|  |  |  |  |
| --- | --- | --- | --- |
| 姓名： | 王哲霖 | 学号： | 2307110346 |
| 学院： | 计算机与电子信息学院 | 专业： | 计算机 |
| 班级： | 233班 | 时间： | 10.22 |
|  |  | 指导教师： | 何华光 |

# 实验名称： 加法器

一、实验目的

了解常用加法器的设计方法。

掌握四位超前进位并行加法器 74LS283。

掌握算术逻辑运算器（ALU）74LS181。

二、实验原理

基本逻辑电路案例，加法器，全加器，并、串行全加器，基本逻辑运算单元

三、实验设备及器件

1、与或非门电路器若干个；

2、数字信号源若干个；

3、仿真软件：Dream Logic 2019。

四、实验内容及过程

1、四位超前进位并行加法器 74LS283

（1）任务描述：

串行进位加法器须将低位全加器产生的进位信号逐位向高一位全加器上传递。因此，串行进位加法器 求和的最高位的输出结果必须等到各位进位信号逐位传递后才能形成，工作速度很慢。为了提高工作速度， 可采用四位超前进位并行加法器

（2）实验步骤

1. 打开项目下的原理图文件 74LS283.dlsche。该原理图文件中绘制了 74LS283 的内部原理图。其输 入信号包括两个加数 A1~A4、B1~B4,以及进位输入 C0。输出信号包括和 S1~S4，最高进位输出 C4。

2. 启动仿真。根据 74LS283 的功能表 3-1，验证其加法运算功能。并将表格补充完整。注意表格中 使用的是十六进制数。

3. 在 74LS283 中使用一个红色矩形框将所有用于计算 S1的逻辑门包含在其中。并写出通过这些逻辑 门计算 S1的表达式，然后证明这个表达式与本实验 2.2 节中推导的 S1的表达式（将表达式 2 带入 表达式 3 可得到）等效。绘制红色矩形框的方法是，选择“绘制”菜单中的“矩形”，然后在原理 图中将所有用于计算 S1的逻辑门包含在其中，绘制完毕后，选中矩形框，在右侧的“属性”视图 中，将矩形框的“填充”属性修改为“否”，将轮廓的“颜色”修改为红色。

4. 请读者再尝试证明在 74LS283 中使用逻辑门计算 S2的表达式与本实验 2.2 节中推导的 S2的表达式是等效的。

（3）任务1小结

（4）电路图、步骤截图、真值表及实验数据表等

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入信号 | | | 输出信号 | |
| 四位加数A | 四位加数B | 进位输入CO | 四位和S | 进位输出C4 |
| 3 | 5 | 0 | 8 | 0 |
| 3 | 5 | 1 | 9 | 0 |
| 7 | 9 | 0 | 0 | 1 |
| 7 | 9 | 1 | 1 | 1 |
| F | F | 0 | E | 1 |
| F | F | 1 | F | 1 |

2、算术逻辑运算单元 74LS181

（1）任务描述：

算术逻辑运算单元（ALU）74LS181 是一种功能较强的组合逻辑电路，它能进行多种算术运算和逻辑运算。它的基本逻辑结构是超前进位加法器。

1. 实验步骤

1. 打开项目下的原理图文件 74LS181.dlsche。该原理图文件中绘制了 74LS181 的内部原理图（该原 理图有问题，可用芯片 74LS181 来替代进行）。

2. 启动仿真，根据表 3-2 对 74LS181 进行功能验证，由于 74LS181 的运算功能很多，可以选择一部 分功能进行验证，请读者自行设计一个功能验证记录表，至少要验证表 3-2 中有阴影的功能项，并记录所有的输入输出信号。

（3）任务1小结

（4）电路图、步骤截图、真值表及实验数据表等

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **输入端** | | **输出端（16进制）** | | |
|  | **方式** | **M=1逻辑运算** | **M=0算术运算** | |
| **AB(16进制)** | **S3S2S1S0** | **逻辑运算** | **CN=1(无进位)** | **CN=0(有进位)** |
| **00** | **0 0 0 0** | **f** | **0** | **1** |
| **ff** | **0 0 0 0** | **0** | **f** | **0** |
| **99** | **0 0 0 0** | **6** | **9** | **a** |
| **10** | **0 1 1 0** | **1** | **0** | **1** |
| **11** | **0 1 1 0** | **0** | **f** | **0** |
| **00** | **0 1 1 0** | **0** | **f** | **0** |
| **00** | **1 0 0 1** | **f** | **0** | **1** |
| **51** | **1 0 0 1** | **6** | **6** | **7** |
| **fe** | **1 0 0 1** | **e** | **d** | **e** |
| **10** | **1 0 1 1** | **0** |  |  |
| **11** | **1 0 1 1** | **1** |  |  |
| **00** | **1 0 1 1** | **0** |  |  |
| **00** | **1 1 1 0** | **0** |  |  |
| **10** | **1 1 1 0** | **1** |  |  |
| **01** | **1 1 1 0** | **1** |  |  |
| **ee** | **1 1 1 1** | **d** |  |  |
| **10** | **1 1 1 1** | **0** |  |  |
| **25** | **1 1 1 1** | **1** |  |  |

3、加法器与译码器综合设计

（1）任务描述：

本任务要求读者在原理图文件 top.dlsche 中使用一片 74LS181 设计一个简单的运算单元（ALU），使 该 ALU 在 3 位输入信号 X0~X2控制下可以完成 8 种不同的运算，并通过仿真进行功能验证。具体要求如下：

通过 6 个输入信号 S3、S2、S1、S0、M、CN 可以控制 74LS181 完成 48 种不同的运算。假设一个使用 74LS181 做运算器的 4 位处理器，它的指令编码中只有 3 位可用于对运算方式进行编码（支持 23=8 种不同 运算）。请读者设计一个译码电路，该译码电路可以接收 3 位运算方式编码，然后输出 6 个控制信号，使 用这 6 个控制信号控制 74LS181 完成 8 种不同的运算。

假设这 3 位编码分别为 X2、X1、X0，译码输出最小项的组合控制信号分别为 S3、S2、S1、S0、M、CN。X2X1X0的值可以为 0~7，分别对应表 3-3 中的 8 种不同的运算。

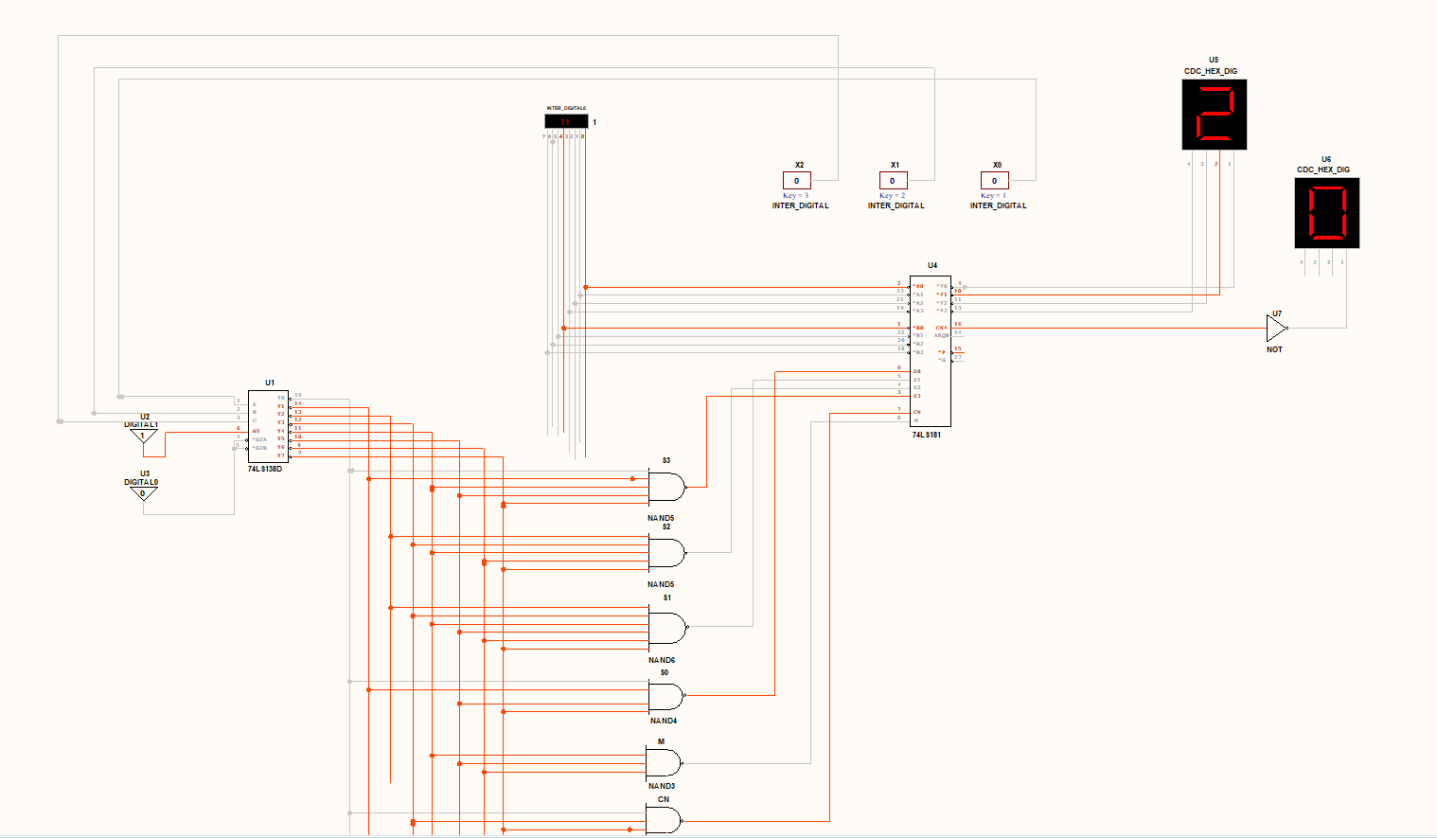
（2）实验步骤

读者应首先根据表 3-3 设计一个真值表，该真值表的输入信号是 X0~X2，输出信号是 74LS181 的 6 个控制信号，然后根据真值表设计译码电路。译码电路可以使用一片 74LS138 将输入信号译码后，再单 独设计一个编码电路，通过基本的逻辑门将 8 个信号编码为 6 个控制信号；也可以使用基本的逻辑门将 3 个输入信号直接译码为 6 个控制信号。

（3）任务1小结

（4）电路图、步骤截图、真值表及实验数据表等

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **X2** | **X1** | **X0** | **S3** | **S2** | **S1** | **S0** | **M** | **CN** |
| **0** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **1** |
| **0** | **0** | **1** | **1** | **0** | **0** | **1** | **0** | **0** |
| **0** | **1** | **0** | **0** | **1** | **1** | **0** | **0** | **0** |
| **0** | **1** | **1** | **0** | **1** | **1** | **0** | **0** | **1** |
| **1** | **0** | **0** | **1** | **1** | **1** | **0** | **1** | **X** |
| **1** | **0** | **1** | **1** | **0** | **1** | **1** | **1** | **X** |
| **1** | **1** | **0** | **0** | **1** | **1** | **0** | **1** | **X** |
| **1** | **1** | **1** | **1** | **1** | **1** | **1** | **0** | **1** |



五、实验收获与心得

1、本次实验了解了加法器的基本使用方法，了解了基本逻辑运算单元