

# 深圳大学实验报告

课程名称： 硬件描述语言及数字系统设计

实验项目名称： 实验一 加法器设计

学院： 医学院

专业： 生物医学工程

指导教师： 但果、董磊

报告人： 陈焕鑫 学号： 2016222042 班级： 生工 2 班

实验时间： 2018.9.19

实验报告提交时间： 2018.10.8

教务部制

## 一、实验平台：

安装了 ISE 软件的 PC 计算机  
硬件描述语言及数字系统设计实验平台  
JTAG 下载线

## 二、实验目的：

使用 ISE 软件建立一个完整的 EDA 工程；  
撰写一个简单的 VHDL 程序，并且能够通过 ISE 软件修改语法错误  
编写测试激励（仿真代码），并且能够对 VHDL 程序进行仿真调试  
编写引脚约束文件  
用 ISE 软件生成 .bit 文件，并且学会使用 ISE iMPACT 将 .bit 下载到 FPGA

## 三、实验内容：

用 VHDL 代码描述一个带输入进位和输出仅为 2 位加法器  
通过 ISE 软件对设计好的加法器进行编译  
编写测试激励即 Test Bench 文件，对编译好的 2 位加法器进行仿真调试  
编写引脚约束文件，用 ISE 软件生成 bit 文件，下载到硬件描述语言及数字系统设计平台进行板级验证

## 四、实验原理：

本此实验实验的加法器的组合方法是逐级进位。该方法的优势是需要较少的硬件资源。逐级进位是将两个全加器串联起来的。全加器的输入输出框图如图 4-1 所示。

全加器真值表

a	b	Cin	s	Cout
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

全加器表达式

$$s = a \oplus b \oplus \text{Cin}$$
$$\text{Cout} = ab + b\text{Cin} + a\text{Cin}$$

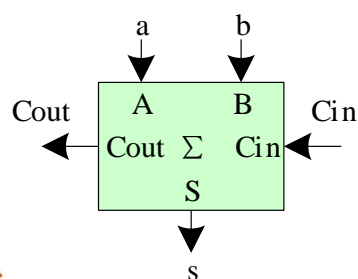


图 4-1

画出全加器的真值表，根据真值表写出全加器的逻辑表达式，即得到输入与输出的关系表达式。

将两个全加器之间通过进位端口按照如图 4-2 相连接即可得到一个计算 2 位二进制的逐级进位加法器。

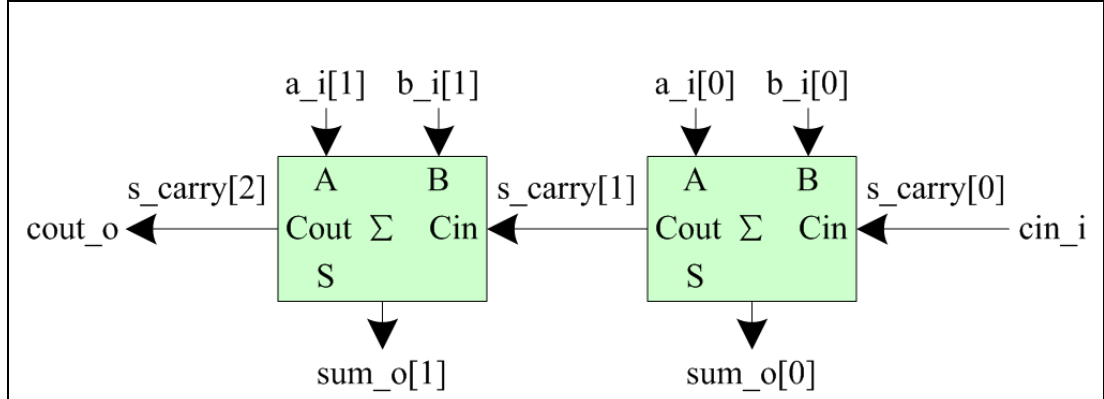


图 4-2

### 五、实验方法步骤及 VHDL 代码：

根据加法器的原理进行代码设计，将两个加法器串联，其中一个加法器的输出进位作为第二个加法器的输入进位。

具体的 VHDL 代码如下所示：

```
--模块名称：adder
--摘要提示：
--当前版本：1.0.0
--模块作者：
--完成日期：2018 年 9 月 19 日
--内容提要：
--需要注意：

--取代版本：
--模块作者：
--完成日期：
--修改内容：
--修改文件：

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

-----ENTITY DECLARATION-----
entity adder is
  port(
    a_i      : in  std_logic_vector(1 downto 0);
    b_i      : in  std_logic_vector(1 downto 0);
    cin_i    : in  std_logic;
```

```

sum_o  : out std_logic_vector(1 downto 0);
cout_o : out std_logic
);
end adder;

-----ARCHITECTURE STRUCTURAL-----

architecture rtl of adder is

    signal s_carry : std_logic_vector(2 downto 0);
begin

    s_carry(0) <= cin_i;

    s_carry(1) <= (a_i(0) and b_i(0)) or (a_i(0) and s_carry(0)) or (b_i(0) and
s_carry(0));
    sum_o(0) <= a_i(0) xor b_i(0) xor s_carry(0);

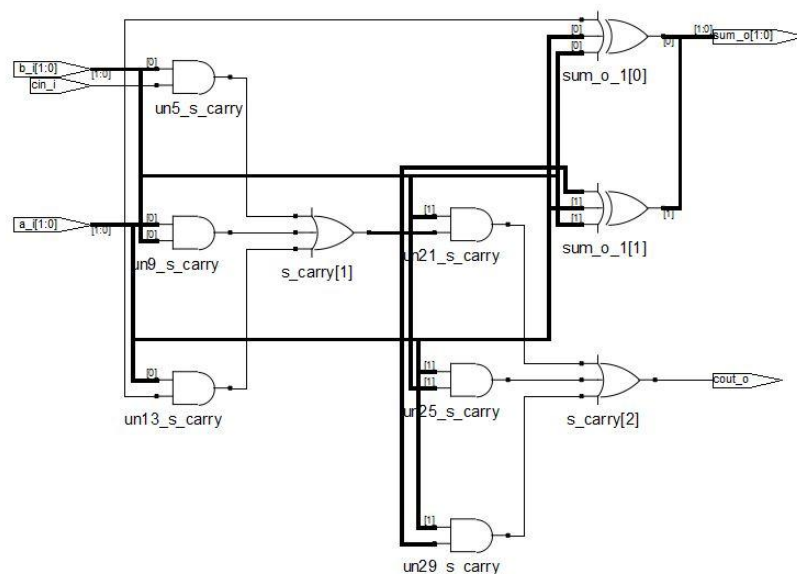
    s_carry(2) <= (a_i(1) and b_i(1)) or (a_i(1) and s_carry(1)) or (b_i(1) and
s_carry(1));
    sum_o(1) <= a_i(1) xor b_i(1) xor s_carry(1);

    cout_o <= s_carry(2);
end rtl;

```

## 六、综合、仿真结果及分析:

### 1、RTL 级电路综合:



## 2、电路仿真

### 测试激励 Test Bench 代码

```
-----  
--模块名称: adder_tb  
--摘要提示: 测试激励  
--当前版本: 1.0.0  
--模块作者:  
--完成日期: 2018 年 9 月 19 日  
--内容提要:  
--需要注意:  
-----  
  
--取代版本:  
--模块作者:  
--完成日期:  
--修改内容:  
--修改文件:  
-----  
  
LIBRARY ieee;  
USE ieee.std_logic_1164.ALL;  
  
ENTITY adder_tb IS  
END adder_tb;  
  
ARCHITECTURE behavior OF adder_tb IS  
  
    -- Component Declaration for the Unit Under Test (UUT)  
  
    COMPONENT adder  
    PORT (  
        a_i : IN  std_logic_vector(1 downto 0);  
        b_i : IN  std_logic_vector(1 downto 0);  
        cin_i : IN  std_logic;  
        sum_o : OUT std_logic_vector(1 downto 0);  
        cout_o : OUT std_logic  
    );  
    END COMPONENT;  
  
    --Inputs  
    signal a_i : std_logic_vector(1 downto 0) := (others => '0');  
    signal b_i : std_logic_vector(1 downto 0) := (others => '0');  
    signal cin_i : std_logic := '0';
```

```

--Outputs
signal sum_o : std_logic_vector(1 downto 0);
signal cout_o : std_logic;
-- No clocks detected in port list. Replace <clock> below with
-- appropriate port name

```

**BEGIN**

```

-- Instantiate the Unit Under Test (UUT)

```

```

uut: adder PORT MAP (
    a_i => a_i,
    b_i => b_i,
    cin_i => cin_i,
    sum_o => sum_o,
    cout_o => cout_o
);

```

```

-- Stimulus process

```

```

stim_proc: process

```

```

begin

```

```

    -- hold reset state for 100 ns.

```

```

    wait for 100 ns;

```

```

    -- insert stimulus here

```

```

    a_i <= "01";

```

```

    b_i <= "10";

```

```

    cin_i <= '1';    --sum=00;cout_o=1

```

```

    wait for 100 ns;

```

```

    a_i <= "10";

```

```

    b_i <= "10";

```

```

    cin_i <= '0';    --sum=00;cout_o=1

```

```

    wait for 100ns;

```

```

    a_i <= "00";

```

```

    b_i <= "10";

```

```

    cin_i <= '1';    --sum=11;cout_o=0

```

```

    wait for 100ns;

```

```

    a_i <= "11";

```

```

    b_i <= "11";

```

```
cin_i <= '1';  --sum=11;cout_o=1

wait;

end process;

END;
```

仿真结果如下图所示：

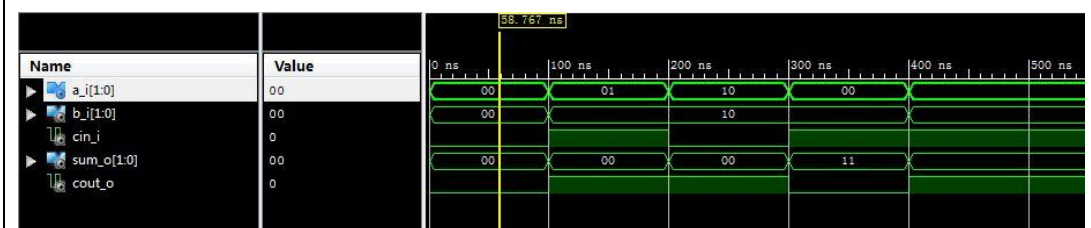


图 6-1

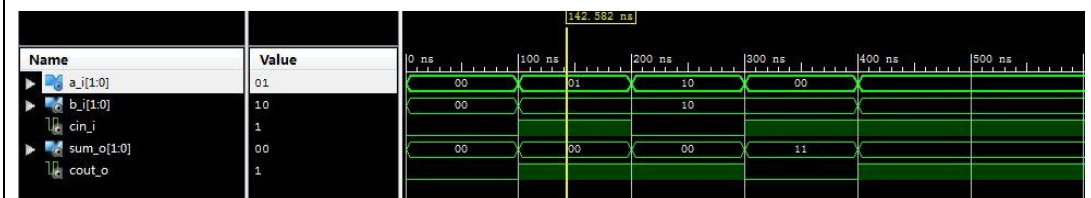


图 6-2

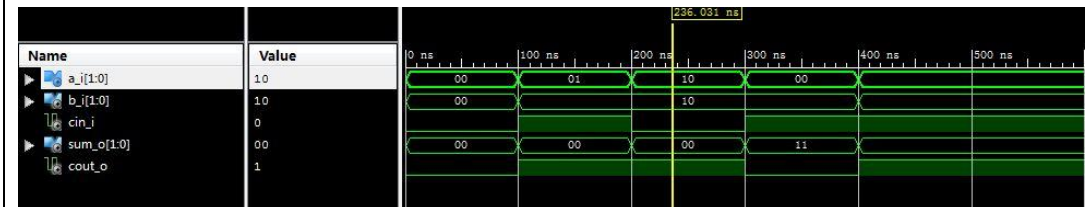


图 6-3

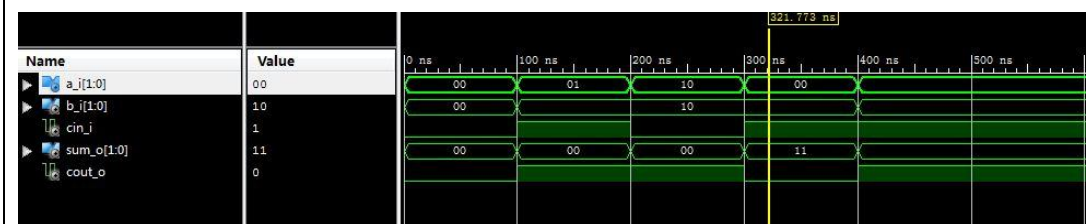


图 6-4

在图 6-1 中可以看出，当 a\_i 为“00”，b\_i 为“00”，cin\_i 为‘0’时，输出 sum\_o 为“00”，进位 cout\_o 为‘0’。即进位为 0， $a = 0, b = 0, \text{sum} = 0 + 0 + 0 = 0$ 。

在图 6-2 中可以看出，当 a\_i 为“01”，b\_i 为“10”，cin\_i 为‘1’时，输出 sum\_o 为“00”，进位 cout\_o 为‘1’。即进位为 1， $a = 1, b = 2, \text{sum} = 1 + 1 + 2 = 4$ 。

在图 6-3 中可以看出，当 a\_i 为“10”，b\_i 为“10”，cin\_i 为‘0’时，输出 sum\_o 为“00”，进位 cout\_o 为‘1’。即进位为 0， $a = 2, b = 2, \text{sum} = 0 + 2 + 2 = 4$ 。

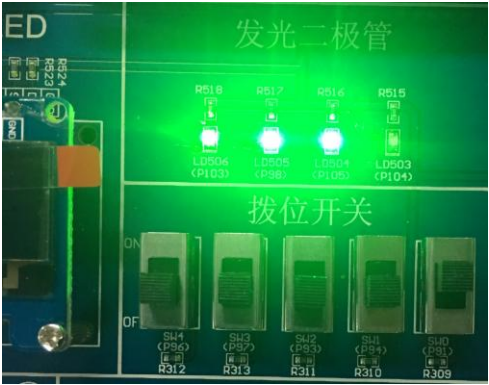
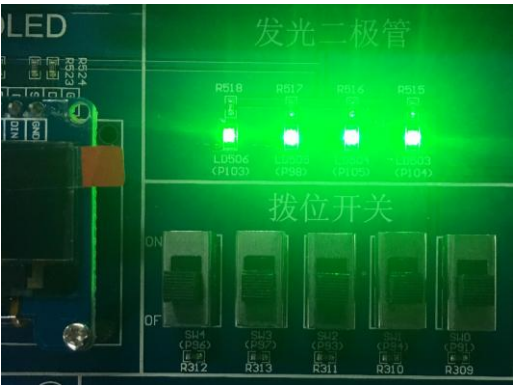
在图 6-4 中可以看出，当 a\_i 为“00”，b\_i 为“10”，cin\_i 为‘1’时，输出 sum\_o 为“11”，进位 cout\_o 为‘0’。即进位为 1， $a = 0, b = 2, \text{sum} = 1 + 0 + 2 = 3$ 。

## 七、FPGA 板级验证及结果分析

1、在工程中加入引脚约束文件，代码如下

```
##-----  
##模块名称: adder.ucf  
##摘要提示:  
##当前版本: 1.0.0  
##模块作者:  
##完成日期: 20xx 年 xx 月 xx 日  
##内容提要:  
##需要注意:  
##-----  
  
##取代版本:  
##模块作者:  
##完成日期:  
##修改内容:  
##修改文件:  
##-----  
  
Net "cin_i" LOC=P96;  
  
Net "a_i<1>" LOC=P97;  
Net "a_i<0>" LOC=P93;  
  
Net "b_i<1>" LOC=P94;  
Net "b_i<0>" LOC=P91;  
  
Net "cout_o" LOC=P105;  
  
Net "sum_o<1>" LOC=P98;  
Net "sum_o<0>" LOC=P103;
```

综合之后生成 bit 文件烧进试验箱中观察到如下图所示的现象:





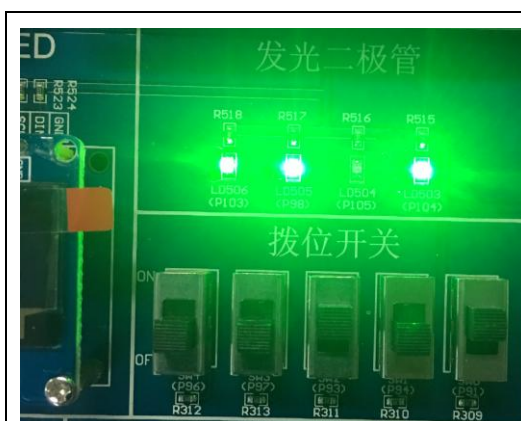


图 7-3

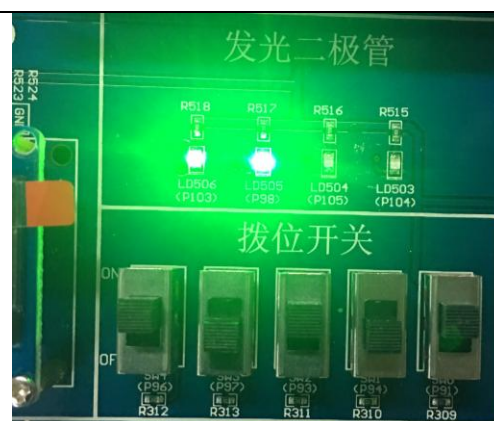
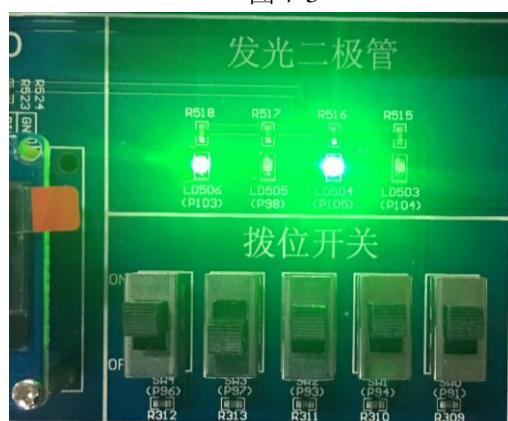


图 7-4



### **实验总结:**

通过这次实验我学会了使用 ISE 软件建立一个完整的 EDA 工程，撰写一个简单的 VHDL 程序，并且能够通过 ISE 软件修改语法错误，编写测试激励（仿真代码），并且能够对 VHDL 程序进行仿真调试，会编写引脚约束文件，能够用 ISE 软件生成.bit 文件，并且学会使用 ISE iMPACT 将.bit 下载到 FPGA。

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：

- 注： 1、报告内的项目或内容设置，可根据实际情况加以调整和补充。  
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。