**实验** **2：组合逻辑电路设计**

**一、实验目的**

1. 学习组合逻辑电路的设计方法和步骤，掌握译码器、编码器等组合部件的设计方法。

2. 掌握串行加法器设计方法，理解减法和比较运算的实现方法。

3. 掌握汉明码校验电路的设计方法。

4. 掌握桶形移位器的设计方法。

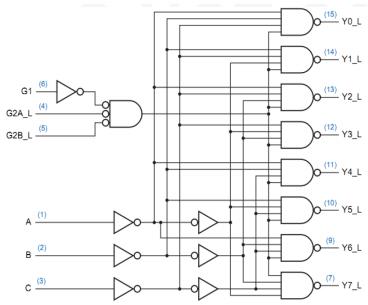
**二、实验环境**

Logisim 2.16

**三、实验内容**

**1. 译码器实验**

根据如图 2.1 所示的 3-8 译码器芯片 74X138 的电路原理图，设计一个由反相逻辑门电路构成的 3-8 译 码器，并对电路进行仿真测试，以验证电路的功能。



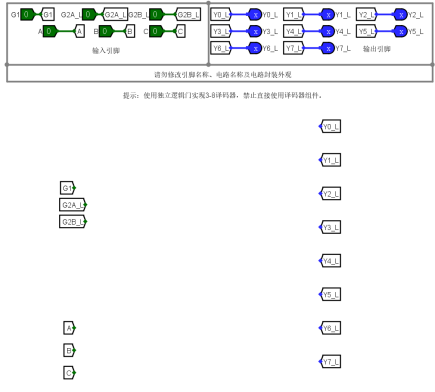
|  |
| --- |
|  |

图 2.1 3-8 译码器 74X138 原理图

实验步骤如下：

1） 根据图 2.2 所示布局图添加逻辑门。打开 Logisim ，选择 8 个 4 输入与非门、7 个非门、1 个与

门、7 个输入引脚、8 个输出引脚，并将上述元件布局到Logisim 的工作区中适当位置。可通过以下方式设 置输入端口数：以与非门为例，选择某个与非门，在属性窗口的输入端口数输入框中设置数字 4。



|  |
| --- |
|  |

图 2.2 3-8 译码器布局图

2） 添加连线。将输入引脚、逻辑门的输入端、输出端、输出引脚等通过连接线相连。

3） 添加标识符。选中输入/输出引脚，在属性表中添加引脚标识符；选中逻辑门，在属性表中添加逻 辑门标识符；点击快捷工具栏中的文本工具，在电路空白处添加电路的描述文字。标识符、注释文字的字体、 大小、颜色和位置等均可在属性表中修改。

4） 仿真测试。进入仿真状态，改变输入引脚赋值，记录输出引脚值，填写表 2.1 所示的译码器功能 表以验证实验结果。保存电路，文件名为 lab2.1.circ。

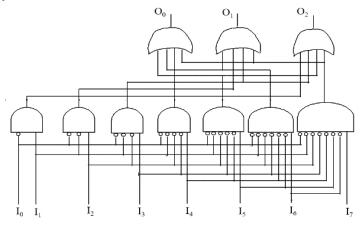
表 12.1 74X138 功能表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 输 入 | | | | | | 输 出 |
| G1 G2A\_L G2B\_L | | | C B A | | | Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0 |
| X X 0 1 1 1 1 1 1 | 1 X X 0 0 0 0 0 0 | X 1 X 0 0 0 0 0 0 | X X X 0 0 0 0 1 1 | X X X 0 0 1 1 0 0 | X X X 0 1 0 1 0 1 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1  1 | 0  0 | 0  0 | 1  1 | 1  1 | 0  1 |  |

**2. 编码器实验**

根据如图 2.3 所示的 8-3 优先级编码器原理图，设计一个由逻辑门电路构成的 8-3 优先级编码器，并将 编码器输出连接到一个十六进制数码管，通过数码管的输出显示来验证和测试电路。



|  |
| --- |
|  |

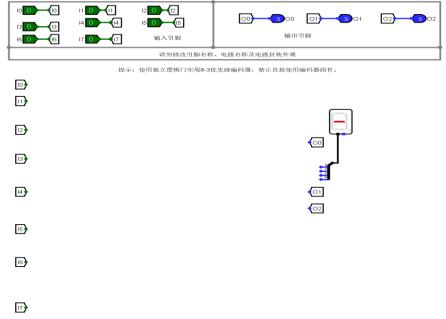
图 2.3 8-3 优先级编码器原理图

实验步骤如下：

1） 根据图 2.4 所示的布局图添加逻辑门。在 Logisim 的在工作区中放置与门、或门、输入引脚、分 线器、16 进制数码管等。将或门输入端口数属性改为 4，每个与门的输入端口数属性改为原理图所示个 数，并修改输入引脚的极性（是否反转）。

2） 添加线路。将输入引脚、逻辑门的输入端、输出端、输出引脚等通过连接线相连。

3） 添加标识符。选中输入/输出引脚，在属性表中添加引脚标识符；选中逻辑门，在属性表中添加门 标识符；点击快捷工具栏中文本工具，在电路空白处添加描述文字。



|  |
| --- |
|  |

图 2.4 8-3 优先级编码器电路布局图

4） 仿真测试。进入仿真状态，改变输入引脚赋值，记录输出引脚值，填写表 2.2 所示的优先级编码 器功能表，以验证实验结果。保存电路设计文件 lab2.2.circ。

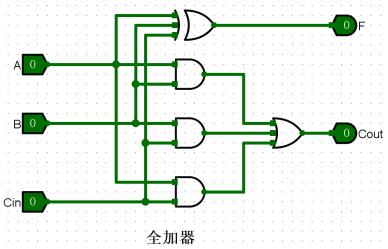
表 2.2 8-3 优先级编码器功能表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输 入 | | | | | | | | 输 出 | |
| I0 I1 I2 I3 I4 I5 I6 I7 | | | | | | | | O0 O1 O2 | Hex 显示 |
| 1  0  0  0  0  0  0  0 | X 1 0 0 0 0 0 0 | X X 1 0 0 0 0 0 | X X X 1 0 0 0 0 | X X X X 1 0 0 0 | X X X X X 1 0 0 | X X X X X X 1 0 | X X X X X X X 1 |  |  |

**3. 加减法器实验**

串联 4 个全加器子电路实现 4 位串行进位加法器。将加数、被加数和和分别连接到 16 进制数码显示管 进行验证。实验步骤如下：

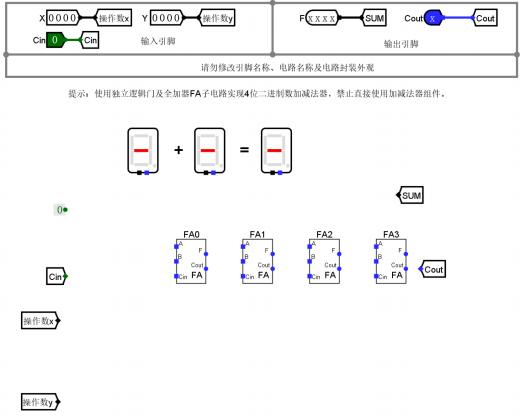
1） 全加器实验。在 Logisim 中添加子电路：全加器 FA 。根据全加器逻辑电路图，在 Logisim 工作区 中添加逻辑门、连线和标识符。然后将或门输入端口数改为 3；将输入引脚、逻辑门的输入端和输出端、 输出引脚等通过连接线相连。选中输入、输出引脚，在属性表中添加引脚标识符。选中逻辑门，在属性表 中添加门标识符。点击快捷工具栏中文本工具，在电路空白处添加描述文字；如图 2.5 所示。



|  |
| --- |
|  |

图 2.5 全加器电路图 验证电路功能，查看子电路外观及引脚布局。

2） 4 位串行进位加法器实验。在 Logisim 主电路 main 工作区中按图 2.6 所示进行组件布局，其中包 含输入输出引脚、隧道、分线器、全加器子电路、常量和 16 进制数字显示等组件。修改组件属性，连接 端口，实现 4 位串行进位加法器电路，并验证功能，记录测试数据。



|  |
| --- |
|  |

图 2.6 4 位串行加减法器组件布局图

3） 4 位二进制数补码减法运算实验。根据 Cin 输入值区分加减法运算，当 Cin=0 时，执行补码加法

运算 F=X+Y；当 Cin=1 时，执行补码减法运算 F=X-Y 。修改上述电路，添加必要的组件，例如位扩 展器、异或门等，进行功能验证，并分析结果数据。保存电路设计文件 lab2.3.circ。

**4、汉明码校验电路（验收）**

数据校验大多采用“冗余校验”的思想，即除原数据信息外，还增加若干位附加的编码，这些新增编码 称为校验位。图 2.7 给出了一般情况下的处理过程。

出错信号



数据输出

M

M ´

数据输入 M

|  |
| --- |
| 比较 |

P ´

|  |
| --- |
| F |

P

P ´´

|  |
| --- |
| F |

|  |
| --- |
| 纠错器 |

|  |
| --- |
| 存储器 或  传输线路 |

图 2.7 数据校验过程示意图

当数据 M 被存入存储器或从源部件开始传输时，对 M 进行某种运算（用函数 F 来表示），以产生相应 的代码 P=F(M)，这里 P 就是校验位。这样原数据信息 M 和相应的校验位 P 一起被存储或传送。当数据被 读出或传送到目标部件时，和数据信息一起被存储或传送的校验位也被得到，用于检错和纠错。假定读出 后的数据为 M´, 通过同样的运算 F 对 M´也得到一个新的校验位 P´=F(M´)，假定原来被存储的校验位 P 取 出后其值为 P´´, 将校验位 P´´与新生成的校验位 P´进行比较运算，生成一个故障字，根据故障字可以确定 是否发生了差错。

最简单的数据检错方法是奇偶校验，通过判断数据 M 中 1 的个数是否发生了奇偶性变化来进行检错。 若发生奇偶变化，则故障位 S= P´⊕P´´=1。

汉明码（Hamming Code，也译为海明码）的主要思想是，将数据按某种规律分成若干组，对每组进行 相应的奇偶检测，以提供多位校验信息，得到相应的故障字，根据故障字对发生的错误进行定位，并将其 纠正。汉明校验码实质上就是一种多重奇偶校验码。

对于只能对单个位出错的情况进行定位和纠错的单纠错码（SEC），进行汉明校验的主要思想如下：将 需要进行检/纠错的数据分成 *i* 组，每组对应 1 位校验位，共有 *i* 位校验位，因此，故障字为 *i* 位。若故障 字为 0，表示无错；否则故障字的数值就是出错位在码字中的位置编号。除去 0 的情况，*i* 位故障字的编码 个数为 2*i*-1，因此构造的码字最多有 2*i*-1 位，例如，当 *i*=3 时，码字可以有 7 位，其中 3 位为校验位，4 位 为数据位。为了方便判断码字中出错的是校验位还是数据位，可将校验位的位置编号设为 2 的幂次，即校 验位排在第 1（001）、2（010）、4（100）、 … 的位置上，其余位置上为数据位。这样，当故障字中只有一

位为 1 时，说明是校验位出错，否则就是数据位出错。例如，当 *i*=3 时，假设校验码为 P3P2P1，数据信息 为 M4M3M2M1 ，则码字排列为 P1P2M1P3M2M3M4。通常把上述由数据位和校验位构成的码字称为汉明码。 图 2.8 给出了7 位汉明码的故障字和出错情况的对应关系。



图 2.9 8 位汉明码的故障字和出错情况对应关系

如图 2.9 所示，第 1 组的故障位 S1 由校验位 P1 和数据位 M1 、M2 、M4 生成，第 2 组的故障位 S2 由校 验位 P2 和数据位 M1 、M3 、M4 生成、第 3 组的故障位 S3 由校验位 P3 和数据位 M2 、M3 、M4 生成。假设在 终部件得到的数据位 M´为 M4M3M2M1 ，校验位 P´´为 P3P2P1，每组采用偶校验，则根据 M´得到 P´的每一 位如下：

P1 ´ = M1 ⊕M2 ⊕M4 P2 ´ = M1 ⊕M3 ⊕M4 P3 ´ = M2 ⊕M3 ⊕M4

因为故障字 S=P´⊕P´´,因此，根据 P´和 P´´得到故障字的每一位如下：

S1 = M1 ⊕M2 ⊕M4 ⊕P1 S2 = M1 ⊕M3 ⊕M4 ⊕P2 S3 = M2 ⊕M3 ⊕M4 ⊕P3

因此，在终部件的汉明码检/纠错电路只要根据在终部件得到的数据位 M4M3M2M1 和校验位 P3P2P1 形 成的码字，按图 2.9 所示的方式划分成 3 组，每组按照上述偶校验方式，得到每一组的故障位 S*i*，由故障 位构成的故障字 S3S2S1 的值就能确定码字中哪一位发生了错误。

图 2.10 给出了7 位汉明码检/纠错电路原理图。

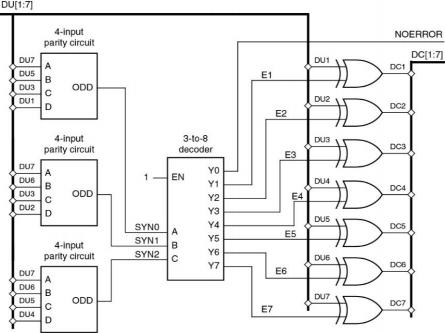
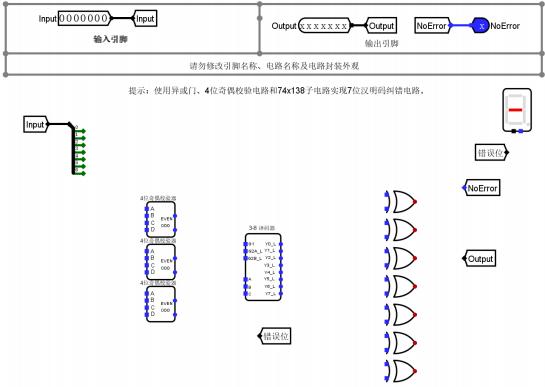


图 2.10 7 位汉明码检/纠错电路原理图

在 Logisim 的工作区中利用译码器、奇偶校验子电路、隧道和分线器等组件实现 7 位汉明码纠错功

能，并显示发生错误位置和状态，参考组件布局图如 2.11 所示，输入测试数据并进行验证说明，保存设计 文件为 lab2.4.circ。



|  |
| --- |
|  |

图 2.11 7 位汉明码纠错电路组件布局图

**s、桶形移位器**

桶形移位器采用组合逻辑的方式来实现移位功能，能在一个时钟周期内完成多位移动，具有很高 的效率，常被用在 ALU 中来实现移位运算。它具有 n 位数据输入和 n 位数据输出，以及指定移动方 向、移动类型（算术/逻辑/循环）和移动位数等。

8 位桶形移位器的输入输出引脚图，如图 2-12 所示。其中输入数据 din 和输出数据 dout 均为 8

位，移位位数 shamt 为 3 位。选择端 L/R 表示左移和右移，选择端 A/L 表示算术、逻辑和循环移位。 （L/R,A/L）赋值为 00 时表示逻辑右移，赋值为 01 表示算术右移，赋值为 10 时表示逻辑左移，赋值 为 11 时表示循环左移。循环右移可以使用循环左移通过改变移位位数来实现。

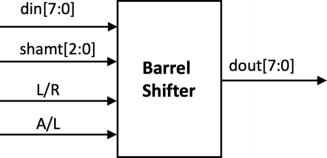
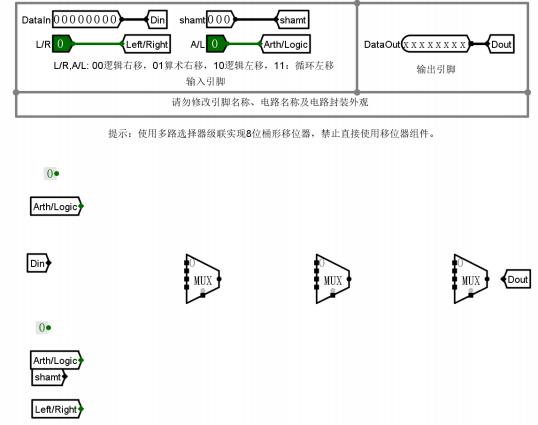


图 2-12 8 位桶形移位器逻辑符号图

桶形移位器的功能有 2 位 L/R 和 A/L，因而可以使用四路选择来实现数据的输入。8 位输入数据 需要 3 位移位长度标识。需要使用 3 级四路选择器级联实现移动位数，第一级可移动 1 位，第二级可 移动 2 位，第三极可移动 4 位。在边界右移时，需考虑移入位是 0 还是输入数据的最高位 din[7]（算 术右移），在左移时，需考虑移入位是 0 或者是输入数据的高位（循环左移）。该移位电路为组合逻辑 电路，在输入改变时无需时钟信号就可以直接得到移位结果。

在 Logisim 的工作区中放置多路选择器、分线器、常量 0 等组件实现 8 位桶形移位器，参考布局图如 2.14 所示，输入测试数据并进行验证说明，保存设计文件 lab2.5.circ。



|  |
| --- |
|  |

图 2.14 8 位桶形移位器组件布局图

**四、思考题**

1. 修改实验中的加法器电路，生成进位标志 CF、溢出标志 OF、符号标志 SF 和结果为零标志位 ZF。

2. 在执行比较指令时，通常使用减法运算后，判断标志位的方式来实现，试通过上述加法器实验举例 说明判别的方法。

3. 如何使用 8 位桶形移位器扩展到32 位桶形移位器。