MUX2_LUT5/MUX2_LUT6/MUX2_LUT7/MUX2_LUT8。

高阶 LUT 的构造方式如下：两个 LUT4 和 MUX2_LUT5 可组合实现 LUT5，两个组合实现的 LUT5 和 MUX2_LUT6 可组合实现 LUT6，两个组合实现的 LUT6 和 MUX2_LUT7 可组合实现 LUT7，两个组合实现的 LUT7 和 MUX2_LUT8 可组合实现 LUT8。

以 LUT5 为例介绍 Wide LUT 的使用。

端口示意图

图 3-5 LUT5 端口示意图



端口介绍

表 3-13 LUT5 端口介绍

端口名	I/O	描述
I0	Input	数据输入信号
I1	Input	数据输入信号
I2	Input	数据输入信号
I3	Input	数据输入信号
I4	Input	数据输入信号
F	Output	数据输出信号

参数介绍

表 3-14 LUT5 参数介绍

参数	范围	默认	描述
INIT	32'h00000~32'hffff	32'h00000	LUT5 初始值

真值表

表 3-15 LUT5 真值表

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]

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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 是多路复用器，拥有多路输入，通过通道选择信号确定其中一路数据传送到输出端。高云原语中有 2 选 1 和 4 选 1 等多路复用器。

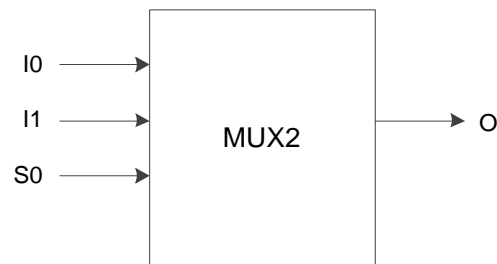
3.2.1 MUX2

原语介绍

MUX2(2-to-1 Multiplexer)是 2 选 1 的复用器，根据选择信号，从两个输入中选择其中一个作为输出。

端口示意图

图 3-6 MUX2 端口示意图



端口介绍

表 3-16 MUX2 端口介绍

端口	I/O	描述
I0	Input	数据输入信号
I1	Input	数据输入信号
S0	Input	数据选择信号
O	Output	数据输出信号

真值表

表 3-17 MUX2 真值表

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

原语例化

Verilog 例化:

```
MUX2 instName (
    .I0(I0),
    .I1(I1),
    .S0(S0),
```

```

        .O(O)
    );
Vhdl 例化:
    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
        );

```

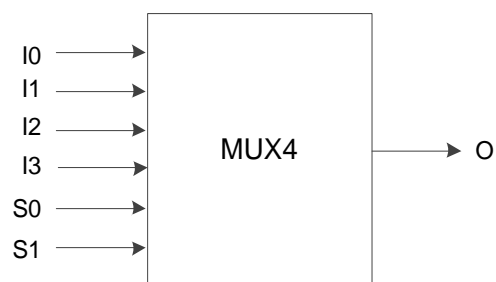
3.2.2 MUX4

原语介绍

MUX4(4-to-1 Multiplexer)是 4 选 1 的多路复用器，根据选择信号，从四个输入中选择其中一个作为输出。

端口示意图

图 3-7 MUX4 端口示意图



端口介绍

表 3-18 MUX4 端口介绍

端口	I/O	描述
I0	Input	数据输入信号

端口	I/O	描述
I1	Input	数据输入信号
I2	Input	数据输入信号
I3	Input	数据输入信号
S0	Input	数据选择信号
S1	Input	数据选择信号
O	Output	数据输出信号

真值表

表 3-19 MUX4 真值表

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

原语例化

Verilog 例化:

```

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

```

Vhdl 例化:

```

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

原语介绍

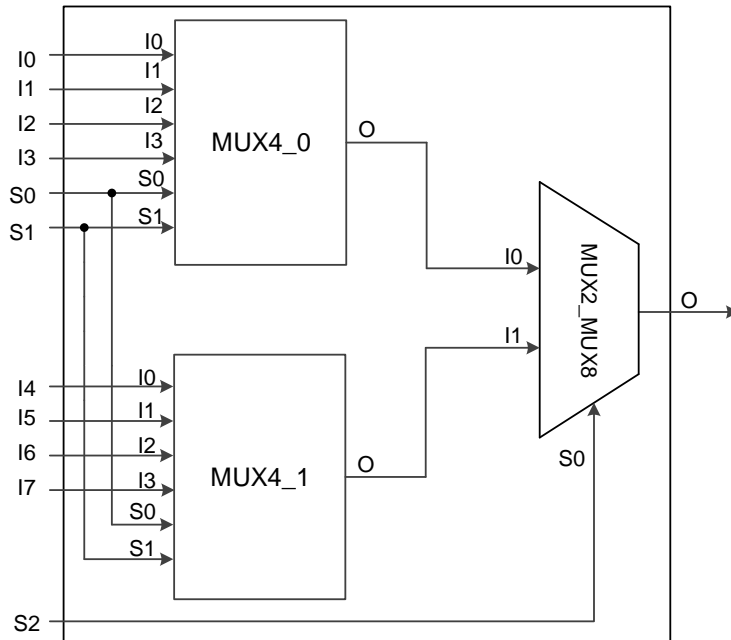
Wide MUX 是通过 MUX4 和 MUX2 构造高阶 MUX，高云 FPGA 目前支持的构造高阶 MUX 的 MUX2 有 MUX2_MUX8/ MUX2_MUX16/ MUX2_MUX32。

高阶 MUX 的构造方式如下：两个 MUX4 和 MUX2_MUX8 可组合实现 MUX8，两个组合实现的 MUX8 和 MUX2_MUX16 可组合实现 MUX16，两个组合实现的 MUX16 和 MUX2_MUX32 可组合实现 MUX32。

以 MUX8 为例介绍 Wide MUX 的使用。

端口示意图

图 3-8 MUX8 端口示意图



端口介绍

表 3-20 MUX8 端口介绍

端口	输入/输出	描述
I0	Input	数据输入信号
I1	Input	数据输入信号
I2	Input	数据输入信号
I3	Input	数据输入信号
I4	Input	数据输入信号
I5	Input	数据输入信号
I6	Input	数据输入信号
I7	Input	数据输入信号
S0	Input	数据选择信号
S1	Input	数据选择信号
S2	Input	数据选择信号
O	Output	数据输出信号

真值表

表 3-21 MUX8 真值表

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

原语例化

Verilog 例化:

```

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 例化:

```

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

原语介绍

ALU(2-input Arithmetic Logic Unit)2 输入算术逻辑单元，实现了 ADD/SUB/ADDSUB 等功能，具体功能如表 3-22 所示。

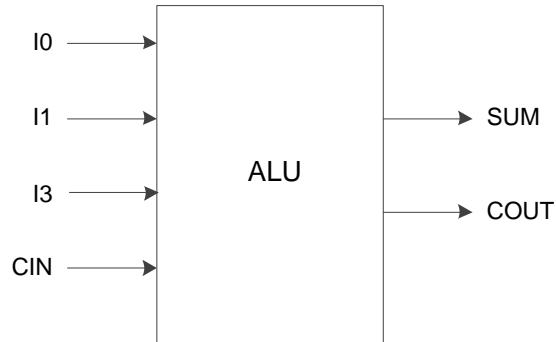
表 3-22 ALU 功能

项目	描述
ADD	加法运算
SUB	减法运算
ADDSUB	加/减法运算，由 I3 选择：1，加法；0，减法
CUP	加计数器
CDN	减计数器
CUPCDN	加/减计数器，由 I3 选择：1，加计数器；0，减计数器
GE	大于等于比较器
NE	不等于比较器

项目	描述
LE	小于等于比较器
MULT	乘法器

端口示意图

图 3-9 ALU 端口示意图



端口介绍

表 3-23 ALU 端口介绍

端口	Input/Output	描述
I0	Input	数据输入信号
I1	Input	数据输入信号
I3	Input	数据选择信号，用于 ADDSUB 加减选择或 CUPCDN 的加减计数器选择
CIN	Input	数据进位输入信号
COUT	Output	数据进位输出信号
SUM	Output	数据输出信号

参数介绍

表 3-24 ALU 参数介绍

参数	范围	默认	描述
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;

参数	范围	默认	描述
			6:CUP; 7:CDN; 8:CUPCDN; 9:MULT

原语例化

Verilog 例化:

```

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

```

Vhdl 例化:

```

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

触发器是时序电路中常用的基本元件，FPGA 内部的时序逻辑都可通过 FF 结构实现，常用的 FF 有 DFF、DFFE、DFFS、DFFSE 等，其区别在于复位方式、触发方式等方面。

与 FF 相关的原语有 20 个，如表 3-25 所示。

表 3-25 与 FF 相关的原语

原语	描述
DFF	D 触发器
DFFE	带时钟使能 D 触发器
DFFS	带同步置位 D 触发器
DFFSE	带时钟使能、同步置位 D 触发器
DFFR	带同步复位 D 触发器
DFFRE	带时钟使能、同步复位 D 触发器
DFFP	带异步置位 D 触发器
DFFPE	带时钟使能、异步置位 D 触发器
DFFC	带异步清零 D 触发器
DFFCE	带时钟使能、异步清零 D 触发器
DFFN	下降沿 D 触发器
DFFNE	下降沿带时钟使能 D 触发器
DFFNS	下降沿带同步置位 D 触发器
DFFNSE	下降沿带时钟使能、同步置位 D 触发器
DFFNR	下降沿带同步复位 D 触发器
DFFNRE	下降沿带时钟使能、同步复位 D 触发器
DFFNP	下降沿带异步置位 D 触发器
DFFNPE	下降沿带时钟使能、异步置位 D 触发器
DFFNC	下降沿带异步清零 D 触发器
DFFNCE	下降沿带时钟使能、异步清零 D 触发器

放置规则

表 3-26 FF 类型

编号	类型 1	类型 2
1	DFFS	DFFR
2	DFFSE	DFFRE
3	DFFP	DFFC

编号	类型 1	类型 2
4	DFFPE	DFFCE
5	DFFNS	DFFNR
6	DFFNSE	DFFNRE
7	DFFNP	DFFNC
8	DFFNPE	DFFNCE

- 相同类型的 DFF, 可以放在同一个 CLS 的 2 个 FF 上, 除数据输入 pin 外的其它输入必须共线;
- 不同类型的 DFF, 表 3-26 中同一编号的两种类型可以放在同一个 CLS 的 2 个 FF 上, 除数据输入 pin 外的其它输入必须共线;
- 可以约束 DFF 和 ALU 在同一个 CLS 的相同或不同位置;
- 可以约束 DFF 和 LUT 在同一个 CLS 的相同或不同位置。

注!

共线是指必须是同一条 net, 经过反相器前后的两条 net 为不共线, 不可放在同一个 CLS。

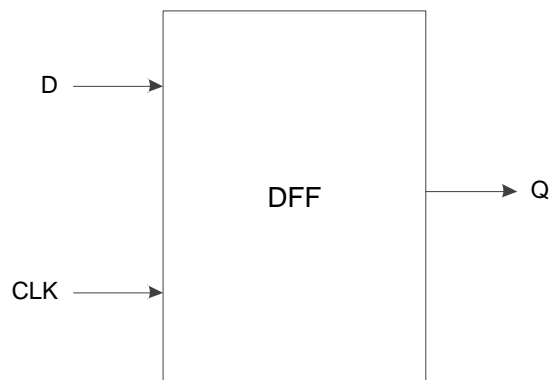
3.4.1 DFF

原语介绍

DFF(D Flip-Flop)是其中最简单常用的一种触发器, 常用于信号采样和处理, 是上升沿触发的 D 触发器。

端口示意图

图 3-10 DFF 端口示意图



端口介绍

表 3-27 DFF 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
Q	Output	数据输出信号

参数介绍

表 3-28 DFF 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFF 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );
```

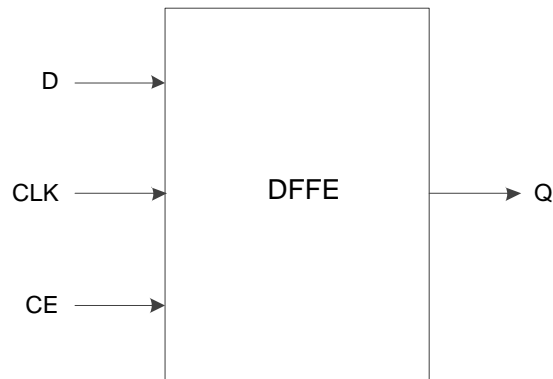
3.4.2 DFFE

原语介绍

DFFE(D Flip-Flop with Clock Enable)是上升沿触发的 D 触发器，具有时钟使能功能。

端口示意图

图 3-11 DFFE 端口示意图



端口介绍

表 3-29 DFFE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-30 DFFE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFE 初始值

原语例化

Verilog 例化:

```
DFFE instName (
    .D(D),
    .CLK(CLK),
    .CE(CE),
    .Q(Q)
```

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

Vhdl 例化:

```
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
  );
```

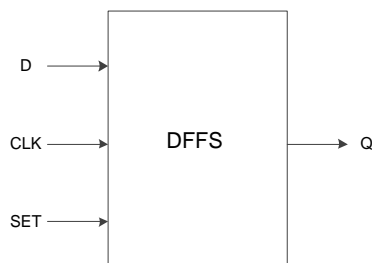
3.4.3 DFFS

原语介绍

DFFS(D Flip-Flop with Synchronous Set)是上升沿触发的 D 触发器，具有同步置位功能。

端口示意图

图 3-12 DFFS 端口示意图



端口介绍

表 3-31 DFFS 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
SET	Input	同步置位信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-32 DFFS 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DFFS 初始值

原语例化

Verilog 例化:

```

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

```

Vhdl 例化:

```

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
);

```

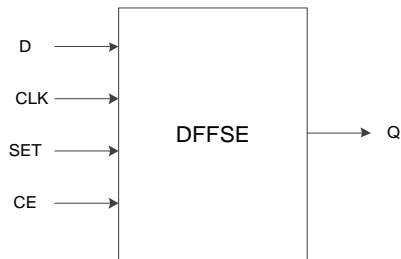
3.4.4 DFFSE

原语介绍

DFFSE(D Flip-Flop with Clock Enable and Synchronous Set)是上升沿触发的 D 触发器，具有同步置位和时钟使能功能。

端口示意图

图 3-13 DFFSE 端口示意图



端口介绍

表 3-33 DFFSE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
SET	Input	同步置位信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-34 DFFSE 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DFFSE 初始值

原语例化

Verilog 例化:

```

    DFFSE instName (
        .D(D),

```

```

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

```

Vhdl 例化:

```

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
    );

```

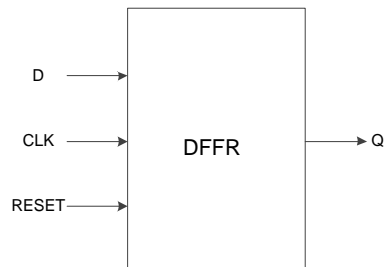
3.4.5 DFFR

原语介绍

DFFR(D Flip-Flop with Synchronous Reset)是上升沿触发的 D 触发器，具有同步复位功能。

端口示意图

图 3-14 DFFR 端口示意图



端口介绍

表 3-35 DFFR 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
RESET	Input	同步复位信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-36 DFFR 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFR 初始值

原语例化

Verilog 例化:

```

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

Vhdl 例化:

```

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
    );

```

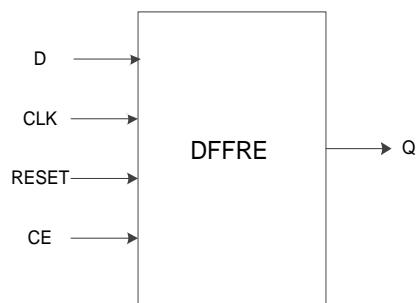
3.4.6 DFFRE

原语介绍

DFFRE(D Flip-Flop with Clock Enable and Synchronous Reset)是上升沿触发的 D 触发器，具有同步复位和时钟使能功能。

端口示意图

图 3-15 DFFRE 端口示意图



端口介绍

表 3-37 DFFRE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
RESET	Input	同步复位信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-38 DFFRE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFRE 初始值

原语例化

Verilog 例化:

```

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

```

Vhdl 例化:

```

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
    );

```

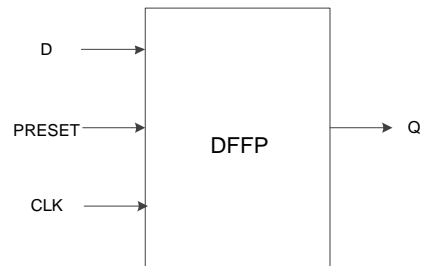
3.4.7 DFFP

原语介绍

DFFP(D Flip-Flop with Asynchronous Preset)是上升沿触发的 D 触发器，具有异步置位功能。

端口示意图

图 3-16 DFFP 端口示意图



端口介绍

表 3-39 DFFP 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
PRESET	Input	异步置位信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-40 DFFP 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DFFP 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```

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
  );

```

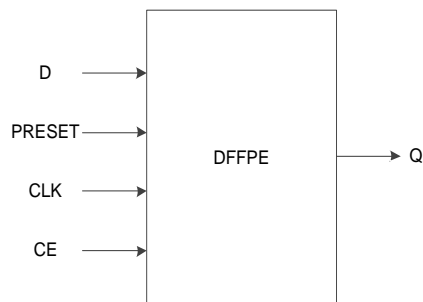
3.4.8 DFFPE

原语介绍

DFFPE(D Flip-Flop with Clock Enable and Asynchronous Preset)是上升沿触发的 D 触发器，具有异步置位和时钟使能功能。

端口示意图

图 3-17 DFFPE 端口示意图



端口介绍

表 3-41 DFFPE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号

端口	I/O	描述
PRESET	Input	异步置位信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-42 DFFPE 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DFFPE 初始值

原语例化

Verilog 例化:

```

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

```

Vhdl 例化:

```

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
);

```

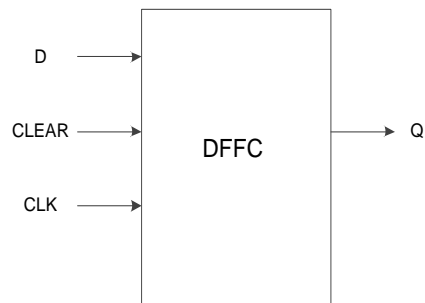
3.4.9 DFFC

原语介绍

DFFC(D Flip-Flop with Asynchronous Clear)是上升沿触发的 D 触发器，具有异步清零功能。

端口示意图

图 3-18 DFFC 端口示意图



端口介绍

表 3-43 DFFC 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
CLEAR	Input	异步清零信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-44 DFFC 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFC 初始值

原语例化

Verilog 例化:

```

DFFC instName (
    .D(D),

```

```

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

```

Vhdl 例化:

```

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

```

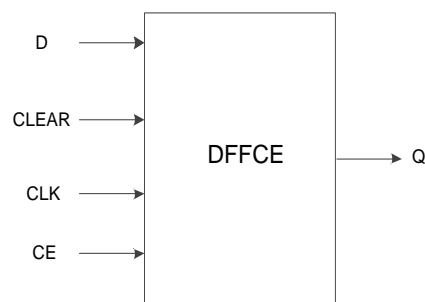
3.4.10 DFFCE

原语介绍

DFFCE(D Flip-Flop with Clock Enable and Asynchronous Clear)是上升沿触发的 D 触发器，具有异步清零和时钟使能功能。

端口示意图

图 3-19 DFFCE 端口示意图



端口介绍

表 3-45 DFFCE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
CLEAR	Input	异步清零信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-46 DFFCE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFCE 初始值

原语例化

Verilog 例化:

```

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

```

Vhdl 例化:

```

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
   );

```

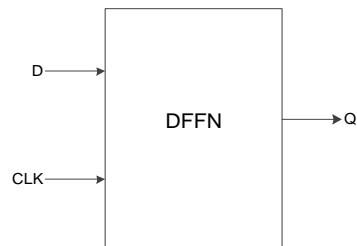
3.4.11 DFFN

原语介绍

DFFN(D Flip-Flop with Negative-Edge Clock)是下降沿触发的 D 触发器。

端口示意图

图 3-20 DFFN 端口示意图



端口介绍

表 3-47 DFFN 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
Q	Output	数据输出信号

参数介绍

表 3-48 DFFN 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFN 初始值

原语例化

Verilog 例化:

```
DFFN instName (
```

```

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

```

Vhdl 例化:

```

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
    );

```

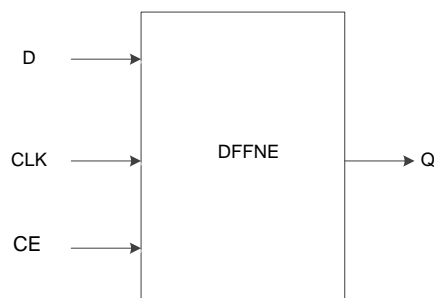
3.4.12 DFFNE

原语介绍

DFFNE(D Flip-Flop with Negative-Edge Clock and Clock Enable)是下降沿触发的 D 触发器，具有时钟使能功能。

端口示意图

图 3-21 DFFNE 端口示意图



端口介绍

表 3-49 DFFNE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-50 DFFNE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFNE 初始值

原语例化

Verilog 例化:

```

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

```

Vhdl 例化:

```

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
    );

```

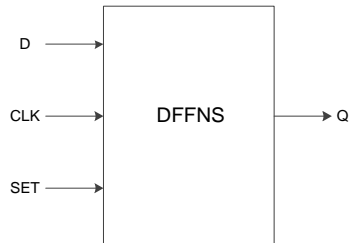
3.4.13 DFFNS

原语介绍

DFFNS(D Flip-Flop with Negative-Edge Clock and Synchronous Set) 是下降沿触发的 D 触发器，具有同步置位功能。

端口示意图

图 3-22 DFFNS 端口示意图



端口介绍

表 3-51 DFFNS 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
SET	Input	同步置位信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-52 DFFNS 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DFFNS 初始值

原语例化

Verilog 例化:

```

DFFNS instName (
    .D(D),
    .CLK(CLK),

```



```

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

```

Vhdl 例化:

```

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
    );

```

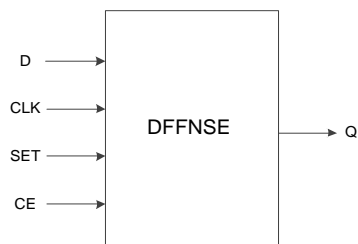
3.4.14 DFFNSE

原语介绍

DFFNSE(D Flip-Flop with Negative-Edge Clock,Clock Enable,and Synchronous Set)是下降沿触发的 D 触发器,具有同步置位和时钟使能功能。

端口示意图

图 3-23 DFFNSE 端口示意图



端口介绍

表 3-53 DFFNSE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
SET	Input	同步置位信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-54 DFFNSE 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DFFNSE 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );

```

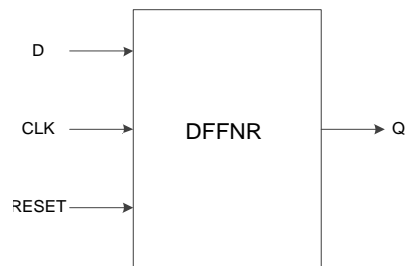
3.4.15 DFFNR

原语介绍

DFFNR(D Flip-Flop with Negative-Edge Clock and Synchronous Reset)是下降沿触发的 D 触发器，具有同步复位功能。

端口示意图

图 3-24 DFFNR 端口示意图



端口介绍

表 3-55 DFFNR 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
RESET	Input	同步复位信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-56 DFFNR 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFNR 初始值

原语例化

Verilog 例化:

```

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

```

Vhdl 例化:

```

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
    );

```

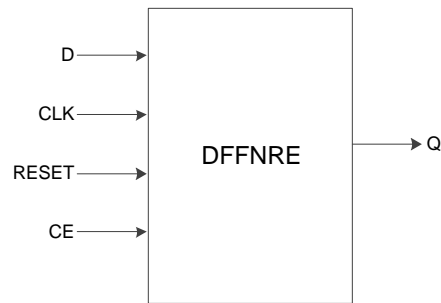
3.4.16 DFFNRE

原语介绍

DFFNRE(D Flip-Flop with Negative-Edge Clock,Clock Enable, and Synchronous Reset)是下降沿触发的 D 触发器，具有同步复位和时钟使能功能。

端口示意图

图 3-25 DFFNRE 端口示意图



端口介绍

表 3-57 DFFNRE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
RESET	Input	同步复位信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-58 DFFNRE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFNRE 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );

```

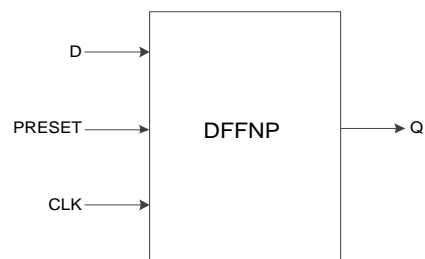
3.4.17 DFFNP

原语介绍

DFFNP(D Flip-Flop with Negative-Edge Clock and Asynchronous Preset)是下降沿触发的 D 触发器，具有异步置位功能。

端口示意图

图 3-26 DFFNP 端口示意图



端口介绍

表 3-59 DFFNP 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
PRESET	Input	异步置位信号，高电平有效

端口	I/O	描述
Q	Output	数据输出信号

参数介绍

表 3-60 DFFNP 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DFFNP 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );
```

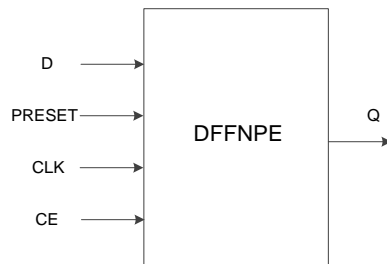
3.4.18 DFFNPE

原语介绍

DFFNPE(D Flip-Flop with Negative-Edge Clock,Clock Enable, and Asynchronous Preset)是下降沿触发的 D 触发器，具有异步置位和时钟使能功能。

端口示意图

图 3-27 DFFNPE 端口示意图



端口介绍

表 3-61 DFFNPE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
PRESET	Input	异步置位信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-62 DFFNPE 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DFFNPE 初始值

原语例化

Verilog 例化:

```
DFFNPE instName (
    .D(D),
    .CLK(CLK),
    .PRESET(PRESET),
    .CE(CE),
    .Q(Q)
```



```
);
defparam instName.INIT=1'b1;
```

Vhdl 例化:

```
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
    );
```

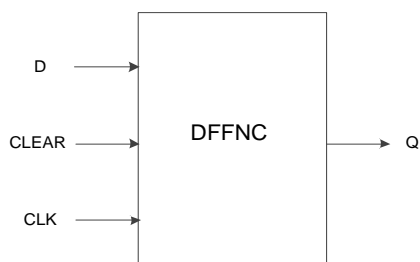
3.4.19 DFFNC

原语介绍

DFFNC(D Flip-Flop with Negative-Edge Clock and Asynchronous Clear)是下降沿触发的 D 触发器，具有异步清零功能。

端口示意图

图 3-28 DFFNC 端口示意图



端口介绍

表 3-63 DFFNC 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
CLEAR	Input	异步清零信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-64 DFFNC 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFNC 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );

```

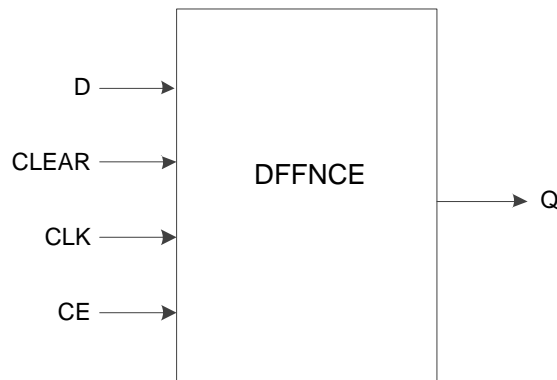
3.4.20 DFFNCE

原语介绍

DFFNCE(D Flip-Flop with Negative-Edge Clock,Clock Enable and Asynchronous Clear)是下降沿触发的 D 触发器，具有异步清零和时钟使能功能。

端口示意图

图 3-29 DFFNCE 端口示意图



端口介绍

表 3-65 DFFNCE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLK	Input	时钟输入信号
CLEAR	Input	异步清零信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-66 DFFNCE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DFFNCE 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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

锁存器是一种对电平触发的存储单元电路，其可在特定输入电平作用下改变状态。与 LATCH 相关的原语有 12 个，如表 3-67 所示。

表 3-67 与 LATCH 相关的原语

原语	描述
DL	数据锁存器
DLE	带锁存使能的数据锁存器
DLC	带异步清零的数据锁存器
DLCE	带异步清零和锁存使能的数据锁存器
DLP	带异步预置位的数据锁存器
DLPE	带异步预置位和锁存使能的数据锁存器
DLN	低电平有效的数据锁存器
DLNE	带锁存使能的低电平有效的数据锁存器
DLNC	带异步清零的低电平有效的数据锁存器
DLNCE	带异步清零和锁存使能的低电平有效的数据锁存器
DLNP	带异步预置位的低电平有效的数据锁存器
DLNPE	带异步预置位和锁存使能的低电平有效的数据锁存器

放置规则

表 3-68 LATCH 类型

编号	类型 1	类型 2
1	DLC	DLP
2	DLCE	DLPE
3	DLNC	DLNP
4	DLNCE	DLNPE

- 相同类型的 DL，可以放在同一个 CLS 的 2 个 FF 上，除数据输入 pin 外的其它输入必须共线；
- 不同类型的 DL，表 3-68 中同一编号的两种类型可以放在同一个 CLS 的 2 个 FF 上，除数据输入 pin 外的其它输入必须共线；
- 可以约束 DL 和 ALU 在同一个 CLS 的相同或不同位置；
- 可以约束 DL 和 LUT 在同一个 CLS 的相同或不同位置。

注！

共线是指必须是同一条 net，经过反相器前后的两条 net 为不共线，不可放在同一个 CLS。

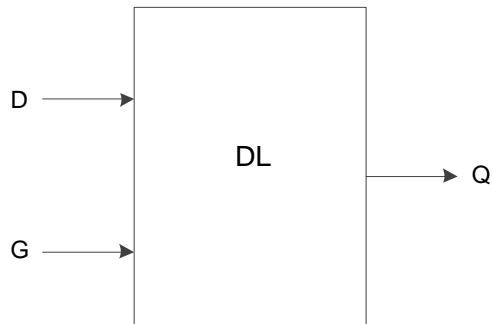
3.5.1 DL

原语介绍

DL(Data Latch)是其中最简单常用的一种锁存器，控制信号 **G** 高电平有效。

端口示意图

图 3-30 DL 端口示意图



端口介绍

表 3-69 DL 端口介绍

端口	I/O	描述
D	Input	数据输入信号
G	Input	数据控制信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-70 DL 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DL 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```

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
  );

```

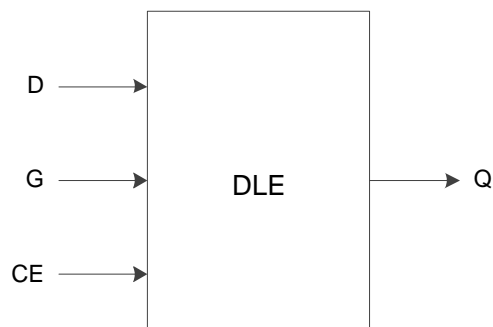
3.5.2 DLE

原语介绍

DLE(Data Latch with Latch Enable)是具有使能控制的一种锁存器，控制信号 G 高电平有效。

端口示意图

图 3-31 DLE 端口示意图



端口介绍

表 3-71 DLE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
G	Input	数据控制信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-72 DLE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DLE 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );
```

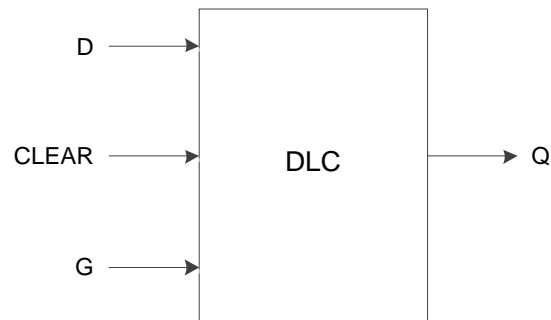

3.5.3 DLC

原语介绍

DLC(Data Latch with Asynchronous Clear)是具有异步清零功能的一种锁存器，控制信号 G 高电平有效。

端口示意图

图 3-32 DLC 端口示意图



端口介绍

表 3-73 DLC 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLEAR	Input	异步清零信号，高电平有效
G	Input	数据控制信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-74 DLC 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DLC 初始值

原语例化

Verilog 例化:

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

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

Vhdl 例化:

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

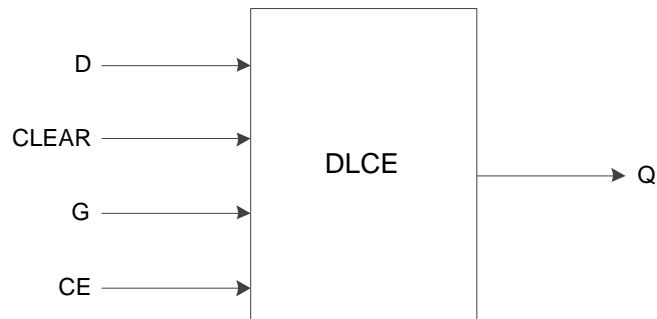
3.5.4 DLCE

原语介绍

DLCE(Data Latch with Asynchronous Clear and Latch Enable)是具有使能控制和异步清零功能的一种锁存器，控制信号 G 高电平有效。

端口示意图

图 3-33 DLCE 端口示意图



端口介绍

表 3-75 DLCE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLEAR	Input	异步清零信号，高电平有效
G	Input	数据控制信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-76 DLCE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DLCE 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );

```

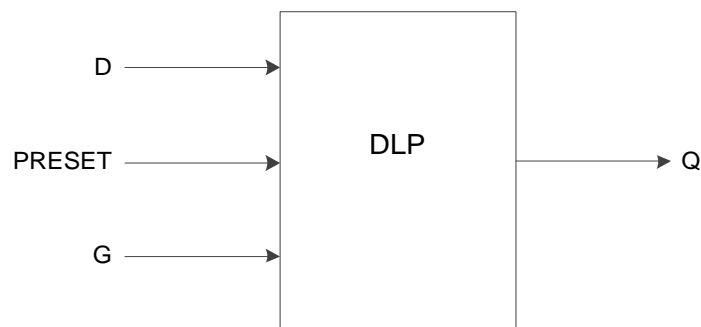
3.5.5 DLP

原语介绍

DLP(Data Latch with Asynchronous Preset)是具有置位功能的一种锁存器，控制信号 G 高电平有效。

端口示意图

图 3-34 DLP 端口示意图



端口介绍

表 3-77 DLP 端口介绍

端口	I/O	描述
D	Input	数据输入信号
PRESET	Input	异步置位信号，高电平有效
G	Input	数据控制信号，高电平有效
Q	Output	数据输出信号

参数介绍

表 3-78 DLP 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DLP 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );
```

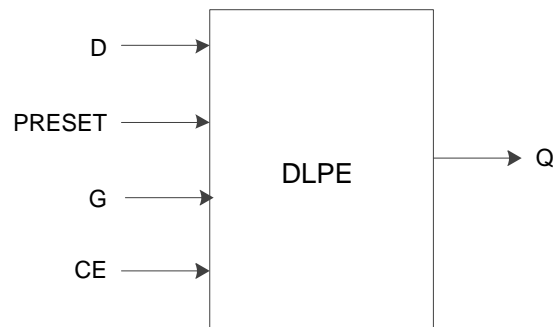
3.5.6 DLPE

原语介绍

DLPE(Data Latch with Asynchronous Preset and Latch Enable)是具有使能控制和置位功能的一种锁存器，控制信号 G 高电平有效。

端口示意图

图 3-35 DLPE 端口示意图



端口介绍

表 3-79 DLPE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
PRESET	Input	异步置位信号，高电平有效
G	Input	数据控制信号，高电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-80 DLPE 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DLPE 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );

```

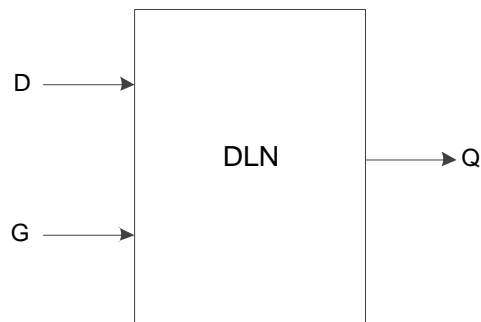
3.5.7 DLN

原语介绍

DLN(Data Latch with Inverted Gate)是控制信号低电平有效的锁存器。

端口示意图

图 3-36 DLN 端口示意图



端口介绍

表 3-81 DLN 端口介绍

端口	I/O	描述
D	Input	数据输入信号
G	Input	数据控制信号，低电平有效

端口	I/O	描述
Q	Output	数据输出信号

参数介绍

表 3-82 DLN 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DLN 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
    );
```

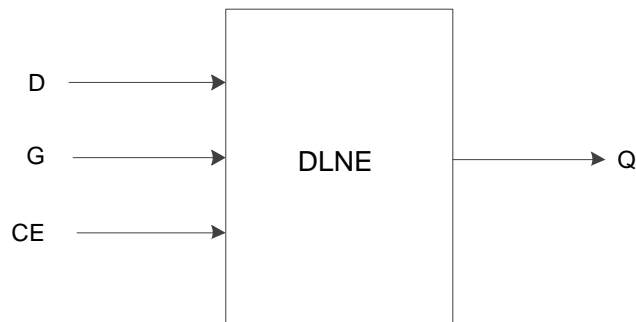

3.5.8 DLNE

原语介绍

DLNE(Data Latch with Latch Enable and Inverted Gate)是一种具有使能控制的锁存器，控制信号 G 低电平有效。

端口示意图

图 3-37 DLNE 端口示意图



端口介绍

表 3-83 DLNE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
G	Input	数据控制信号，低电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-84 DLNE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DLNE 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```

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
  );

```

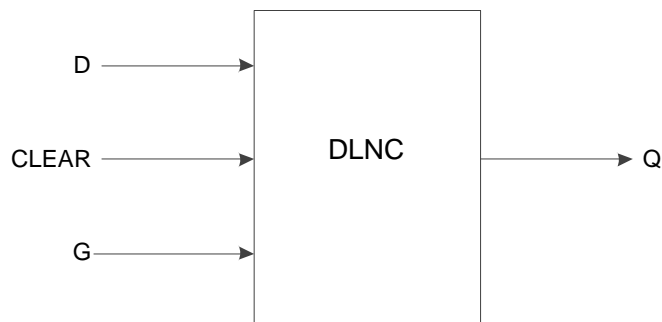
3.5.9 DLNC

原语介绍

DLNC(Data Latch with Asynchronous Clear and Inverted Gate)是一种具有异步清零功能的锁存器，控制信号 G 低电平有效。

端口示意图

图 3-38 DLNC 端口示意图



端口介绍

表 3-85 DLNC 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLEAR	Input	异步清零信号，高电平有效
G	Input	数据控制信号，低电平有效
Q	Output	数据输出信号

参数介绍

表 3-86 DLNC 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DLNC 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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

```

```
);
```

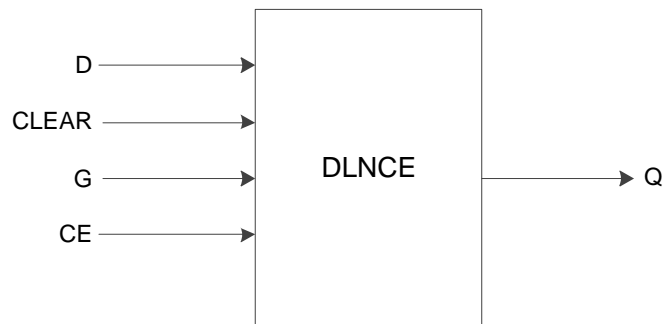
3.5.10 DLNCE

原语介绍

DLNCE(Data Latch with Asynchronous Clear, Latch Enable, and Inverted Gate)是具有使能控制和异步清零功能的一种锁存器，控制信号 G 低电平有效。

端口示意图

图 3-39 DLNCE 端口示意图



端口介绍

表 3-87 DLNCE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
CLEAR	Input	异步清零信号，高电平有效
G	Input	数据控制信号，低电平有效
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-88 DLNCE 参数介绍

参数	范围	默认	描述
INIT	1'b0	1'b0	DLNCE 初始值

原语例化

Verilog 例化:

```

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

```

Vhdl 例化:

```

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
    );

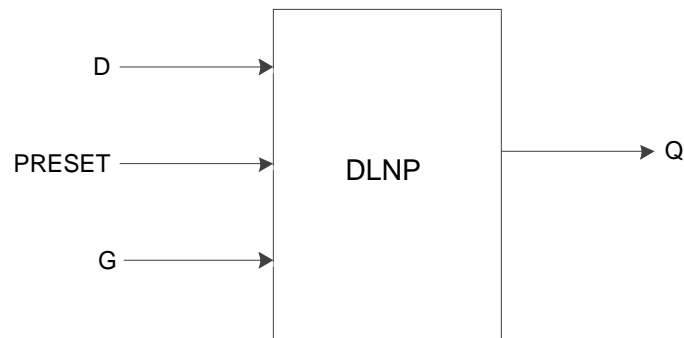
```

3.5.11 DLNP**原语介绍**

DLNP(Data Latch with Asynchronous Preset and Inverted Gate)是具有置位功能的一种锁存器，控制信号 G 低电平有效。

端口示意图

图 3-40 DLNP 端口示意图



端口介绍

表 3-89 DLNP 端口介绍

端口	I/O	描述
D	Input	数据输入信号
PRESET	Input	异步置位信号，高电平有效
G	Input	数据控制信号，低电平有效
Q	Output	数据输出信号

参数介绍

表 3-90 DLNP 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DLNP 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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
        );

```

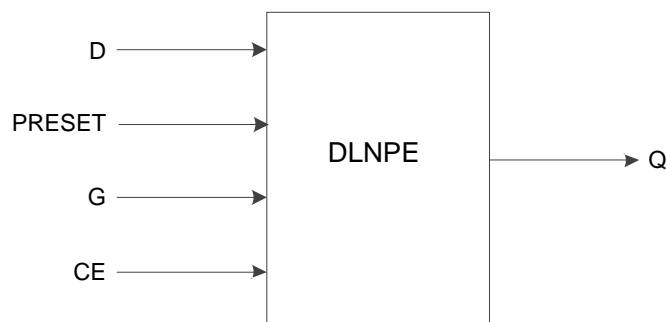
3.5.12 DLNPE

原语介绍

DLNPE(Data Latch with Asynchronous Preset,Latch Enable and Inverted Gate)是具有使能控制和置位功能的一种锁存器，控制信号 G 低电平有效。

端口示意图

图 3-41 DLNPE 端口示意图



端口介绍

表 3-91 DLNPE 端口介绍

端口	I/O	描述
D	Input	数据输入信号
PRESET	Input	异步置位信号，高电平有效
G	Input	数据控制信号，低电平有效

端口	I/O	描述
CE	Input	时钟使能信号
Q	Output	数据输出信号

参数介绍

表 3-92 DLNPE 参数介绍

参数	范围	默认	描述
INIT	1'b1	1'b1	DLNPE 初始值

原语例化

Verilog 例化:

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

Vhdl 例化:

```
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 原语可参考 [UG285, Gowin 存储器\(BSRAM & SSRAM\)用户指南](#)。

