

# lab02 report

---

郑航 520021911347

## lab02 report

- 1 实验目的
- 2 实验原理
  - 2.1 一位全加器原理分析
  - 2.2 四位全加器原理分析
- 3 实验过程
  - 3.1 一位全加器
  - 3.2 四位全加器
- 4 实验结果
- 5 工程实现
- 6 反思总结

## 1 实验目的

---

- 掌握 Xilinx 逻辑设计工具 Vivado的基本操作
- 掌握使用 Verilog HDL进行简单的逻辑设计
- 掌握功能仿真
- 约束文件的使用和直接写法
- 生成 Bitstream 文件
- 上板验证

其中下载验证部分暂时不做

## 2 实验原理

---

### 2.1 一位全加器原理分析

- 功能：对两个输入运算位以及一个输入的进位位进行相加，并输出相加的该位结果和进位结果
- 输入：a, b两个运算位和ci一个进位位
- 输出：s一个结果位, c0一个进位位

一位全加器的真值表如下：

$a$	$b$	$c_i$	$s$	$c_o$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

只需要通过逻辑运算将输入输出对应起来即可，其中：

$$s = a \oplus b \oplus c_i$$

$$c_o = (a \wedge b) \vee (a \wedge c_i) \vee (b \wedge c_i)$$

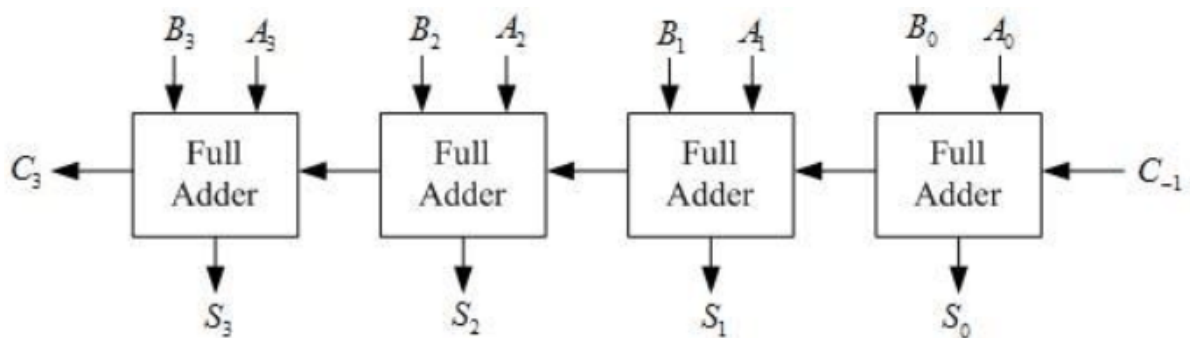
## 2.2 四位全加器原理分析

- 功能：与一位全加器类似，只是其输入和输出的运算位由1位变为4位，进位位保持不变
- 输入：a，b两个4位的运算数和 $c_i$ 一个进位位
- 输出：s一个四位的结果数和 $c_o$ 一个进位位

其实现思路是将四个2.1中实现的一位全加器进行串联，其中：

- 前一个一位全加器的  $c_o$  端接入后一个一位全加器的  $c_i$  端，第一个一位全加器的  $c_i$  端为整个四位全加器的  $c_i$  端，最后一个一位全加器的  $c_o$  端为整个四位全加器的 $c_o$ 进位输出
- 四位的a和b分别拆为四个部分，按顺序放于串联的四个一位全加器中，结果进行合并即可

一位加法器连接示意图：



## 3 实验过程

### 3.1 一位全加器

根据2.1中所述的原理，只需要实现那两个逻辑表达式即可，我们使用四个wire型的变量s1，c1，c2和c3作为中间变量存储运算结果，并使用Verilog内置的逻辑运算单元实现我们的目标运算

完整代码如下：

```


```

```

1  module adder_1bit(
2      input a,
3      input b,
4      input ci,
5      output s,
6      output co
7  );
8      wire s1, c1, c2, c3;
9      and (c1, a, b),
10         (c2, b, ci),
11         (c3, a, ci);
12      xor (s1, a, b),
13         (s, s1, ci);
14      or (co, c1, c2, c3);
15
16  endmodule

```

## 3.2 四位全加器

根据2.2中所述的原理，四位全加器只是在一位全加器的基础上进行连接即可，并不需要实现任何运算，其连接图如2.2中所示，连接过程中，我们需要设置三位的ct作为中间变量，存储进位结果

完整代码如下：

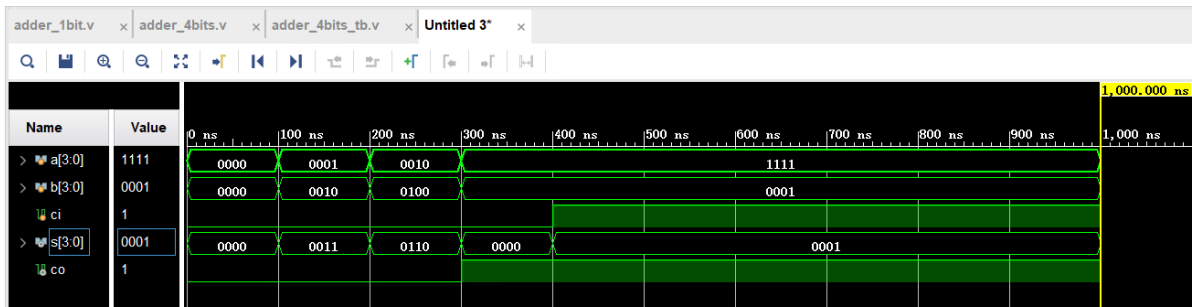
```

1  module adder_4bits(
2      input [3:0] a,
3      input [3:0] b,
4      input ci,
5      output [3:0] s,
6      output co
7  );
8
9      wire [2 : 0] ct;
10
11      adder_1bit a1(.a(a[0]), .b(b[0]), .ci(ci), .s(s[0]), .co(ct[0])),
12                  a2(.a(a[1]), .b(b[1]), .ci(ct[0]), .s(s[1]), .co(ct[1])),
13                  a3(.a(a[2]), .b(b[2]), .ci(ct[1]), .s(s[2]), .co(ct[2])),
14                  a4(.a(a[3]), .b(b[3]), .ci(ct[2]), .s(s[3]), .co(co));
15
16  endmodule

```

## 4 实验结果

我们本实验只需要对四位全加器进行仿真验证即可，仿真代码实验指导书中已给出，在此不加展示仿真波形如下：



可以看到，我们的四位全加器运算结果完全正确，实验成功

## 5 工程实现

Top 和xdc约束文件都已实现，详见工程文件目录中，在此也不加展示

## 6 反思总结

本实验实现了一位全加器和四位全加器这两个基础部件的设计与仿真。利用上个实验lab1所学习到的Verilog语法，结合一些Verilog自带的逻辑运算单元可以较简单的完成本模块的功能。在本实验中，我深刻认识到了先进行细致的原理分析然后再开始动手实现的良好实验习惯的重要性，对真值表和连接图等的分析使得我对整个实验的实现思路非常理解，完成起来也比较顺手，也能有所收获

实验中，通过先实现一位全加器，再进而实现四位全加器的思路非常重要，在后续的很多实验中也延续了这种设计思路，先将功能拆分并分为多个小模块，先进行实现，这样对大模块的实现就转化为了多个小模块的连接，这对于完成较复杂的模块来说是非常重要的

通过本次实验，我对Verilog的语法和代码逻辑更加熟悉，这对后续进行处理器模块的实现奠定了良好的基础