



Gowin 时钟资源(Clock) 用户指南

UG286-1.2, 2018-01-05

版权所有©2018 广东高云半导体科技股份有限公司

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
2016/05/18	1.05	初始版本
2016/07/15	1.06	标准化插图
2016/08/31	1.07	适用 GW2A 系列 FPGA 产品
2016/10/27	1.08	适用 GW2AR 系列 FPGA 产品
2017/09/22	1.09	根据最新软件原语库修改
2017/10/16	1.10	增加 GW1N 的对应描述及相关示意图
2018/01/05	1.2	更新高速时钟相关内容。

目录

目录	i
图目录	iv
表目录	v
1 关于本手册	1
1.1 手册内容	1
1.2 适用产品	1
1.3 相关文档	2
1.4 术语、缩略语	3
1.5 技术支持与反馈	3
2 概述	4
3 全局时钟	5
3.1 概述	5
3.2 BUFG	9
3.2.1 介绍	9
3.2.2 原语	9
3.2.3 端口	9
3.3 DCS	9
3.3.1 介绍	9
3.3.2 原语	9
3.3.3 端口	10
3.3.4 参数	10
3.3.5 功能图	11
3.3.6 时序图	11
3.4 DQCE	12
3.4.1 介绍	12
3.4.2 原语	12
3.4.3 端口	12
3.4.4 功能图	13

4 高速时钟	14
4.1 概述	14
4.2 CLKDIV	16
4.2.1 介绍	16
4.2.2 原语	16
4.2.3 端口	17
4.2.4 参数	17
4.2.5 应用示意图	17
4.3 DDR 存储器接口时钟管理 DQS	17
5 OSC	19
5.1 概述	19
5.2 功能框图	19
5.3 原语	19
5.4 端口	20
5.5 参数	20
5.6 频率值	20
6 DLL	21
6.1 概述	21
6.2 功能框图	21
6.3 原语	21
6.3.1 DLL 原语	21
6.3.2 DLLDLY 原语	22
6.4 端口	23
6.4.1 DLL 端口	23
6.4.2 DLLDLY 端口	24
6.5 参数	25
6.5.1 DLL 参数	25
6.5.2 DLLDLY 参数	25
6.6 应用示意图	26
7 PLL	27
7.1 概述	27
7.2 功能框图	27
7.3 原语	28
7.4 端口	30
7.5 参数	31
7.6 频率计算	35

7.7 举例 35

图目录

图 3-1 GW1N/GW1NR 系列产品 GCLK 象限分布示意图.....	6
图 3-2 GW2A/GW2AR 系列产品时钟资源.....	7
图 3-3 GW2A/GW2AR 系列产品 GCLK 象限分布示意图.....	8
图 3-4 DCS 接口示意图.....	11
图 3-5 DCS Rising Edge 模式下的时序示意图.....	11
图 3-6 DCS Falling Edge 模式下的时序示意图.....	11
图 3-7 DQCE 结构示意图.....	13
图 4-1 GW1N-4/GW1NR-4 HCLK 示意图.....	14
图 4-2 GW1N-9/GW1NR-9 HCLK 示意图.....	15
图 4-2 GW1NS-2 HCLK 示意图.....	15
图 4-3 GW2A/GW2AR 系列产品 HCLK 示意图.....	15
图 4-4 CLKDIV 应用示意图.....	17
图 4-5 DQS 示意图.....	18
图 5-1 OSC 示意图.....	19
图 6-1 DLL 示意图.....	21
图 6-2 DLL 应用示意图.....	26
图 7-1 PLL 示意图.....	27

表目录

表 1-1 术语、缩略语	3
表 3-1 BUFG 端口信号	9
表 3-2 DCS 端口信号	10
表 3-3DCS 参数描述	10
表 3-4 DQCE 端口信号	12
表 4-1 CLKDIV 端口信号	17
表 4-2 CLKDIV 参数描述	17
表 5-1 OSC 端口信号	20
表 5-2 OSC 参数描述	20
表 6-1 DLL 端口信号	23
表 6-2DLLDLY 端口信号	24
表 6-3 DLL 参数描述	25
表 6-4 DLLDLY 参数描述	25
表 7-1 PLL 端口信号	30
表 7-2 PLL 参数	31
表 7-3 PLL 相位参数调整对照表	32
表 7-4 PLL 占空比参数调整对照表	34
表 7-5PLL 延迟参数调整对照表	34

1 关于本手册

1.1 手册内容

本文档介绍了时钟资源部分的功能、定义及使用方法。

该手册主要包括四个部分：

1. 时钟资源章节，主要介绍了时钟资源的整体布局、时钟资源种类及相关时钟资源原语介绍；
2. OSC 章节，介绍了内部晶振作为用户时钟的频率范围、参数设置及原语例化等；
3. DLL 章节，介绍了 DLL 和 DLLDLY 的功能、定义等；
4. PLL 章节，介绍了 PLL 的功能、定义等。

1.2 适用产品

本手册中描述的信息适用于以下高云半导体 FPGA 产品：

1. GW1N 系列 FPGA 产品：GW1N-1、GW1N-2、GW1N-4、GW1N-6、GW1N-9
2. GW1NR 系列 FPGA 产品：GW1NR-4、GW1NR-9
3. GW2A 系列 FPGA 产品：GW2A-18、GW2A-55
4. GW2AR 系列 FPGA 产品：GW2AR-18

1.3 相关文档

通过登录高云半导体网站 <http://www.gowinsemi.com.cn> 可以下载、查看以下相关文档：

1. GW1N 系列 FPGA 产品数据手册
2. GW1NR 系列 FPGA 产品数据手册
3. GW1N 系列 FPGA 产品封装与管脚手册
4. GW1NR 系列 FPGA 产品封装与管脚手册
5. GW1N 系列 FPGA 产品编程配置手册
6. GW2A 系列 FPGA 产品数据手册
7. GW2AR 系列 FPGA 产品数据手册
8. GW2A 系列 FPGA 产品封装与管脚手册
9. GW2AR 系列 FPGA 产品封装与管脚手册
10. GW2A 系列 FPGA 产品编程配置手册

1.4 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
FPGA	Field Programmable Gate Array	现场可编程门阵列
OSC	Oscillator	晶体振荡器
DLL	Delay-locked Loop	延迟锁相环
DLLDLY	DLL Delay	DLL 延迟
PLL	Phase-locked Loop	锁相环
GCLK	Global Clock	全局时钟
HCLK	High Clock	高速时钟
BUFG	Buffer Global	全局时钟缓冲器
DCS	Dynamic Clock Selector	动态时钟选择器
DQCE	Dynamic Quadrant Clock Enable	动态使能象限时钟
CLKDIV	Clock Divider	时钟分频器

1.5 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：<http://www.gowinsemi.com.cn>

E-mail：support@gowinsemi.com

Tel: +86 755 8262 0391

2 概述

本章介绍了高云半导体 **FPGA** 产品的时钟资源，包括专用的时钟输入、缓冲区和布线资源。时钟的基础设施提供了一系列低电容、低偏移互连线，非常适合承载高频信号，最大限度地减少时钟偏差和提高性能，可应用于所有的时钟信号。

时钟资源及布线对 **FPGA** 高性能的应用至关重要。高云半导体 **FPGA** 产品提供了专用全局时钟网络(**GCLK**)，直接连接到器件的所有资源。除了 **GCLK** 资源，还提供了锁相环 (**PLL**)、延迟锁相环 (**DLL**)、高速时钟 **HCLK** 和 **DDR** 存储器接口数据脉冲时钟 **DQS** 等时钟资源。

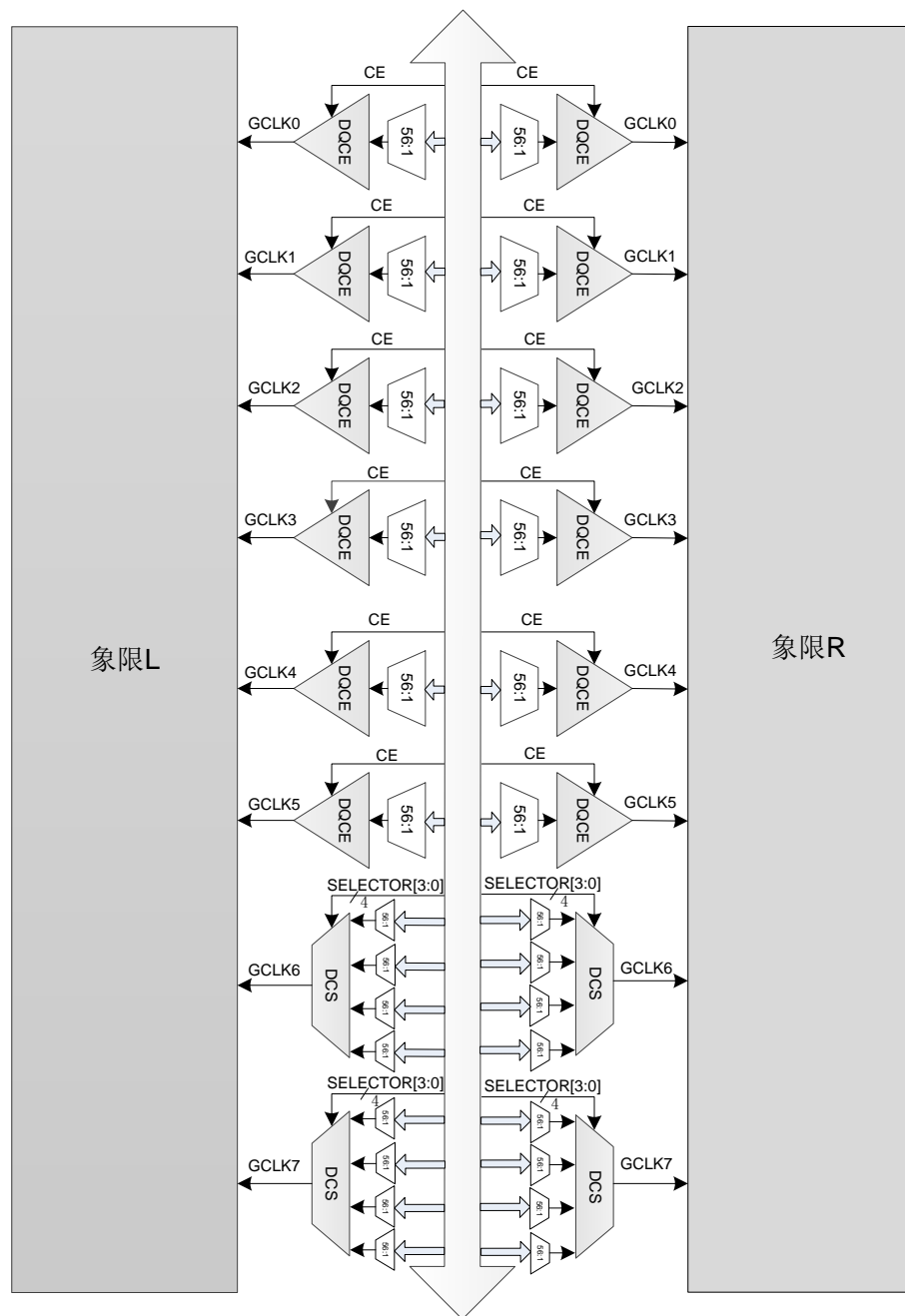
3 全局时钟

3.1 概述

GW1N/GW1NR 系列 FPGA 产品

GCLK 在 GW1N/GW1NR 系列产品中按象限分布,分成 L、R 两个象限,如图 3-1 所示: FPGA 被分成两个象限,每个象限提供 8 个 GCLK 网络,每个 GCLK 可选时钟源包括专用的时钟输入管脚和普通的布线资源单元 (CRU),使用专用的时钟输入管脚可以取得更好的时钟性能。

图 3-1 GW1N/GW1NR 系列产品 GCLK 象限分布示意图



GW2A/GW2AR 系列 FPGA 产品

GCLK 在 GW2A/GW2AR 系列 FPGA 产品中按象限分布，分为四个象限，每个象限提供 8 个 GCLK 网络。GCLK 的可选时钟源包括专用的时钟输入管脚和普通布线资源，使用专用的时钟输入管脚具有更好的时钟性能。

图 3-2 GW2A/GW2AR 系列产品时钟资源

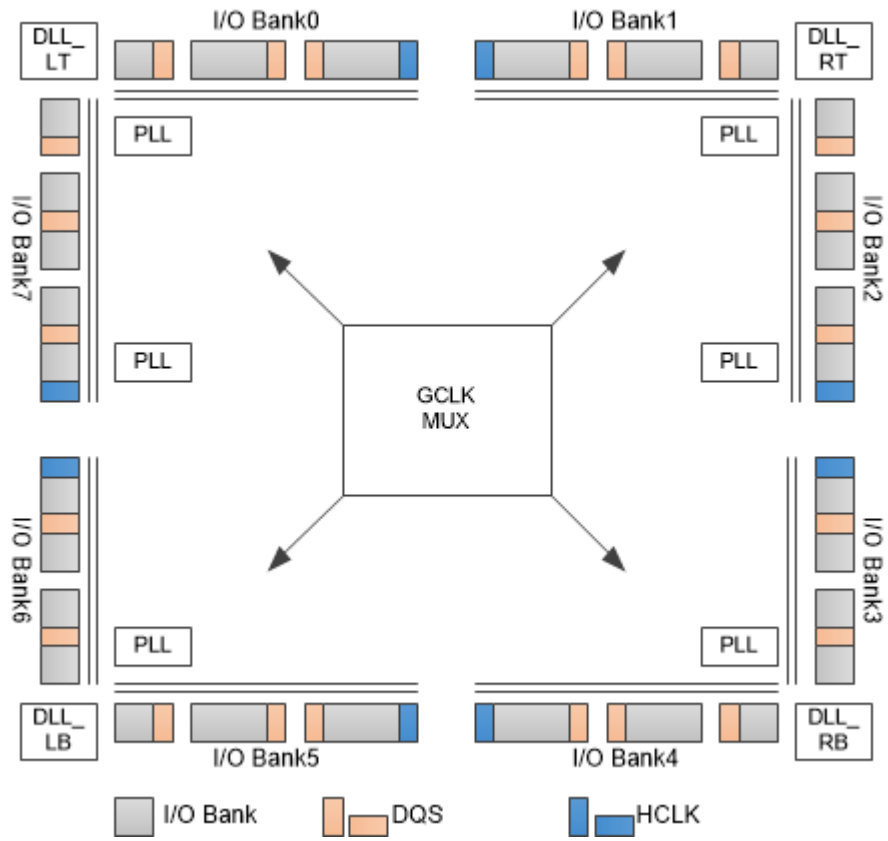
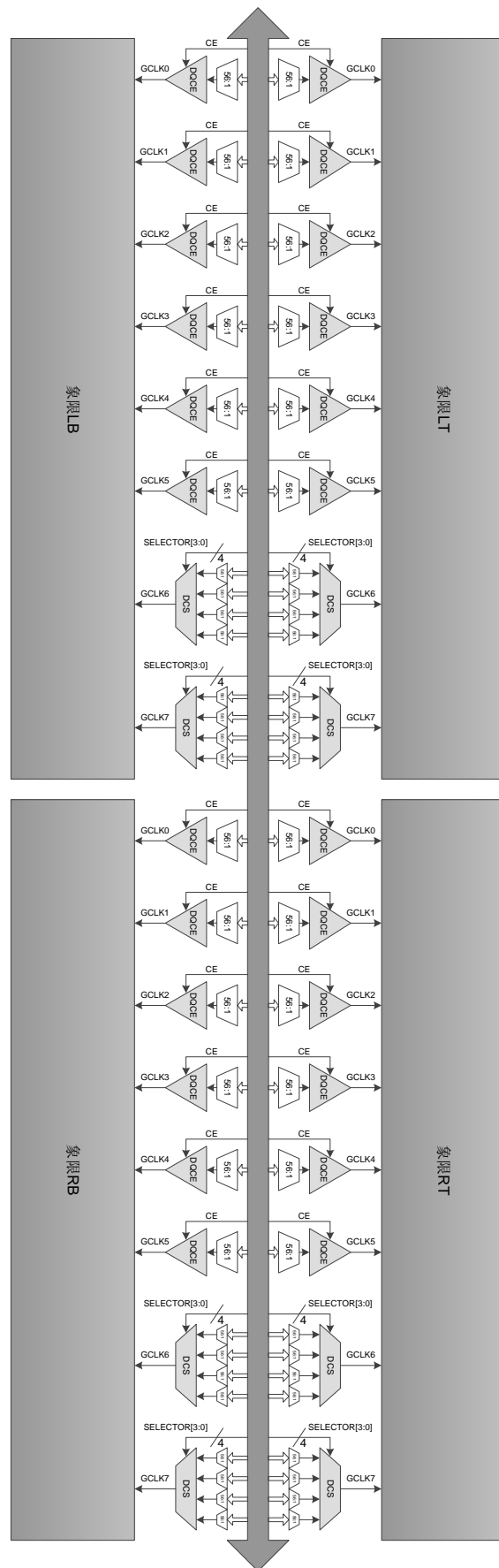


图 3-3 GW2A/GW2AR 系列产品 GCLK 象限分布示意图



3.2 BUFG

3.2.1 介绍

BUFG 是具有高扇出的全局时钟缓冲器，具有高扇出驱动能力的缓冲器，可以将信号连到时钟抖动可以忽略不计的全局时钟网络。BUFG 组件还可应用于典型的高扇出信号和网络，如复位信号和时钟使能信号。如果要对全局时钟实现 PLL 或 DLL 等时钟管理，则需要该缓冲器。

3.2.2 原语

VHDL

```
COMPONENT BUFG
  PORT(
    O : out std_logic;
    I : in std_logic
  );
end COMPONENT;
```

Verilog

```
module BUFG (O, I);
  output O;
  input I;
endmodule
```

3.2.3 端口

表 3-1 BUFG 端口信号

端口	描述
I	时钟输入
O	时钟输出

3.3 DCS

3.3.1 介绍

DCS (Dynamic Clock Selector)，每个象限的 GCLK6~GCLK7 由 DCS 控制，选择四个输入时钟中的一个作为全局时钟。内部逻辑可以通过 CRU 在四个时钟输入之间动态选择，输出不带毛刺的时钟。

3.3.2 原语

VHDL

```
COMPONENT DCS
  GENERIC (
    DCS_MODE : string := "RISING"
  );
```

```

PORT (
    CLK0 : IN std_logic;
    CLK1 : IN std_logic;
    CLK2 : IN std_logic;
    CLK3 : IN std_logic;
    CLKSEL : IN std_logic_vector(3 downto 0);
    SELFORCE : IN std_logic;
    CLKOUT : OUT std_logic
);
end COMPONENT;

```

Verilog

```

module DCS (CLK0, CLK1, CLK2, CLK3, CLKSEL, SELFORCE, CLKOUT);
input CLK0, CLK1, CLK2, CLK3, SELFORCE;
input [3:0] CLKSEL;
output CLKOUT;
parameter DCS_MODE = "RISING";
endmodule

```

3.3.3 端口

表 3-2 DCS 端口信号

端口	描述
CLKSEL [3:0]	时钟选择信号;
CLK0	时钟输入 0
CLK1	时钟输入 1
CLK2	时钟输入 2
CLK3	时钟输入 3
SELFORCE	0: glitchless 模式 1: Non-glitchless 模式
CLKOUT	时钟输出

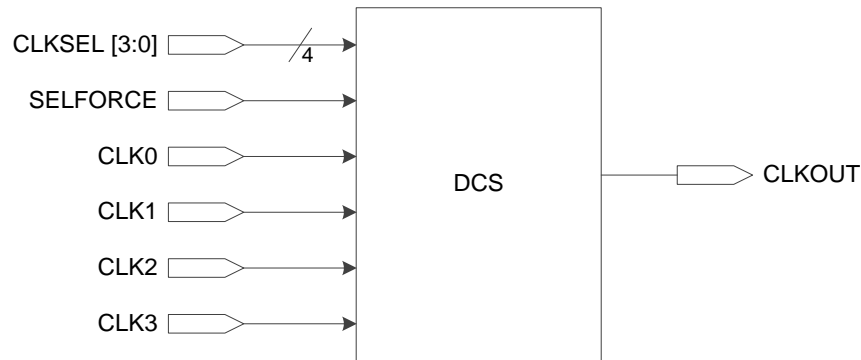
3.3.4 参数

表 3-3DCS 参数描述

参数	描述	取值选项	默认值
DCS_MODE	RISING edge 模式; FALLING edge 模式; Clock buffer 模式	CLK0,CLK1,CLK2,CLK3, GND,VCC,RISING,FALLIN, CLK0_GND,CLK1_GND, CLK2_GND,CLK3_GND, CLK0_VCC,CLK1_VCC, CLK2_VCC,CLK3_VCC	RISING

3.3.5 功能图

图 3-4 DCS 接口示意图

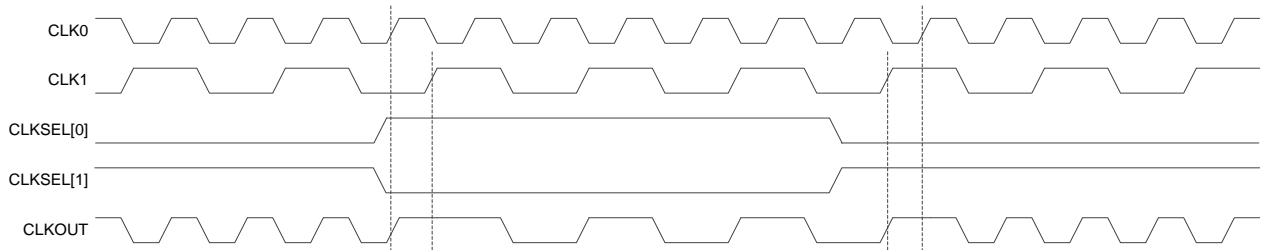


3.3.6 时序图

1. DCS Rising Edge 模式

在当前选择时钟的上升沿后转入常量 1，在新选择时钟的上升沿后转入新时钟，如图 3-5 所示。

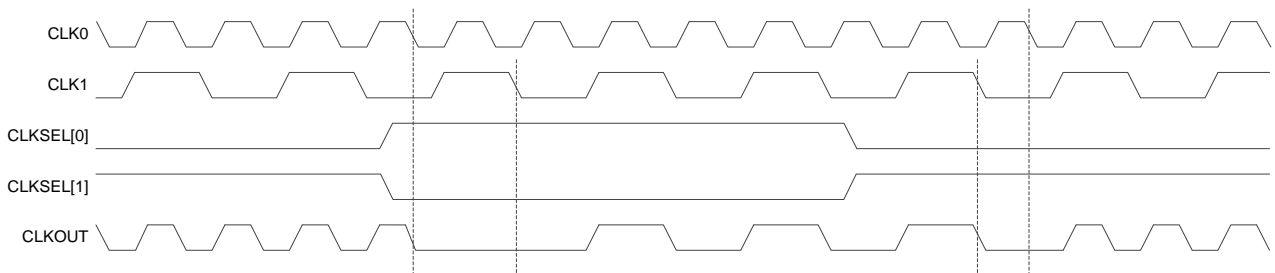
图 3-5 DCS Rising Edge 模式下的时序示意图



2. DCS Falling Edge 模式

在当前选择时钟的下降沿后转入常量 0，在新选择时钟的下降沿后转入新时钟，如图 3-6 所示。

图 3-6 DCS Falling Edge 模式下的时序示意图



3. Clock Buffer 模式

此模式下，DCS 简化为普通的 Clock Buffer。

3.4 DQCE

3.4.1 介绍

通过 DQCE(Dynamic Quadrant Clock Enable)可动态打开/关闭 GCLK0~GCLK5。关闭 GCLK0~GCLK5 时钟，GCLK0~GCLK5 驱动的内部逻辑不再翻转，降低了器件的总体功耗。

3.4.2 原语

VHDL

```

COMPONENT DQCE
  PORT (
    CLKOUT : OUT std_logic;
    CE : IN std_logic;
    CLKIN : IN std_logic
  );
end COMPONENT;
end components;

```

Verilog

```

module DQCE(CLKIN,CE,CLKOUT);
input CLKIN;
input CE;
output CLKOUT;
endmodule

```

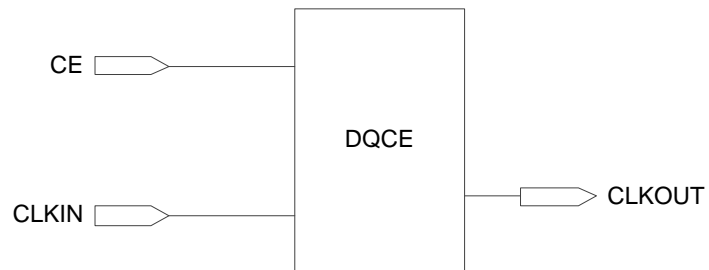
3.4.3 端口

表 3-4 DQCE 端口信号

端口	描述
CE	使能信号，高电平有效
CLKIN	时钟输入
CLKOUT	时钟输出

3.4.4 功能图

图 3-7 DQCE 结构示意图



4 高速时钟

4.1 概述

高云半导体 FPGA 产品的高速时钟 HCLK，具有低抖动和低偏差性能，可以支持 I/O 完成高性能数据传输，是专门针对源时钟同步的数据传输接口而设计的。

GW1N-1 有 2 个 HCLK 资源，位于 Bank2。

GW1N-4/ GW1NR-4 有 6 个 HCLK 资源，其中 Bank1、Bank2 和 Bank3 各有 2 个，每个 Bank 上下部分各一个。

GW1N-9/ GW1NR-9 有 8 个 HCLK 资源，其中 Bank0、Bank1、Bank2、Bank3 各有 2 个，每个 Bank 上下部分各一个。任何一个 Bank 中 HCLK 时钟输入信号可送到其他任何一个 Bank 中 HCLK。

GW1NS-2 有 8 个 HCLK 资源，Bank0、Bank1、Bank2、Bank3 各有 2 个 HCLK。

图 4-1 GW1N-4/GW1NR-4 HCLK 示意图

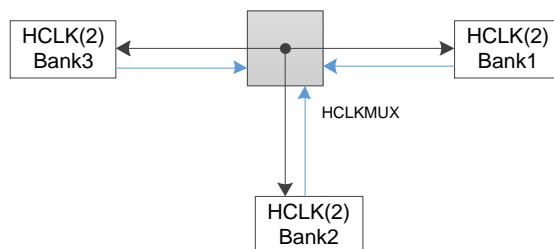


图 4-2 GW1N-9/GW1NR-9 HCLK 示意图

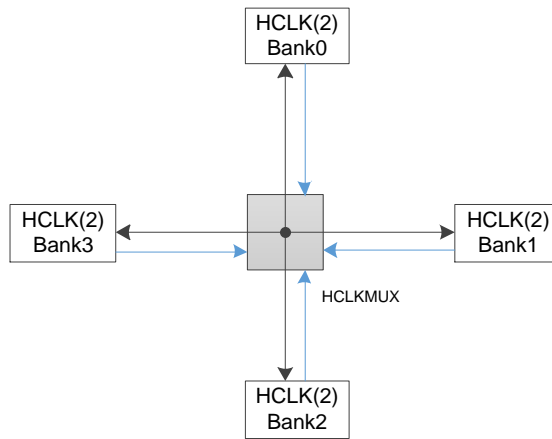
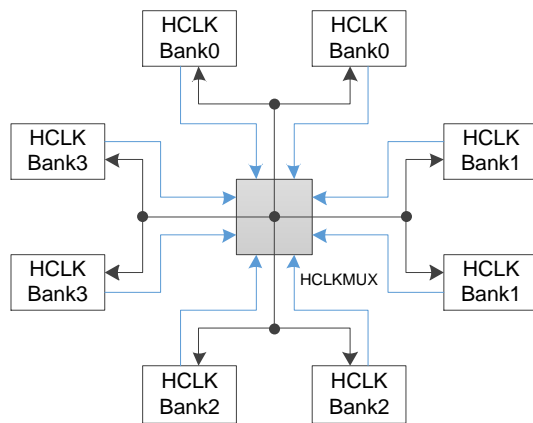
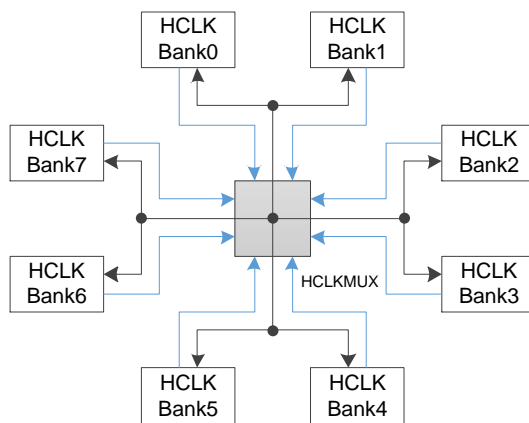


图 4-3 GW1NS-2 HCLK 示意图



GW2A/GW2AR、GW2A55 产品有 8 个 HCLK 资源，每个 Bank 有一个 HCLK。

图 4-4 GW2A/GW2AR 系列产品 HCLK 示意图



高速时钟 HCLK 中间有个 HCLKMUX 模块，HCLKMUX 能将任何一个 Bank 中的 HCLK 时钟输入信号送到其他任何一个 Bank 中，这使得 HCLK 的使用更加灵活。

HCLK 可以提供给用户使用的功能模块如下：

- DHCE：动态的高速时钟使能模块，功能类似于 DQCE。可动态的打开/关闭高速时钟信号；
- CLKDIV：高速时钟分频模块，每个 Bank 中有一个 CLKDIV。生成和输入时钟相位一致的分频时钟，用于 IO 逻辑工作模式中；
- DHCS：动态的高速时钟选择器；
- PADCLKDEL：动态延迟调整模块，用于专用时钟管脚输入的时钟信号。

4.2 CLKDIV

4.2.1 介绍

高速时钟模块 CLKDIV 对时钟进行 2、3.5、4、5、8 分频，可为 IDE54/IVIDEO/OSER4/OVIDEO 等 IO 逻辑提供时钟。

4.2.2 原语

VHDL

```

COMPONENT CLKDIV
  GENERIC(
    DIV_MODE : STRING := "2";
    GSREN : STRING := "false"
  );
  PORT(
    HCLKIN : IN std_logic;
    RESETN : IN std_logic;
    CALIB: In std_logic;
    CLKOUT : OUT std_logic
  );
end COMPONENT;
```

Verilog

```

module CLKDIV(HCLKIN, RESETN, CALIB, CLKOUT);
input HCLKIN;
input RESETN;
input CALIB;
output CLKOUT;
parameter DIV_MODE = "2";
```



```
parameter GSREN = "false";
endmodule
```

4.2.3 端口

表 4-1 CLKDIV 端口信号

端口	描述
HCLKIN	时钟输入
RESETN	复位信号，低有效
CALIB	动态信号调整，调整输出时钟
CLKOUT	时钟输出

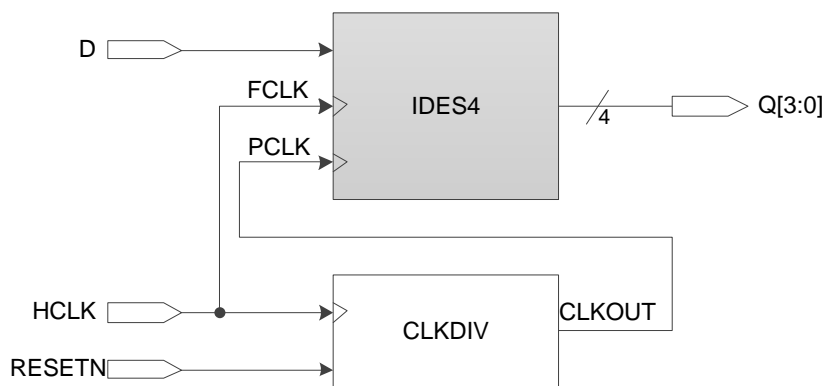
4.2.4 参数

表 4-2 CLKDIV 参数描述

参数	描述	默认
DIV_MODE	分频系数： 2, 3.5, 4, 5, 8	2
GSREN	全局复位使能信号： false, true	false

4.2.5 应用示意图

图 4-5 CLKDIV 应用示意图



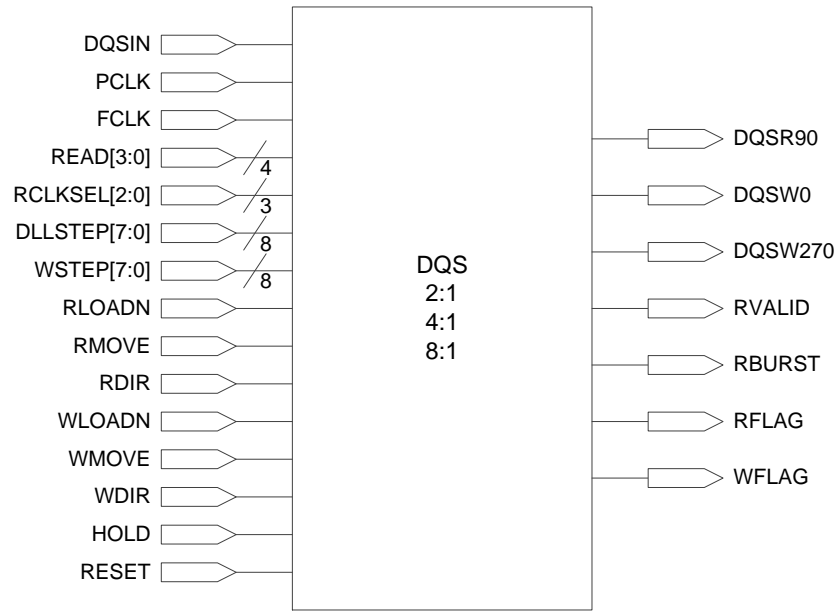
4.3 DDR 存储器接口时钟管理 DQS

GW2A/GW2AR 系列 FPGA 产品的 DQS 模块提供了如下的功能来支持 DDR 存储器接口的时钟需求：

- 接收时钟输入，整理波形并移动 1/4 相位；
- 为输入缓存提供读/写指针；
- 为内部逻辑提供数据有效信号；
- 提供 DDR 输出时钟信号；
- 支持 DDR3 写电压控制。

有 3 种工作模式，用来满足不同的 IO 接口的需求，如图 4-6 所示。

图 4-6 DQS 示意图



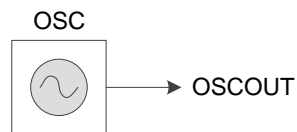
5_{OSC}

5.1 概述

高云半导体 **FPGA** 产品内嵌了一个可编程片内晶振, 为用户提供低抖动、高精度的时钟频率资源, 支持 2.1MHz 到 125MHz 的时钟频率范围, 精度为 $\pm 5\%$ 。

5.2 功能框图

图 5-1 OSC 示意图



5.3 原语

VHDL

```
COMPONENT OSC
  GENERIC (FREQ_DIV : integer := 128
    );
  PORT (
    OSCOUT : out std_logic
  );
end COMPONENT;
```

Verilog

```
module OSC ( OSCOUT );
  parameter FREQ_DIV = 128;
  output OSCOUT;
endmodule
```

5.4 端口

表 5-1 OSC 端口信号

端口	描述
OSCOUT	时钟输出

5.5 参数

表 5-2 OSC 参数描述

参数	描述	默认
FREQ_DIV	分频系数，范围为 2~128，只支持偶数	128

5.6 频率值

片内晶振为用户设计提供时钟源，通过配置工作参数，可以获得多达 64 种时钟频率。输出时钟频率可以通过如下公式计算得到：

$$\text{OSCOUT} = 250\text{MHz} / \text{FREQ_DIV}。$$

其中除数 FREQ_DIV 为配置参数，范围为 2~128，只支持偶数。

6_{DLL}

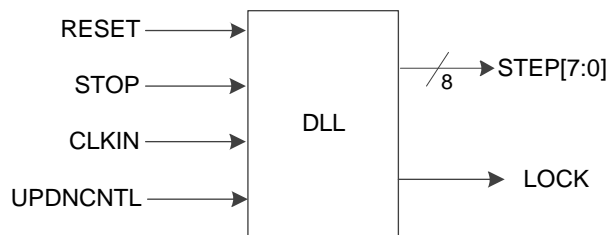
6.1 概述

高云半导体 FPGA 产品提供了延迟锁相环模块 DLL。DLL 模块实现时钟参考延时,通过输出的 STEP 信号送到 DLLDLY 模块中,产生满足 IDDR、IDES4/8/10、IVideo 等模块时延要求的 PCLK 时钟。

DLL 的时钟输入来源包括 GCLK 和相邻的 HCLK。

6.2 功能框图

图 6-1 DLL 示意图



6.3 原语

6.3.1 DLL 原语

VHDL

```

COMPONENT DLL
  GENERIC(
    DLL_FORCE : integer := 0;
    CLKINSEL  : STRING := "00001";
    DIV_SEL   : bit := '1';
    CODESCAL : STRING := "000";
    SCAL_EN   : STRING := "true"
  );
  PORT(
    CLKIN:IN std_logic:= '0';
  
```

```

        STOP: In std_logic:= '0';
        RESET : In std_logic:= '0';
        UPDNCNTL : In std_logic:= '0';
        LOCK : OUT std_logic;
        STEP : OUT std_logic_vector(7 downto 0)
    );
end COMPONENT;

```

Verilog

```

module DLL (HCLKIN, CLKIN, STOP, RESET, UPDNCNTL, STEP, LOCK);
input CLKIN;
input STOP;
input UPDNCNTL;
input RESET;
output [7:0]STEP;
output LOCK;
parameter DLL_FORCE = 0;
parameter CLKINSEL="00001";
parameter CODESCAL="000";
parameter SCAL_EN="true";
parameter DIV_SEL = 1'b0;
endmodule

```

6.3.2 DLLDLY 原语

VHDL

```

COMPONENT DLLDLY
    GENERIC(
        DLL_INSEL : bit := '0';
        DLY_SIGN : bit := '0';
        DLY_ADJ : STRING := "00000000"
    );
    PORT(
        DLLSTEP : IN std_logic_vector(7 downto 0);
        CLKIN:IN std_logic;
        DIR,LOADN,MOVE: In std_logic;
        CLKOUT : OUT std_logic;
        FLAG : OUT std_logic
    );
end COMPONENT;

```

Verilog

```

module DLLDLY (CLKIN, DLLSTEP, DIR, LOADN, MOVE, CLKOUT, FLAG);
input CLKIN;
input [7:0] DLLSTEP;
input DIR,LOADN,MOVE;
output CLKOUT;
output FLAG;
parameter DLL_INSEL = 1'b0;
parameter DLY_SIGN = 1'b0;
parameter DLY_ADJ = "00000000";
endmodule

```

6.4 端口

6.4.1 DLL 端口

表 6-1 DLL 端口信号

端口	方向	描述
CLKIN	I	来自全局时钟输入
STOP	I	时钟控制信号： 0: 启用输入时钟和内部震荡时钟 1: 关断输入时钟和内部震荡时钟
RESET	I	DLL 复位信号： 1: 复位 DLL 0: 不复位
UPDNCNTL	I	更新控制信号： 0: 更新 1: 保持
STEP	O	DLL 延时调整量
LOCK	O	锁定指示信号： 1: 锁定 0: 未锁定

6.4.2 DLLDLY 端口

表 6-2DLLDLY 端口信号

端口	描述
DLLSTEP	时延调整量
CLKIN	输入时钟
DIR	调整方向，在 MOVE 下降沿生效 1: 向前调整 0: 向后调整
LOADN	加载控制： 0: 加载 DLLSTEP 数据，在 MOVE 下降沿生效 1: 无效
MOVE	移入 DLLSTEP、DIR 数据，下降沿生效
CLKOUT	输出时钟
FLAG	当 DLLSTEP 数据加微调值总和超出 255 时产生溢出标志。 1: 溢出 0: 正常

6.5 参数

6.5.1 DLL 参数

表 6-3 DLL 参数描述

参数	描述	默认
DLL_FORCE	DLL 强制模式： 1: 强制锁定，step 值为“FF” 0: DLL loop 锁定	0
CODESCAL	000: 相移 101° 001: 相移 112° 010: 相移 123° 011: 相移 135° 100: 相移 79° 101: 相移 68° 110: 相移 57° 111: 相移 45°	000
SCAL_EN	SCAL 使能信号： true: CODESCAL 启用 false: CODESCAL 无效，固定输出 90° 相移	true
DIV_SEL	锁定模式： 1: 快速锁定 0: 正常锁定	0

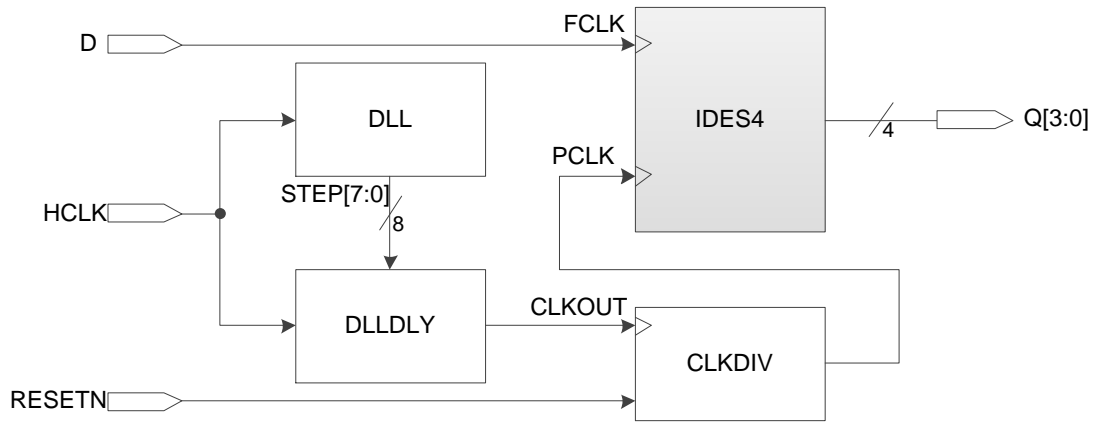
6.5.2 DLLDLY 参数

表 6-4 DLLDLY 参数描述

参数	描述	默认
DLL_INSEL	旁路控制： 0: 旁路 DLL_DLY 1: 启用 DLL_DLY	0
DLY_SIGN	微调符号位： 0: 表示“+” 1: 表示“-”	0
DLY_ADJ	微调值： 当 DLY_SIGN=0 00000000: +0 00000001: +1 11111111: +255 当 DLY_SIGN=1 00000000: -256 00000001: -255 11111111: -1	00000000

6.6 应用示意图

图 6-2 DLL 应用示意图



7 PLL

7.1 概述

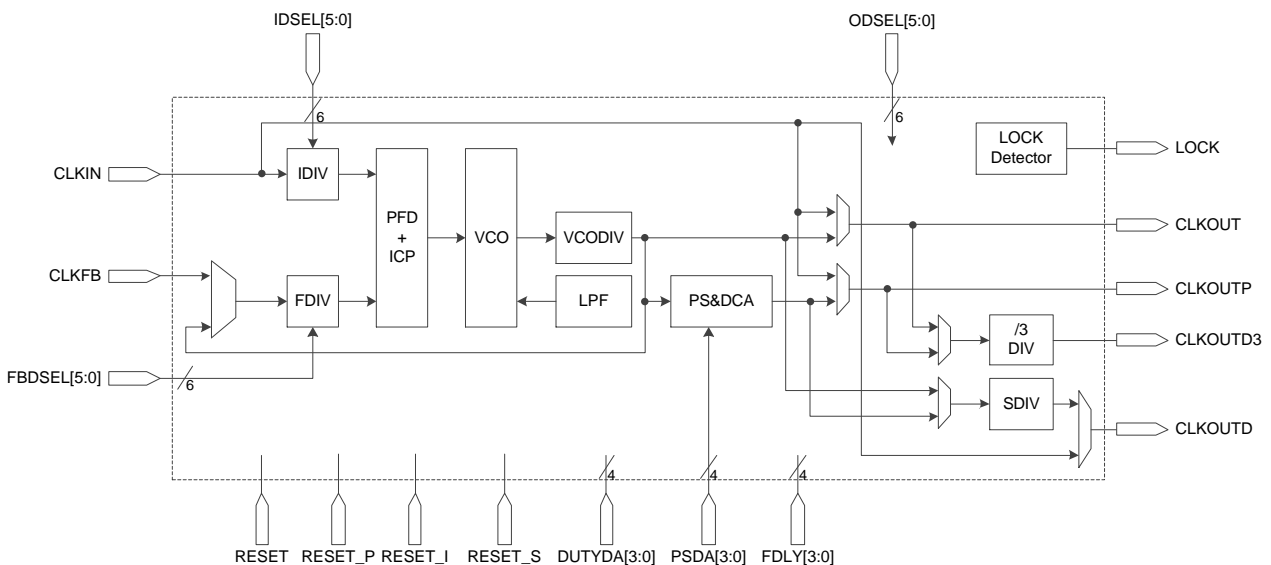
锁相环路是一种反馈控制电路,简称锁相环(PLL, Phase-locked Loop)。利用外部输入的参考时钟信号控制环路内部振荡信号的频率和相位。

PLL 模块能够提供可以综合的时钟频率,通过配置不同的参数可以进行时钟的频率调整(倍频和分频)、相位调整、占空比调整等功能。

PLL 的参考时钟信号可以通过外部 PLL 时钟管脚输入,也可以是通过绕线过去的全局时钟信号、高速时钟信号或普通数据信号。PLL 的反馈信号可以是外部 PLL 反馈信号的管脚的输入,也可以是通过绕线过去的全局时钟信号、高速时钟信号或普通数据信号。

7.2 功能框图

图 7-1 PLL 示意图



7.3 原语

VHDL

COMPONENT PLL

GENERIC(

```

FCLKIN : STRING := "100.0"; --frequency of the clkIn(M)
DEVICE : STRING := "GW1N2K";
--"GW1N1K","GW1N2K","GW1N4K","GW1N6K","GW1N9K"
DYN_IDIV_SEL : STRING := "false";
--true:IDSEL; false:IDIV_SEL
IDIV_SEL : integer := 0;
--Input divider IDIV, 0:1,1:2...63:64. 1~64
DYN_FBDIV_SEL : STRING := "false";
FBDIV_SEL : integer := 0;
--Feedback divider FBDIV, 0:1,1:2...63:64. 1~64
DYN_ODIV_SEL : STRING := "false";
--true:ODSEL; false:ODIV_SEL
ODIV_SEL : integer := 8;--2/4/8/16/32/48/64/80/96/112/128
PSDA_SEL : STRING := "0000";
DYN_DA_EN : STRING := "false";
--true:PSDA or DUTYDA or FDA; false: DA_SEL
DUTYDA_SEL : STRING := "1000";
CLKOUT_FT_DIR : bit := '1';
-- CLKOUT fine tuning direction. '1' only
CLKOUTP_FT_DIR : bit := '1'; -- '1' only
CLKOUT_DLY_STEP : integer := 0; -- 0,1,2,4
CLKOUTP_DLY_STEP : integer := 0; -- 0,1,2
CLKOUTD3_SRC : STRING := "CLKOUT";
--select div3 output, CLKOUTP or CLKOUT
CLKFB_SEL : STRING := "internal";
CLKOUT_BYPASS : STRING := "false";
CLKOUTP_BYPASS : STRING := "false";
CLKOUTD_BYPASS : STRING := "false";
CLKOUTD_SRC : STRING := "CLKOUT";
--select div output, CLKOUTP or CLKOUT
DYN_SDIV_SEL : integer := 2 -- 2~128,only even num

```

```
);
```

PORT(

```
CLKIN : in std_logic_vector;
```

```

    CLKFB    : in std_logic;
    ODSEL    : in std_logic_vector(5 downto 0);
    IDSEL    : in std_logic_vector(5 downto 0);
    FBDSEL   : in std_logic_vector(5 downto 0);
    RESET    : in std_logic: ='0';
    RESET_P  : in std_logic: ='0';
    RESET_I  : in std_logic: ='0';
    RESET_S  : in std_logic : ='0';
    PSDA, FDLY : in std_logic_vector(3 downto 0);
    DUTYDA   : in std_logic_vector(3 downto 0);
    LOCK     : out std_logic;
    CLKOUT   : out std_logic;
    CLKOUTD  : out std_logic;
    CLKOUTP  : out std_logic;
    CLKOUTD3: out std_logic

```

```
);
```

```
end COMPONENT;
```

Verilog

```

module PLL (CLKOUT, CLKOUTP, CLKOUTD, CLKOUTD3, LOCK, CLKIN,
            CLKFB, FBDSEL, IDSEL, ODSEL, DUTYDA, PSDA, FDLY, RESET, RESET_P,
            RESET_I, RESET_S);
    input CLKIN;
    input CLKFB;
    input RESET;
    input RESET_P;
    input RESET_I;
    input RESET_S;
    input [5:0] ODSEL;
    input [5:0] FBDSEL;
    input [5:0] IDSEL;
    input [3:0] PSDA,FDLY;
    input [3:0] DUTYDA;
    output CLKOUT;
    output LOCK;
    output CLKOUTP;
    output CLKOUTD;
    output CLKOUTD3;

```

```

parameter FCLKIN = "100.0"; // frequency of the clkin(M)
parameter DYN_IDIV_SEL= "false";//true:IDSEL; false:IDIV_SEL
parameter IDIV_SEL = 0; // 0:1,1:2...63:64. 1~64
parameter DYN_FBDIV_SEL= "false";
parameter FBDIV_SEL = 0; // 0:1,1:2...63:64. 1~64
parameter DYN_ODIV_SEL= "false";//true:ODSEL; false:ODIV_SEL
parameter ODIV_SEL = 8; // 2/4/8/16/32/48/64/80/96/112/128
parameter PSDA_SEL= "0000";//
parameter DYN_DA_EN = "false";
//true:PSDA or DUTYDA or FDA; false: DA_SEL
parameter DUTYDA_SEL= "1000";//
parameter CLKOUT_FT_DIR = 1'b1; // CLKOUT fine tuning direction. 1'b1 only
parameter CLKOUTP_FT_DIR = 1'b1; // 1'b1 only
parameter CLKOUT_DLY_STEP = 0; // 0,1,2,4
parameter CLKOUTP_DLY_STEP = 0; // 0,1,2
parameter CLKFB_SEL = "internal";//"internal","external";
parameter CLKOUT_BYPASS = "false"; // "true"; "false"
parameter CLKOUTP_BYPASS = "false"; // "true"; "false"
parameter CLKOUTD_BYPASS = "false"; // "true"; "false"
parameter DYN_SDIV_SEL = 2; // 2~128,only even num
parameter CLKOUTD_SRC = "CLKOUT"; //CLKOUT,CLKOUTP
parameter CLKOUTD3_SRC = "CLKOUT"; //CLKOUT,CLKOUTP
parameter DEVICE = "GW1N2K";
//"GW1N1K","GW1N2K","GW1N4K","GW1N6K","GW1N9K"
endmodule

```

7.4 端口

表 7-1 PLL 端口信号

端口名称	方向	描述
CLKIN	I	参考时钟输入
CLKFB	I	反馈时钟输入
RESET	I	PLL 全部复位
RESET_P	I	PLL 关断 (Power Down) 信号
RESET_I	I	IDIV 复位信号
RESET_S	I	SDIV 和 DIV3 复位信号
ODSEL[5: 0]		动态控制 ODIV 值, 范围 1~64
IDSEL [5: 0]	I	动态控制 IDIV 值, 范围 1~64
FBDSEL [5: 0]	I	动态控制 FDIV 值, 范围 1~64
PSDA [3: 0]	I	动态相位控制(上升沿有效)

DUTYDA [3: 0]	I	动态占空比控制(下降沿有效)
FDLY [3: 0]	I	CLKOUTP 动态延迟控制
CLKOUT	O	无相位和占空比调整的时钟输出
CLKOUTP	O	有相位和占空比调整的时钟输出
CLKOUTD	O	来自 CLKOUT 或 CLKOUTP 分频时钟 (由 SDIV 分频器控制)
CLKOUTD3	O	来自 CLKOUT 或 CLKOUTP 的分频时钟 (由 DIV3 分频器控制, DIV3 分频值固定为 3)
LOCK	O	PLL 锁定指示: 1: 锁定 0: 失锁

7.5 参数

表 7-2 PLL 参数

参数	描述
FCLKIN	时钟 CLKIN 的频率值, 类型 STRING=100.0 (默认 100MHz);
DYN_ODIV_SEL	类型 STRING=false; false: 使用参数输入 ODIV_SEL 值; true: 使用参数输入端口输入 ODSEL 值。
DEVICE	类型 STRING=GW1N2K; "GW1N1K", "GW1N2K", "GW1N4K", "GW1N6K", "GW1N9K", "GW2A18", "GW2A55"
DYN_IDIV_SEL	类型 STRING=false; false: 使用参数输入 IN_SEL 值; true: 使用端口输入 INSEL 值。
IDIV_SEL	类型: integer=0; IDIV_SEL 参数值输入范围 0~63; IDIV 有效值在参数输入值基础上加 1, 范围 1~64。
DYN_FBDIV_SEL	类型 STRING= false; false: 使用参数输入 FBDIV_SEL 值; true: 使用端口输入 FBDSEL 值。
FBDIV_SEL	类型 integer=0; FBDIV_SEL 参数值输入范围 0~63; FDIV 有效值在参数输入值基础上加 1, 范围 1~64。
DYN_SDIV_SEL	类型 integer = 2; 取值范围 2~128, 仅能输入偶数值。
ODIV_SEL	ODIV 参数值, 类型 integer = 8; 可选数值: 2、4、8、16、32、48、64、80、96、112、128。
PSDA_SEL	相位调整参数设置, 类型 STRING = 0000; 取值范围 0000~1111(默认 0000)。
DUTYDA_SEL	占空比调整参数设置, 类型 STRING=0000; 取值范围 0000~1111(默认 0000)。
DYN_DA_EN	类型 STRING= false; false: 选择参数 PSDA_SEL、DUTYDA_SEL 作为相位占空比调整信号; true: 选择端口 PSDA、DUTYDA、FDLY 等作为相位占空比控制信号。

参数	描述
CLKOUT_FT_DIR	类型 bit=1; CLKOUT 微调方向: 0: + 1: -
CLKOUT_DLY_STEP	类型 integer= 0, 取值 0、1、2、4; CLKOUT 微调值: 参数值 x delay(delay=50ps)。
CLKOUTP_FT_DIR	CLKOUTP 微调方向: 1'b1: - 1'b0: +
CLKOUTP_DLY_STEP	类型 integer= 0, 取值 0、1、2; CLKOUTP 微调值: 参数值 x delay(delay=50ps)。
CLKFB_SEL	类型 STRING= CLKOUT; CLKOUT: 选择内部反馈; CLKFB: 选择端口 CLKFB 的外部反馈。
CLKOUTD_SRC	CLKOUTD 的源信号, 类型 STRING =CLKOUT; CLKOUT: 选择 CLKOUT; CLKOUTP: 选择 CLKOUTP。
CLKOUTD3_SRC	CLKOUTD3 的源信号, 类型 STRING = CLKOUT; CLKOUT: 选择 CLKOUT; CLKOUTP: 选择 CLKOUTP。
CLKOUT_BYPASS	CLKOUT 时钟旁路控制, 类型 STRING =false; true: 旁路 PLL, CLKOUT= CLKIN; false: 使能 CLKOUT 输出
CLKOUTP_BYPASS	CLKOUTP 时钟旁路控制, 类型 STRING =false; true: 旁路 PLL, CLKOUT= CLKIN; false: 使能 CLKOUTP 输出。
CLKOUTD_BYPASS	CLKOUTD 时钟旁路控制, 类型 STRING =false; true: 旁路 PLL, CLKOUT= CLKIN; false: 使能 CLKOUTD 输出。

表 7-3 PLL 相位参数调整对照表

参数 PSDA_SEL 或端口 PSDA 设置	相位调整
0000	0°
0001	22.5°
0010	45°
0011	67.5°
0100	90°
0101	112.5°
0110	135°
0111	157.5°
1000	180°
1001	202.5°
1010	225°
1011	247.5°

参数 PSDA_SEL 或端口 PSDA 设置	相位调整
1100	270°
1101	292.5°
1110	315°
1111	337.5°

表 7-4 PLL 占空比参数调整对照表

参数 DUTYDA_SEL 或端口 DUTYDA 设置	占空比设置值 (/16)
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

占空比调整需要参考相移设置。例如，当相移设置为“0”（0000）时，50%占空比设置为“8”（1000）。如果相移的设置是“180°”，然后50%占空比的设置将被“0”（0000）。

下面的方程将帮助用户正确选择占空比设置：

- 若 $DUTY[3:0] > PHASE[3:0]$ 时， $DutyCycle = 1/16 \times (DUTY[3:0] - PHASE[3:0])$ 。
- 若 $DUTY[3:0] < PHASE[3:0]$ 时， $DutyCycle = 1/16 \times (16 + DUTY[3:0] - PHASE[3:0])$ 。

可以通过端口 PASEL[3:0] 动态控制输出时钟 CLKOUTP 的延迟。每一步增加一个 0.125ns，共 15 步（1.875ns）。需要结合相移设置实现滞后（时钟信号 CLKOUTP 滞后于输入时钟）和超前（时钟信号 CLKOUTP 超前输入时钟）。

表 7-5 PLL 延迟参数调整对照表

端口 PASEL[3:0]	延迟步数
0000	0(默认)
0001	1
0010	2
0011	3
0100	4

端口 PASEL[3: 0]	延迟步数
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

7.6 频率计算

PLL 可对输入 CLKIN 进行频率调整（倍频和分频），计算公式如下：

- $F_{CLKOUT} = (F_{CLKIN} \times FDIV) / IDIV$
- $F_{VCO} = F_{CLKOUT} \times ODIV$
- $F_{CLKOUTD} = F_{CLKOUT} / SDIV$
- $F_{PDF} = F_{CLKIN} / IDIV = F_{CLKOUT} / FDIV$

注！

- FCLKIN 为输入时钟 CLKIN。
- FCLKOUT 作为 CLKOUT 和 CLKOUTP 时钟。
- FCLKOUTD 为 CLKOUTD 时钟。
- FPDF 为 PDF 鉴相频率。

即可通过调整 IDIV、FDIV、ODIV、SDIV 来得到期望频率的时钟信号。

7.7 举例

若要得到输入时钟1.75倍频的时钟，可设置IDIV_SEL=1（输入分频系数为2），FBDIV_SEL=6（FB分频系数为7），DYN_SDIV_SEL=2（SDIV分频系数为2），根据分频公式可得CLKOUT的频率为CLKIN的3.5倍，CLKOUTD的频率为CLKIN的1.75倍。

Verilog语言的模块调用：

```
.PLL#(
.FCLKIN("100.0"),
.DYN_IN_SEL("false"),
.IN_SEL(5),
.DYN_IDIV_SEL("false"),
```

```
.IDIV_SEL(1),
.DYN_FBDIV_SEL("false"),
.FBDIV_SEL(6),
.ODIV_SEL(8),
.PSDA_SEL (4b"0000"),
.DYN_DA_EN ("false"),
.DUTYDA_SEL(4b"1000"),
.CLKOUT_FT_DIR(1b'1'),
.CLKOUTP_FT_DIR(1b'1'),
.CLKOUT_DLY_STEP(0),
.CLKOUTP_DLY_STEP(0),
.CLKOUTD3_SRC("CLKOUT"),
.CLKFB_SEL(CLKOUT"),
.CLKOUT_BYPASS("false"),
.CLKOUTP_BYPASS("false"),
.CLKOUTD_BYPASS("false"),
.CLKOUTD_SRC(CLKOUT"),
.DYN_SDIV_SEL(2)
)

PLL_inst(
.CLKIN(CLKIN),
.CLKFB(CLKFB),
.INSEL(INSEL),
.IDSEL(IDSEL),
.FBDSEL(FBDSEL),
.RESET(RESET),
.RESET_P(RESET_P),
.RESET_I(RESET_I),
.RESET_S(RESET_S),
.PSDA(PSDA),
.FDLY(FDLY),
.DUTYDA(DUTYDA),
.LOCK(LOCK),
.CLKOUT(CLKOUT),
```

```
.CLKOUTD(CLKOUTD),  
.CLKOUTP(CLKOUTP),  
.CLKOUTD3(CLKOUTD3);
```

