

**可编程逻辑器件应用**

**自学手册**

软件与通信学院编制

2021年02月

# 项目一 格雷码计数器

## 一、设计目的

该项目要求通过Verilog HDL编程实现一个格雷码计数器。通过该项目的实施，要求学生掌握格雷码的编码原理以及与二进制之间的转换关系。

## 二、设计要求

1. 设计标准

该项目主要实现一个基于格雷码的四位计数器。模块功能应包括二进制码计数器的产生、由二进制码产生格雷码、所产生的格雷码再次转换为二进制码等逻辑部分。

该项目由一个功能模块和一个testbench组成。其中功能模块的端口信号如下表所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 端口名 | 端口方向 | 端口宽度 | 功能描述 |
| 1 | clk\_50M | input | 1 | 50M主时钟 |
| 2 | rst\_n | input | 1 | 复位信号，低有效 |
| 3 | gray\_dout | output | 4 | 格雷码计数器的当前计数值输出 |
| 4 | bin\_dout | output | 4 | 由计数格雷码值转换得到的二级制码输出 |

1. 设计结果

自学报告中提供：

（1）功能模块代码

（2）Testbench代码

（3）相关信号的仿真波形（一个完整总体波形+若干个细节波形图，可看到中间结果）

# 项目二 四分频的分频器

## 一、设计目的

该项目要求通过Verilog HDL编程实现一个主时钟四分频的分频器。通过该项目的实施，要求学生掌握分频器的工作原理，以及占空比的设计思想。

## 二、设计要求

1. 设计标准

该项目主要实现一个占空比为50%的时钟四分频分频器。模块功能应包括主时钟计数、分频时钟翻转等逻辑部分。

该项目由一个功能模块和一个testbench组成。其中功能模块的端口信号如下表所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 端口名 | 端口方向 | 端口宽度 | 功能描述 |
| 1 | clk\_50M | input | 1 | 50M主时钟 |
| 2 | rst\_n | input | 1 | 复位信号，低有效 |
| 3 | clk\_div | output | 1 | 四分频时钟输出 |

1. 设计结果

自学报告中提供：

（1）功能模块代码

（2）Testbench代码

（3）相关信号的仿真波形（一个完整总体波形+一个细节波形图，可看到中间结果）

# 项目三 8\*8同步FIFO

## 一、设计目的

该项目要求通过Verilog HDL编程实现一个深度为8个字节的同步先入先出存储单元（FIFO）。通过该项目的实施，要求学生掌握FIFO的工作原理，以及同步设计思想。

## 二、设计要求

1. 设计标准

该项目主要实现一个深度为8、位宽为8bit的同步FIFO存储单元。模块功能应包括读控制、写控制、同时读写控制、FIFO满状态、FIFO空状态等逻辑部分。

该项目由一个功能模块和一个testbench组成。其中功能模块的端口信号如下表所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 端口名 | 端口方向 | 端口宽度 | 功能描述 |
| 1 | clk\_50M | input | 1 | 50M主时钟 |
| 2 | rst\_n | input | 1 | 复位信号，低有效 |
| 3 | wr | input | 1 | FIFO写使能信号，高有效 |
| 4 | rd | input | 1 | FIFO写使能信号，高有效 |
| 5 | din | input | 8 | 写入FIFO的数据 |
| 6 | dout | output | 8 | 读出FIFO的数据 |
| 7 | full | output | 1 | FIFO满状态信号  1：FIFO写满  0：FIFO未写满 |
| 8 | almost\_full | output | 1 | FIFO快满状态信号，  1：FIFO快写满了  0：FIFO未快写满 |
| 9 | empty | output | 1 | FIFO空状态信号，  1：FIFO读空  0：FIFO未读空 |
| 10 | almost\_empty | output | 1 | FIFO快空状态信号，  1：FIFO快读空了  0：FIFO未快读空了 |
| 11 | half | output | 1 | FIFO半满（半空）状态信号，  1：FIFO半满（半空）  0：FIFO未半满（半空） |

**提示：**

（1）设定一个写地址waddr[2:0]，每来一个写使能wr，则写地址加一；

（2）设定一个读地址raddr[2:0]，每来一个读使能rd，则读地址加一；

（3）若同时读写，则读写地址不变，只是将输入数据din直接赋值给dout;

（4）FIFO中数据的个数cout=waddr-raddr;

（5）状态指示：

* 当cout=8时，FIFO满（full=1）;
* 当cout=6时，FIFO快满了（almost\_full=1）;
* 当cout=4时，FIFO半满（半空）（half=1）;
* 当cout=2时，FIFO快空了（almost\_empty=1）;
* 当cout=0时，FIFO空（empty=1）;

1. 设计结果

自学报告中提供：

（1）功能模块代码

（2）Testbench代码

（3）相关信号的仿真波形（一个完整总体波形+若干个细节波形图（读、写、同时读写、各个状态分别有效等），可看到中间结果）