**Lab1 Readme**

**Lab1\_1 3-8 decoder**

1. 真值表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | | output | | | | | | | |
| A2 | A1 | A0 | Y7 | Y6 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

1. 化简译码器布尔表达式

观察后发现对于每一个输出为1时对应的输入都只有一种,因此不需要再化简,直接将对应的布尔表达式写出即可。

Y0 = A2’A1’A0’ Y1 = A2’A1’A0 Y2 = A2’A1A0’ Y3 = A2’A1A0

Y4 = A2A1’A0’ Y5 = A2A1’A0 Y6 = A2’A1A0 Y7 = A2A1A0

1. 设计思路与方法:

根据以上的分析,写出每个输出的布尔表达式,完成译码器的构造。这种方法的Schematic显示出门级构造。Verilog文件为decoder\_3x8.v。

图示, 示意图

描述已自动生成

仿真结果:

电脑萤幕截图

描述已自动生成

**Lab1\_2 4-2 encoder**

1. 真值表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| input | | | | output | |
| A3 | A2 | A1 | A0 | Y1 | Y0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 |

注:普通编码器一次只允许输入一个信号。

1. 化简译码器布尔表达式

首先根据真值表写出每个输出的最小项之和

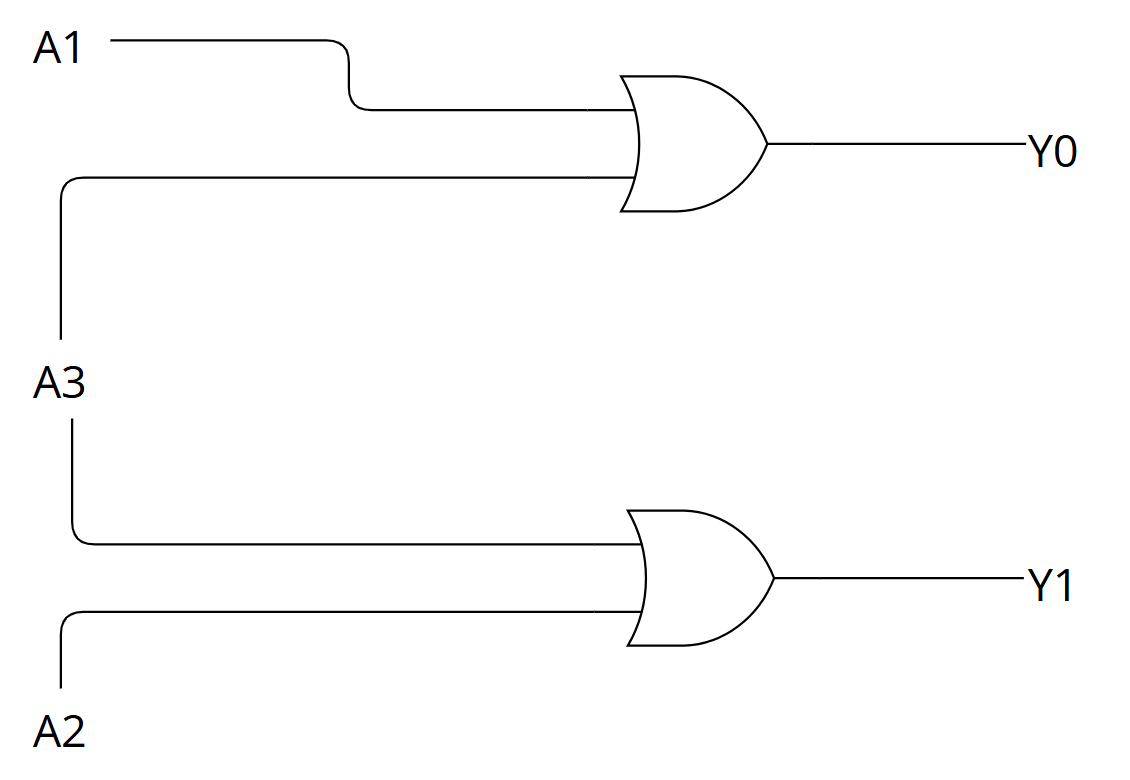
Y0 = A3A2’A1’A0’ + A3’A2’A1A0’

Y1 = A3A2’A1’A0’ + A3’A2A1’A0’

又因为普通编码器一次只允许输入一个信号,表达式可化简为:

Y0 = A3 + A1 Y1 = A3 + A2

根据布尔表达式,其原理图如下



1. 设计思路与方法

运用case语句,将输出Y的四种取值与输入一一对应。除此之外,说明输入不满足一次输入一个信号,作为普通编码器,应当将输出设置为2’bx。Verilog文件为encoder\_4x2.v。

以下是仿真结果图形用户界面

描述已自动生成

RTL schematic:

