

实例学习 Robei 芯片设计系列

三. 编译码器

Robei LLC

1. 实验目的

通过设计简单的编译码器实现对数据的转换。常见的编码方式有格雷码，BCD 码和 8-3 线编码器，16-4 线编码器等。本次实验以 8-3 线优先编码和 3-8 线译码器作为例子，进行数据的编译码实验。

2. 实验准备

优先编码器是将多个二进制输入压缩成更少数目输出的电路算法。优先编码器常用于处理最高优先级请求时控制中断请求。8-3 线编码器是将输入为 8 比特的数据以 3 比特的方式描述出来。8 根输入线路中每次只有一个线路为高电平，其余为低电平。相反，3-8 译码器是用 8 根线对输入的 3 根线数据进行电平转换。

x[7]	x[6]	x[5]	x[4]	x[3]	x[2]	x[1]	x[0]	y[2]	y[1]	y[0]
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

表 1. 8-3 编码器真值表

3. 实验内容

3.1 编码器模型设计

- 1) 新建一个模型命名为 `encoder`，类型为 `module`，同时引脚设置为 2 输入 1 输出。每个引脚的属性和名称参照图 1 进行对应的修改。

Name	Inout	Data Type	Datasize	Function
x	input	wire	7:0	input data
en	input	wire	1	enable
y	output	reg	2:0	output

图 1. 引脚属性

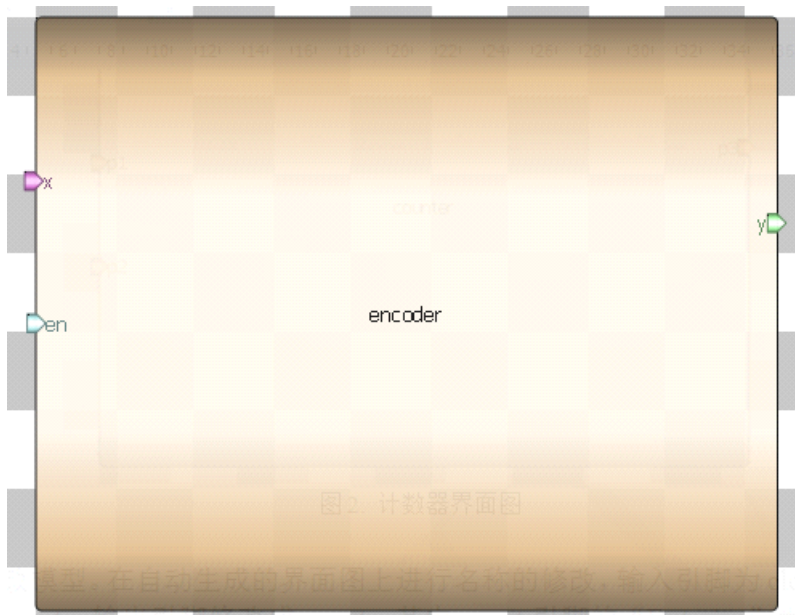


图 2. 编码器界面图

2) 添加代码。点击模型下方的 Code（如图 3 所示）添加代码。

```

encoder
16 always @ (en or x) begin
17   if (en) begin
18     case (x) //学习 case 语句的写法
19       8'b00000001 : y = 3'b000;
20       8'b00000010 : y = 3'b001;
21       8'b00000100 : y = 3'b010;
22       8'b00001000 : y = 3'b011;
23       8'b00010000 : y = 3'b100;
24       8'b00100000 : y = 3'b101;
25       8'b01000000 : y = 3'b110;
26       8'b10000000 : y = 3'b111;
27       default : y = 3'b000; //别忘了设置 default 的值
28     endcase
29   end
30 end

```

图 3. 点击 Code 输入算法

在代码设计区内输入以下 Verilog 代码：

```


always @ (en or x) begin //多个敏感信号的写法
  if(en) begin
    case (x) //学习 case 语句的写法
      8'b00000001 : y = 3'b000;


```

```

8'b00000010 : y = 3'b001;
8'b00000100 : y = 3'b010;
8'b00001000 : y = 3'b011;
8'b00010000 : y = 3'b100;
8'b00100000 : y = 3'b101;
8'b01000000 : y = 3'b110;
8'b10000000 : y = 3'b111;
default : y= 3'b000; //别忘了设置 default 的值
endcase
end
end

```

3) 保存。点击工具栏  图标，或者点击菜单“File”中的下拉菜单“Saveas”，将模型另存到一个文件夹中。

4) 运行。在工具栏点击  或者点击菜单“Build”的下来菜单“Run”，执行代码检查。如果有错误，会在输出窗口中显示。

3.2 译码器模型设计

1) 新建一个模型命名为 decoder，类型为 module，同时引脚设置为 2 输入 1 输出。每个引脚的属性和名称参照图 4 进行对应的修改。

Name	Inout	Data Type	Datasize	Function
din	input	wire	2:0	input data
en	input	wire	1	enable
out	output	wire	7:0	output

图 4. 引脚属性

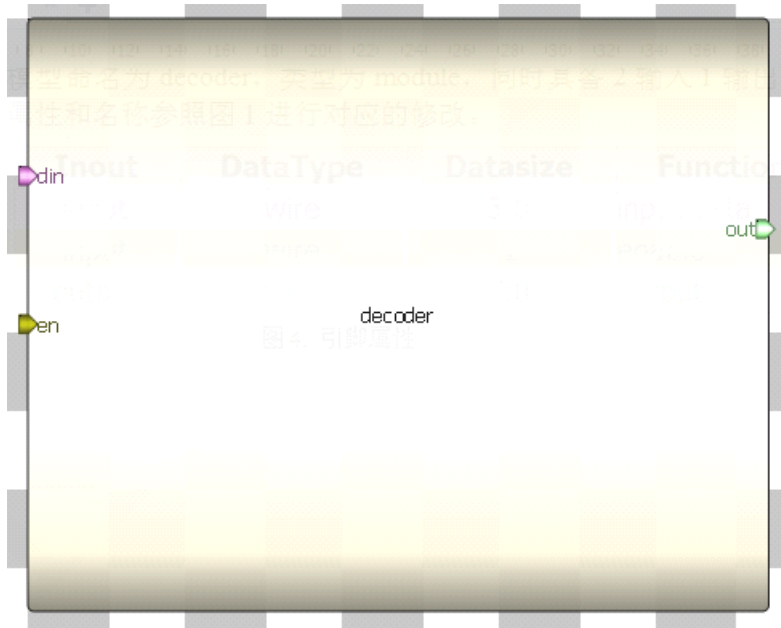


图 5. 译码器界面图

2) 添加代码。点击模型下方的 Code (如图 6 所示) 添加代码。

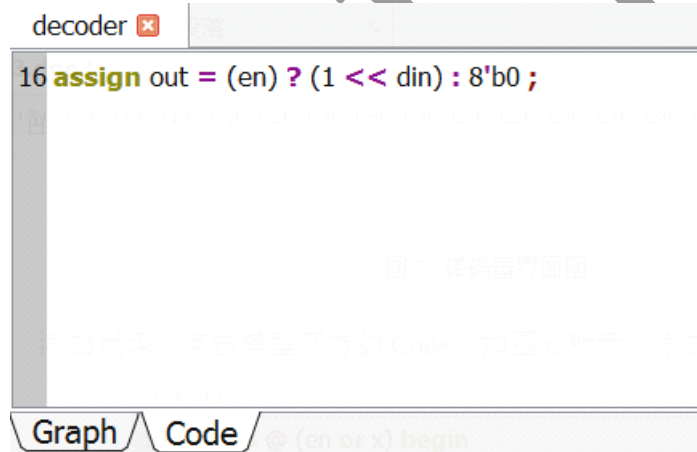




图 6. 点击 Code 输入算法


在代码设计区内输入以下 Verilog 代码:

```
assign out = (en) ? (1 << din) : 8'b0; //如果 en 是 1, 则将 1 右移 din 位, 否则取 0.
```

3) 保存。点击工具栏  图标, 将模型存到与 **encoder** 相同的文件夹中。

4) 运行。在工具栏点击  执行代码检查。如果有错误, 会在输出窗口中显示。

3.2 测试文件设计

- 1) 新建一个 2 输入 1 输出的测试文件。点击工具栏上的  图标，在弹出的对话框中参照图 7 进行设计。

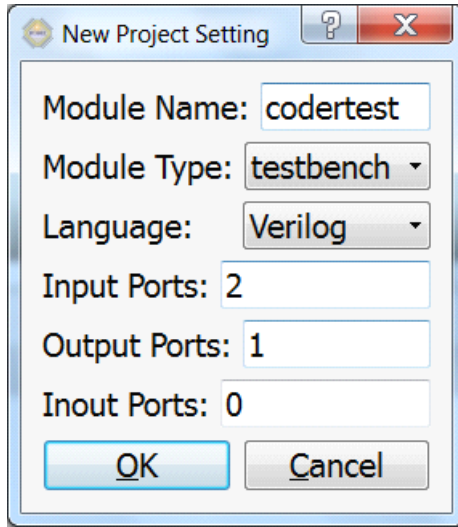



图 7. 新建测试文件

- 2) 修改各个引脚的颜色。选中每个引脚，在属性栏中对照图 8 进行修改引脚属性，并修改其颜色，方便区分不同的引脚信号。

Name	Inout	DataType	Datasize	Function
data	input	wire	2:0	signal data
en	input	wire	1	enable
out	output	wire	2:0	output data

图 8 引脚属性表

- 3) 另存为测试文件。点击工具栏  图标，将测试文件保存到 decoder 和 encoder 模型所在的文件夹下。

- 4) 加入模型。在 Toolbox 工具箱的 Current 栏里，会出现一个 decoder 和 encoder 模型，单击该模型并在 codertest 上添加。点击工具栏中的  图标，连接引脚。

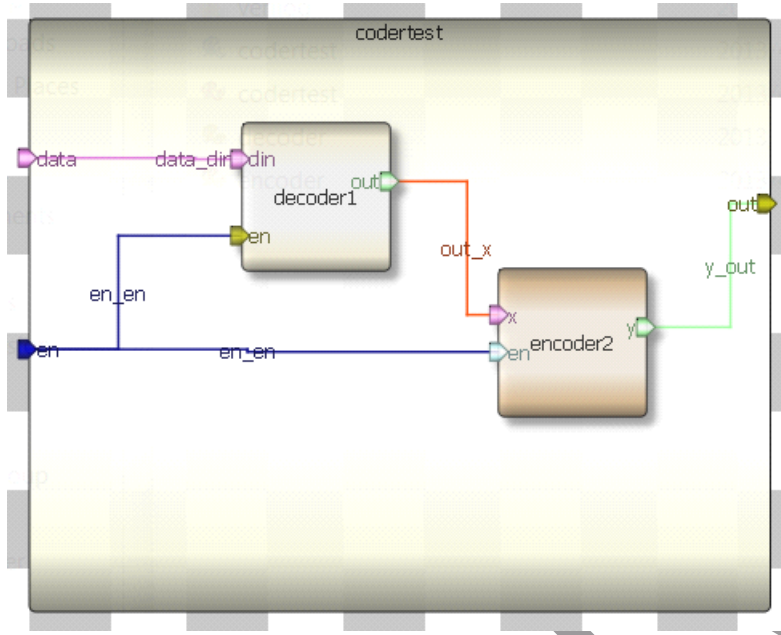


图 9. 添加模型

5) 输入激励。点击测试模块下方的“Code”，输入激励算法。激励代码在结束的时候要用\$finish 结束。

```
initial begin
  data=3'b000;
  en=0;
  #1 data=3'b000;
    en=1;
  #1 data=3'b001;
  #1 data=3'b010;
  #1 data=3'b011;
  #1 data=3'b100;
  #1 data=3'b101;
  #1 data=3'b110;
  #1 data=3'b111;
  #1 data=3'b000;
  #1 data=3'b001;
  #1 $finish;
end
```

```
codertest x
15 initial begin
16 data=3'b000;
17 en=0;
18 #1 data=3'b000;
19 en=1;
20 #1 data=3'b001;
21 #1 data=3'b010;
22 #1 data=3'b011;
23 #1 data=3'b100;
24 #1 data=3'b101;
25 #1 data=3'b110;
26 #1 data=3'b111;
27 #1 data=3'b000;
28 #1 data=3'b001;
29 #1 $finish;
30 end
```

图 10. 激励代码

- 6) 执行仿真并查看波形。点击工具栏 ，查看输出信息。检查没有错误之后点击 。在弹出的 Waveform 窗口上点击右侧 Workspace 中的信号，进行添加并查看。点击波形查看器工具栏上的  图标进行自动缩放。分析仿真结果并对照真值表，查看设计波形输入输出是否一致。

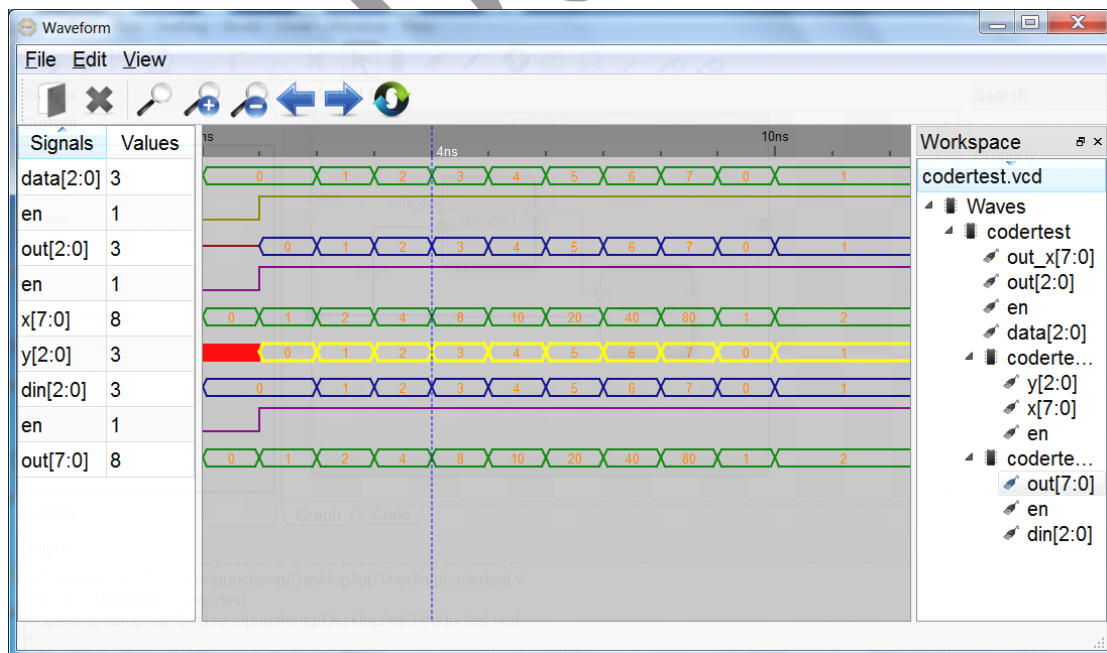


图 11. 查看波形

4.问题与思考

- 1) 使用 Robei 设计一个 BCD 编码器并进行仿真测试。
- 2) 用 Robei 设计一个 16-4 线编码器和 4-16 线译码器，并仿真测试。
- 3) 用 Robei 设计一个格雷编码器并测试结果。

Robei LLC