

# 实验二 (2) 加法器设计实验

计算机1202 张艺瀚

学号: 20123852

January 4, 2015

## 1 实验目的

1. 学习了解加法器工作原理。
2. 学习用VHDL语言设计全加器的设计方法。
3. 学习使用元件例化的方法设计多位加法器。

## 2 实验原理

两个n位二进制数相加的过程, 是从最低有效位开始相加, 形成和数并传送进位最后得到结果。最低位只有加数和被加数相加, 这种两个一位数相加称为半加; 完成加数, 被加数, 低位的进位数三个一位数相加称为全加。实现半加运算的电路称为半加器, 实现全加运算的电路称为全加器。

全加器 (Full Adder) 的真值表如下表所示, 其中 $C_i$ 是低位进位,  $C_{i+1}$ 是向高位的进位。

全加器真值表如图 1 所示。

全加器 (Full Adder) 的逻辑图请参考图 2。

由全加器的真值表可写出 $S_i$ ,  $C_{i+1}$ 的逻辑表达式:

$$\begin{aligned} S_i &= A_i \oplus B_i \oplus C_i \\ C_{i+1} &= A_i B_i + C_i (A_i \oplus B_i) \end{aligned}$$

## 3 实验内容

1. 用VHDL语言设计全加器。
2. 用元件例化方法设计一个四位二进制加法器。

输入			输出	
Ai	Bi	Ci	Si	Ci+1
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

Figure 1: 全加器真值表

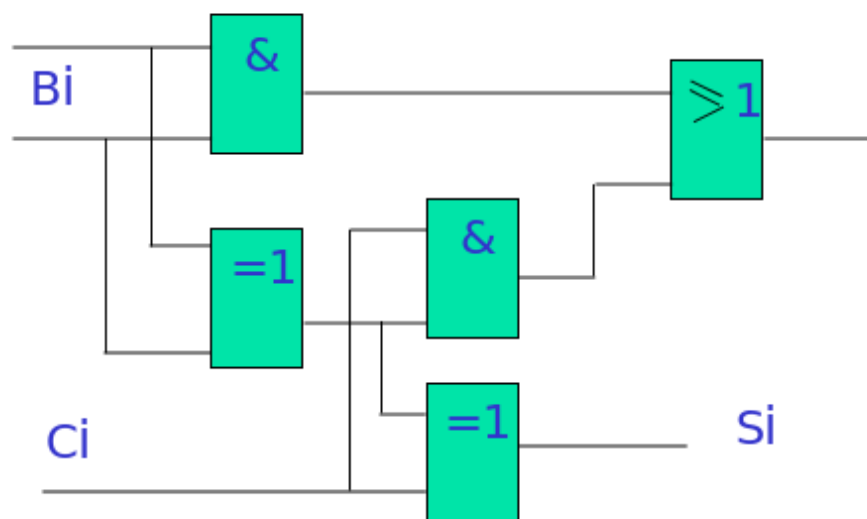


Figure 2: 全加器逻辑图

## 4 实验设备

1. 清华同方PIV2.4G/256M60G

## 5 实验程序

全加器代码清单如下（代码清单 1）

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.ALL;
4 use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6 -- Uncomment the following lines to use the
7 -- declarations that are
8 -- provided for instantiating Xilinx primitive
9 -- components.
10 --library UNISIM;
11 --use UNISIM.VComponents.all;
12
13 entity aone is
14 port(a, b, cin: in std_logic;
15       sum, cout: out std_logic);
16 end aone;
17
18 architecture Behavioral of aone is
19
20 begin
21
22 sum<=(a xor b) xor cin;
23 cout<= (a and b) or ((a xor b) and cin);
24
25 end Behavioral;
```

Listing 1: 全加器代码清单

4位加法器代码清单如下（代码清单 2）

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.ALL;
4 use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6 -- Uncomment the following lines to use the
7 -- declarations that are
8 -- provided for instantiating Xilinx primitive
9 -- components.
```

```

8  --library UNISIM;
9  --use UNISIM.VComponents.all;
10
11  entity addern is
12  port(a, b: in std_logic_vector(4 downto 1);
13        cin: in std_logic;
14        sum: out std_logic_vector(4 downto 1);
15        cout: out std_logic);
16  end addern;
17
18  architecture Behavioral of addern is
19
20  component aone
21  port(a: in std_logic;
22        b: in std_logic;
23        cin: in std_logic;
24        sum: out std_logic;
25        cout: out std_logic);
26  end component;
27
28  signal carry: std_logic_vector(0 to 4);
29
30  begin
31
32  carry(0) <= cin;
33  cout <= carry(3);
34
35  gen: for i in 1 to 4 generate
36  add: aone port map(
37    a => a(i),
38    b => b(i),
39    cin => carry(i-1),
40    sum => sum(i),
41    cout => carry(i)
42  );
43  end generate;
44
45  end Behavioral;

```

Listing 2: 4位加法器代码清单

## 6 仿真结果

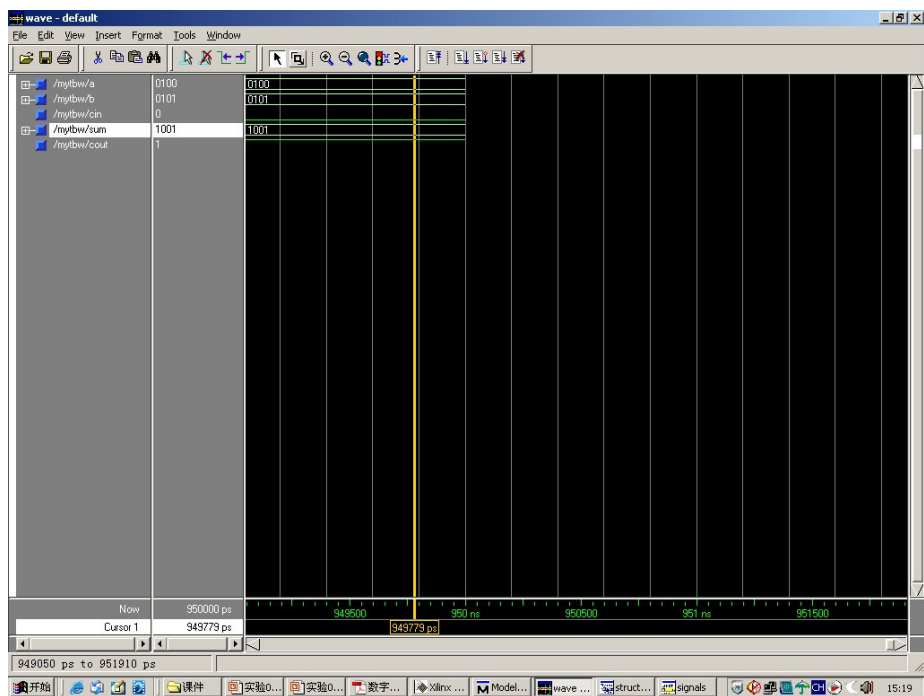


Figure 3: 4位加法器仿真波形图