

## 实验 8 全加器与移位器

“If an a-machine prints two kinds of symbols, of which the first kind (called figures) consists entirely of 0 and 1 (the others being called symbols of the second kind), then the machine will be called a computing machine.”

（“如果一个 a 机器可以输出两种类型的符号，其中第一种（称为数字）仅由 0 和 1 组成（第二种称为符号），那么这个机器可以被称为计算机器。”）

——《On computable numbers, with an application to the Entscheidungsproblem》，A. M. Turing

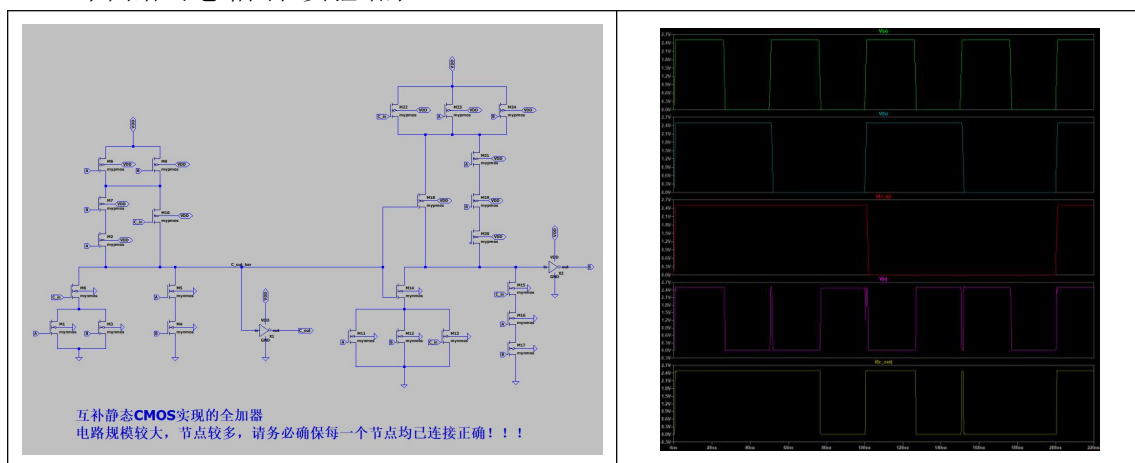
在我们设计了诸如与或非门等基本组合逻辑单元以及锁存器和寄存器等时序逻辑单元后，我们可以将我们的目标，放到构建更大的模块上。ALU 是 CPU 中的基本单元，通过本节实验，我们将使用晶体管搭建一个全加器。

### 8.1. 全加器

全加器的原理我们再次不过多介绍。我们可以直接将标准的表达式变为互补的 CMOS 电路。但是，我们可以进行某些逻辑变换，从而减少晶体管的数目。例如，我们在“和”与“进位产生”的子电路之间共享某些逻辑非常有利，这里我们写出经过变化后的全加器的表达式（共享了反相 C\_out）：

$$\begin{aligned}C_{out} &= AB + AC_{in} + BC_{in} \\S &= ABC_{in} + \overline{C_{out}}(A + B + C_{in})\end{aligned}$$

下面给出电路图和实验结果：

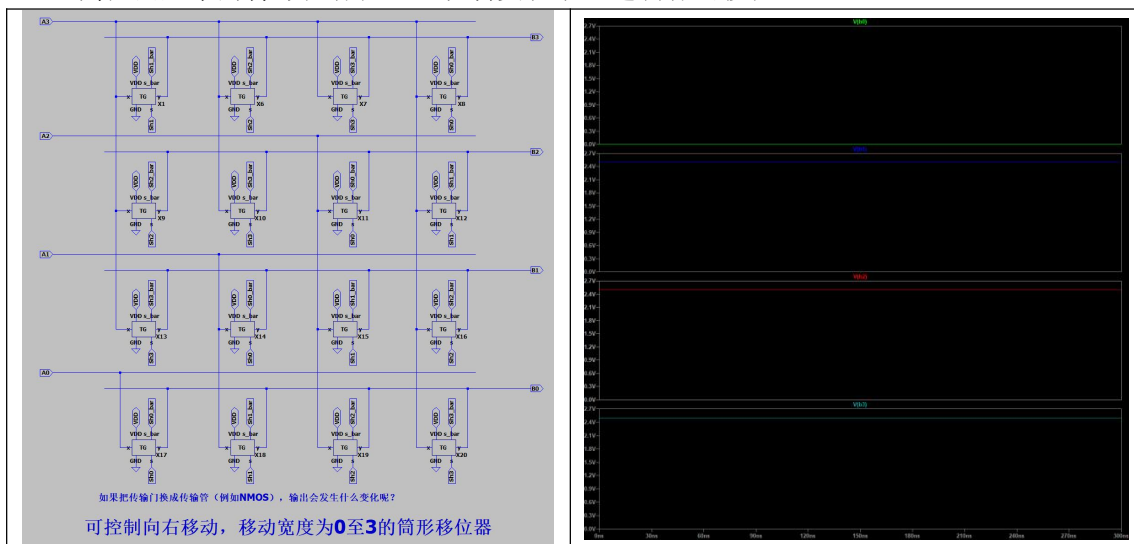


### 8.2. 移位器

#### 8.2.1. 筒形移位器

筒形移位器由一个晶体管（传输门）阵列组成。其行数等于数据的字长，而列数等

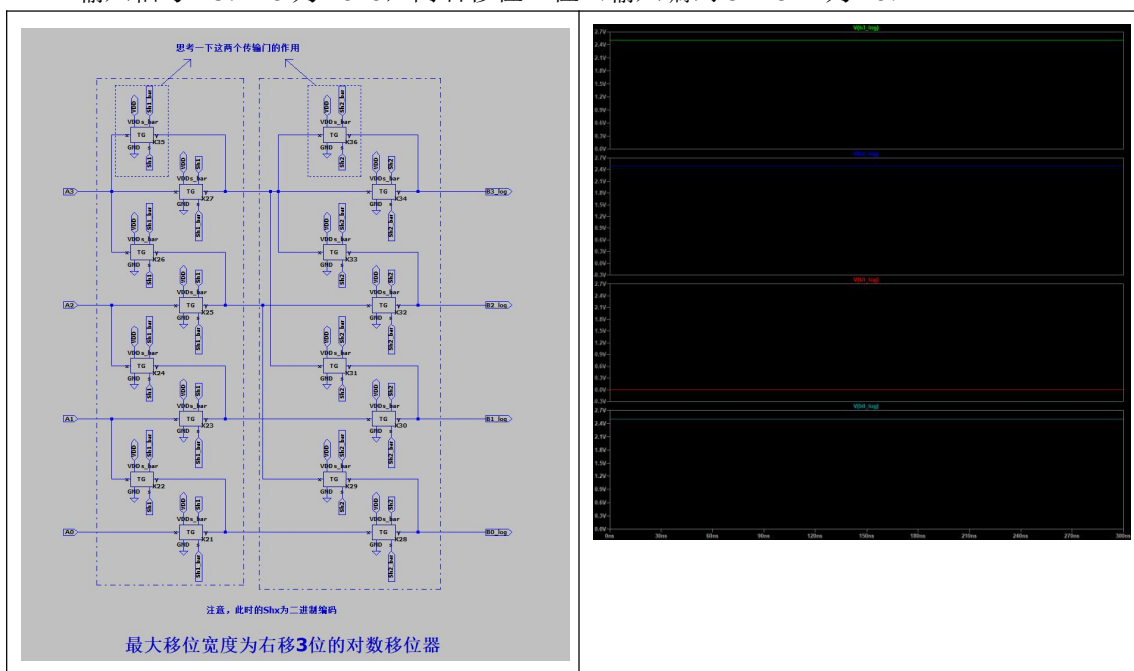
于最大的以为宽度。我们这里给出了一个可控制向右移动，移动宽度位 0 至 3 的桶形移位器，数据的字长为 4。其中， $A_x$  表示输入数据， $B_x$  表示输出数据， $Shx$  表示向右移动  $x$  位，使用独热码编码，例如当我们希望右移 2 位时， $Sh3: Sh0$  等于 0100。尤其可见，我们需要配合使用一个译码器电路，将二进制编码转化为独热码（编码器在数字电路课程中讲解过，此处不再赘述）。下面为电路图和实验结果， $A3: A0$  为 1010（1 表示高电平  $VDD$ ，0 表示低电平  $GND$ ），可以看到输出的  $B3: B0$  为 1110，这说明该移位器最高位由原来的符号位补位，也说明实现了二进制补码移位。



### 8.2.2. 对数移位器

对数移位器采用分级的方法。总的移位值被分解成为几个 2 的指数值，因此对数移位器的输入控制字已经是二进制编码（编码信息体现在电路的连接方式中），所以无需单独的译码器。

下面我们展示了一个 2 位控制字（可以提供 0 至 3 位移位宽度）的右移对数移位器，输入信号  $A3: A0$  为 1010，向右移位 2 位（输入编码  $Sh2:Sh1$  为 10）。



一般来说，筒形移位器适用于较小的移位值，对于较大的移位值，对数移位器在面

积和速度方面更有优势。而且对数移位器很容易参数化，便于自动化生成。

### 8.3. 动手实验内容

- 1 自己实现筒形移位器的左移功能
- 2 实现一个 8 位的最大移位宽度位右移 7 位的对数移位器
- 3 设计右移自动补位为 1 或者 0 的对数移位器