

实验三 加法器、运算器的实现

班级 软件 1801 姓名 肖云杰 学号 201826010113

一、实验目的

1. 了解运算器的内部结构。
2. 熟悉并行加法器和串行加法器的工作原理。
3. 分析模型机的功能，设计 ALU 和移位逻辑。

二、实验内容

- 1、用 VHDL 语言设计 4 位并行加法器
- 2、用 VHDL 语言设计 4 位串行加法器
- 3、用 VHDL 语言设计模型机运算器 ALU
- 4、用 VHDL 语言设计移位逻辑

三、实验方法

1、实验方法

采用基于 FPGA 进行数字逻辑电路设计的方法。
采用的软件工具是 Quartus II。

2、实验步骤

一、 底层文件，全加器

1、新建，编写源代码。

- (1).选择保存项和芯片类型：【File】-【new project wizard】-【next】（设置文件路径+设置 project name 为 all_add）-【next】（设置文件名 xor2.vhd—在【add】）-【properties】（type=VHDL）-【next】-【next】-【finish】
- (2).新建：【file】-【new】（第二个 VHDL File）-【OK】

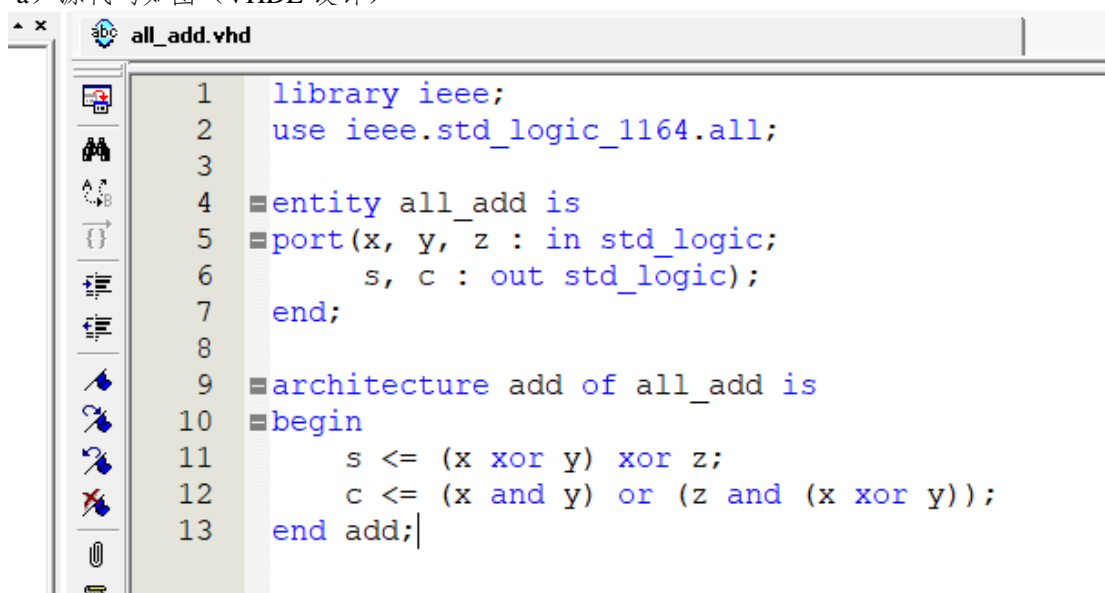
2、写好源代码，保存文件（all_add.vhd）。

3、编译与调试。确定源代码文件为当前工程文件，点击【processing】-【start compilation】进行文件编译，编译成功。

四、实验过程

1、编译过程

a) 源代码如图（VHDL 设计）



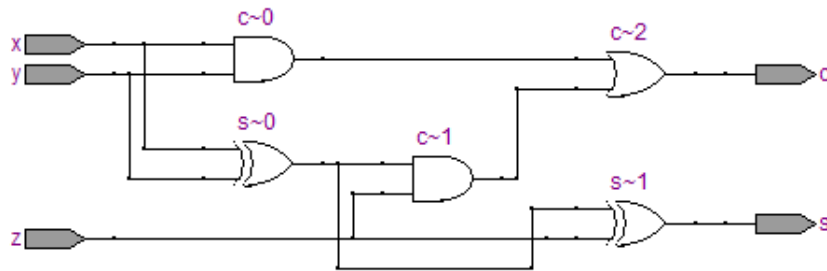
```
1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity all_add is
5 port(x, y, z : in std_logic;
6       s, c : out std_logic);
7 end;
8
9 architecture add of all_add is
10 begin
11     s <= (x xor y) xor z;
12     c <= (x and y) or (z and (x xor y));
13 end add;
```

b) 编译、调试过程

编译过程中有三个警告

警告的原因都是管脚没有分配

c) RTL 视图



d) 结果分析及结论

底层全加器很简单，简单实现即可，不再进行仿真

二、 四位并行加法器

编译过程

源代码如图（VHDL 设计）

```

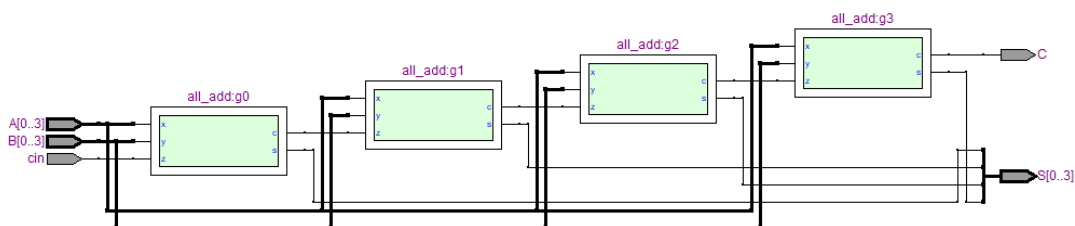
1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity bing_add is
4  port(A, B : in std_logic_vector(0 to 3);
5        cin:in std_logic;
6        S : out std_logic_vector(0 to 3);
7        C : out std_logic);
8  end bing_add;
9  architecture bhv of bing_add is
10 component all_add is
11 port(x,y,z : in std_logic;
12       s,c : out std_logic);
13 end component;
14 signal temp : std_logic_vector(0 to 2);
15 begin
16   g0 : all_add port map(A(3), B(3), cin, S(3), temp(0));
17   g1 : all_add port map(A(2), B(2), temp(0), S(2), temp(1));
18   g2 : all_add port map(A(1), B(1), temp(1), S(1), temp(2));
19   g3 : all_add port map(A(0), B(0), temp(2), S(0), C);
20 end;

```

编译、调试过程

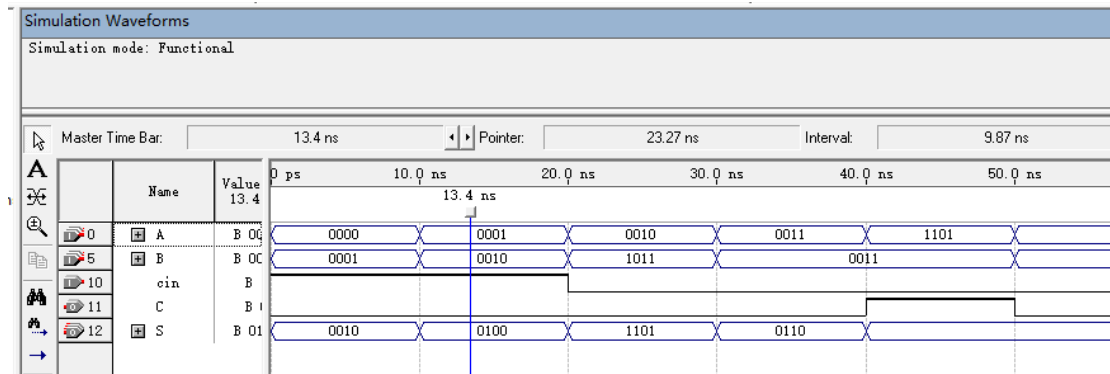
编译过程中有三个警告，分别都是管脚未声明导致

STL 视图



2、 波形仿真

波形仿真波形图



c) 结果分析及结论

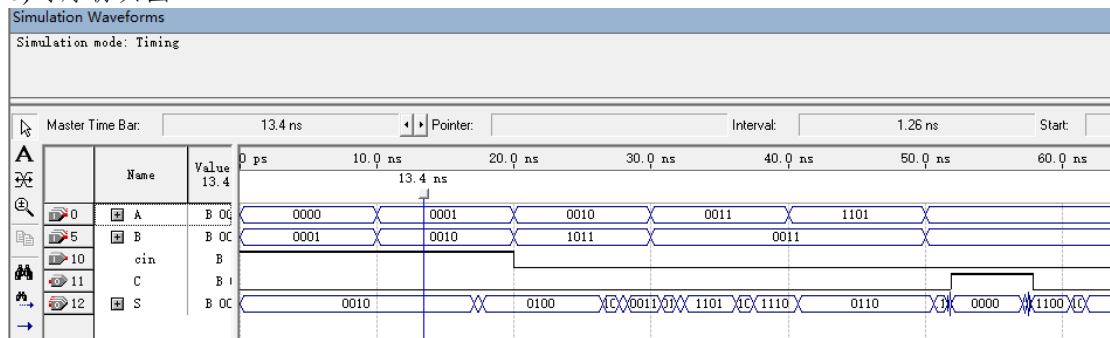
0-10ns: $0+1+1$ (初始进位) = 2 正确
 10-20ns: $1+2+1$ (初始进位) = 4 正确
 20-30ns: $2+11+0$ (初始进位) = 13 正确
 30-40ns: $3+3+0$ (初始进位) = 6 正确
 40-50ns: $13+3+0=16$ (进位为 1 其余为 0) 正确

3、 时序仿真

a) 时序仿真过程

做好上述步骤后, 编译【classic timing analysis】-在 compilation report 中选择【timing analysis】-【tpd】(引脚到引脚的延时)

b) 时序仿真图



b) 结果分析及结论

延迟大体一直, 约 8-9ns

三、 四位串行加法器

编译过程

```

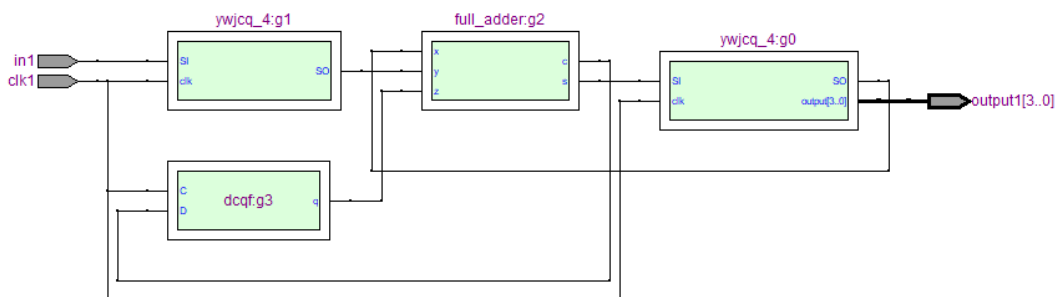
1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity cx_adder is
4  port(in1,clk1 : in std_logic;
5        output1 : out std_logic_vector(3 downto 0)
6        );
7  end cx_adder;
8  architecture rtl of cx_adder is
9  component ywjqc_4
10     port(
11         SI,clk : in std_logic;
12         SO : out std_logic;
13         output : out std_logic_vector(3 downto 0)
14     );
15 end component;
16 component dcqf
17     port(
18         D,C: in std_logic;
19         q : out std_logic
20     );
21 end component;
22 component full_adder
23     port(
24         x,y,z: in std_logic;
25         s,c: out std_logic
26     );
27 end component;
28 signal jc0,jc1,add_s,add_c,dc : std_logic;
29 begin
30     g0 : ywjqc_4 port map(add_s,clk1,jc0,output1);
31     g1 : ywjqc_4 port map(in1,clk1,jc1);
32     g2 : full_adder port map(jc0,jc1,dc,add_s,add_c);
33     g3 : dcqf port map(add_c,clk1,dc);
34 end rtl;

```

编译、调试过程

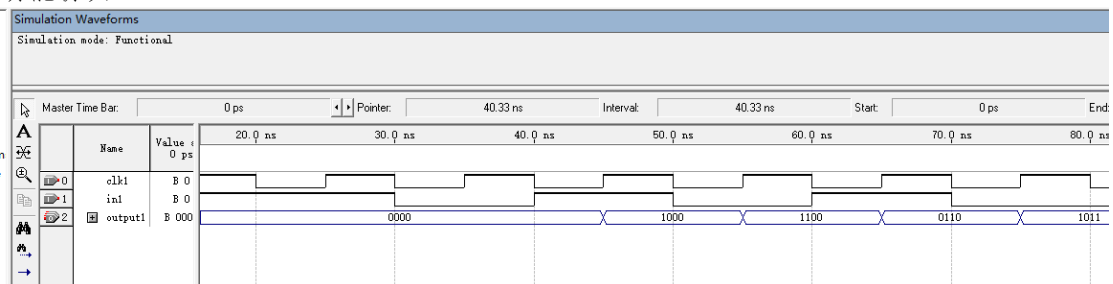
编译和调试过程中，没有报错，
仅有三个因为未分配管脚造成的警告

STL 视图

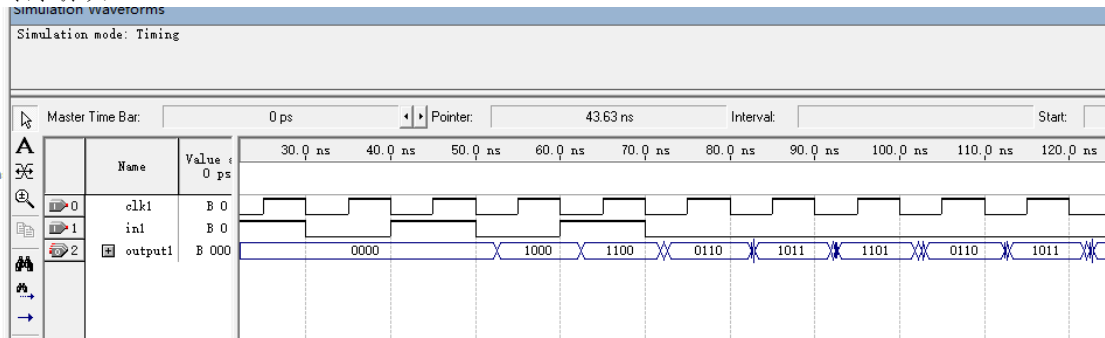


波形仿真

功能仿真



时序仿真



波形分析及结论

每次加数输入，在时钟沿进行相加，结果正确。

四、 ALU 运算

编译过程

VHDL 实现

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  entity alu_ex3 is
5  port(m: in std_logic;
6       s: in std_logic_vector(3 downto 0);
7       a: in std_logic_vector(7 downto 0);
8       b: in std_logic_vector(7 downto 0);
9       T: out std_logic_vector(7 downto 0);
10      cf,cz: out std_logic);
11 end alu_ex3;
12 architecture bhv of alu_ex3 is
13 signal aa,bb,cc : std_logic_vector(8 downto 0);
14 signal jinwei,weiling : std_logic;
15 begin
16   process(m,s,a,b)
17   begin
18     aa<= '0'&a;
19     bb<= '0'&b;
20     if m='1' then
21       if s="1001" then
22         cc<=aa+bb;
23         jinwei<=cc(8);
24         if cc(7 downto 0)="00000000" then weiling<='1';
25         end if;
26       elsif s="0110" then
27         cc<=aa-bb;
28         jinwei<=cc(8);
29         if cc(7 downto 0)="00000000" then weiling<='1';
30         end if;
31       elsif s="1011" then
32         cc<=aa or bb;
33       elsif s="0101" then
34         cc<=not aa;
35       end if;
36     end if;
37     T<=cc(7 downto 0);
38     cf<=jinwei;
39     cz<=weiling;

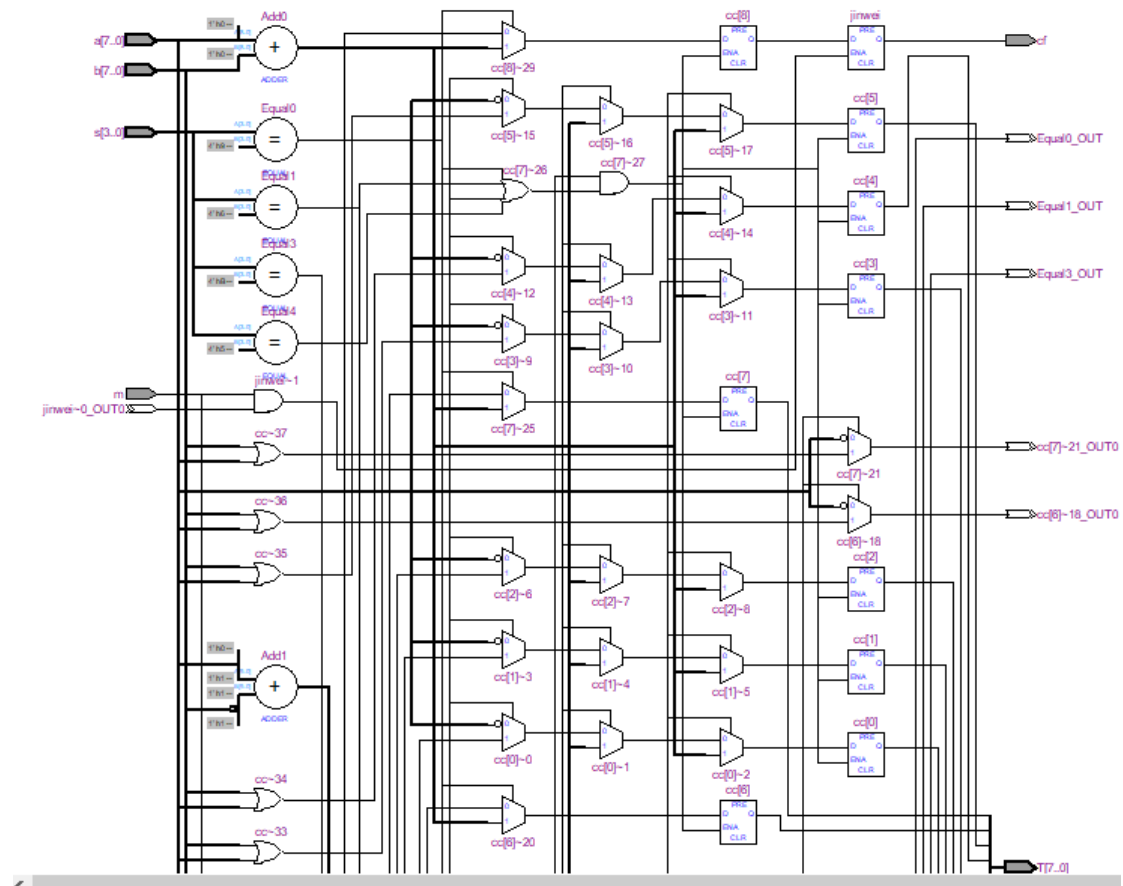
```

编译和调试过程

编译过程中未发现错误，仅有三个未分配管脚的警告

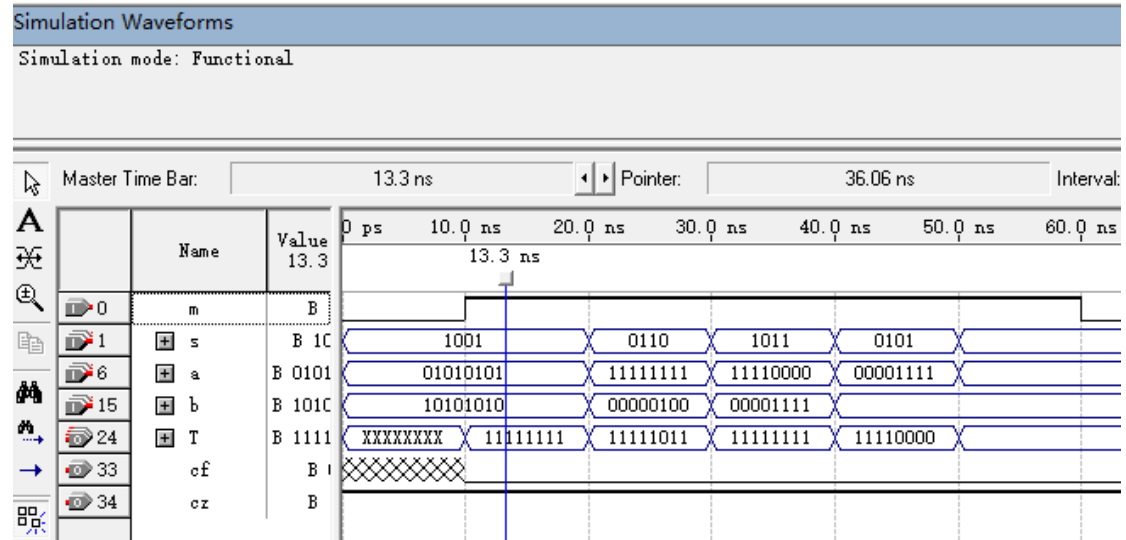
STL 视图

Page Title: alu_ex3

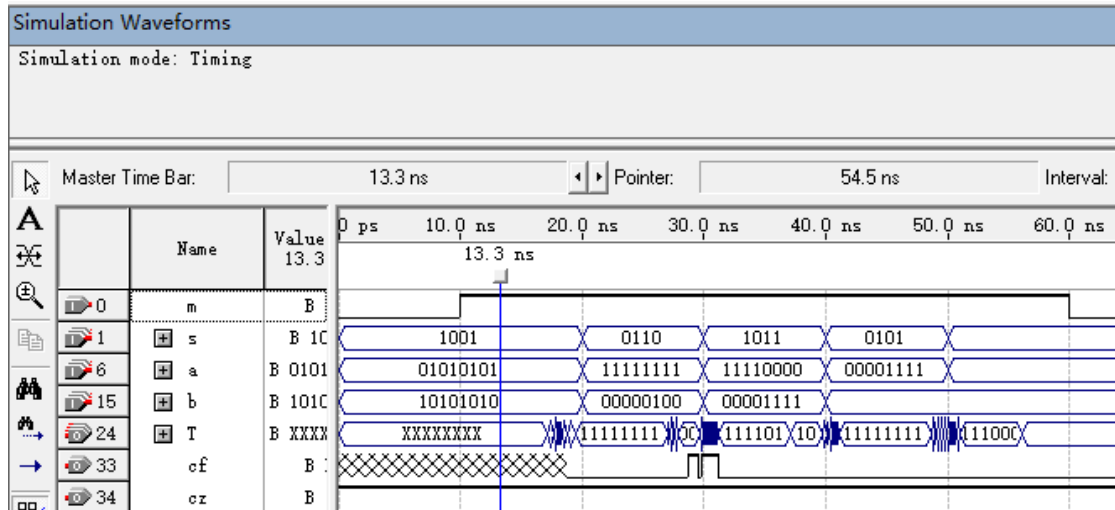


波形仿真

功能仿真



时序仿真



波形结果分析

0-10ns: m=0, 无效
 10-20ns: 功能为加法, 运行正确
 20-30ns: 功能为减法, 运行正确
 30-40ns: 功能为求或, 运行正确
 40-50ns: 功能为求非, 运行正确

时序仿真中, 管教之间数据的传输, 有 8ns 左右的延迟, 正确。

五、 移位器

编译过程

VHDL 实现

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  entity yiwei is
5  port (fbus, flbus, frbus : in std_logic;
6        a:in std_logic_vector(7 downto 0);
7        w:out std_logic_vector(7 downto 0);
8        cf:out std_logic);
9  end yiwei;
10 architecture bhv of yiwei is
11 begin
12   process (fbus, flbus, frbus, a)
13   begin
14     if fbus='1' then
15       w<=a;
16       cf<='0';
17     elsif flbus='1' then
18       w<=a(6 downto 0) & a(7);
19       cf<=a(7);
20     elsif frbus='1' then
21       w<=a(0) & a(7 downto 1);
22       cf<=a(0);
23     else
24       w<="ZZZZZZZZ";
25       cf<='0';
26     end if;
27   end process;
28 end;
29

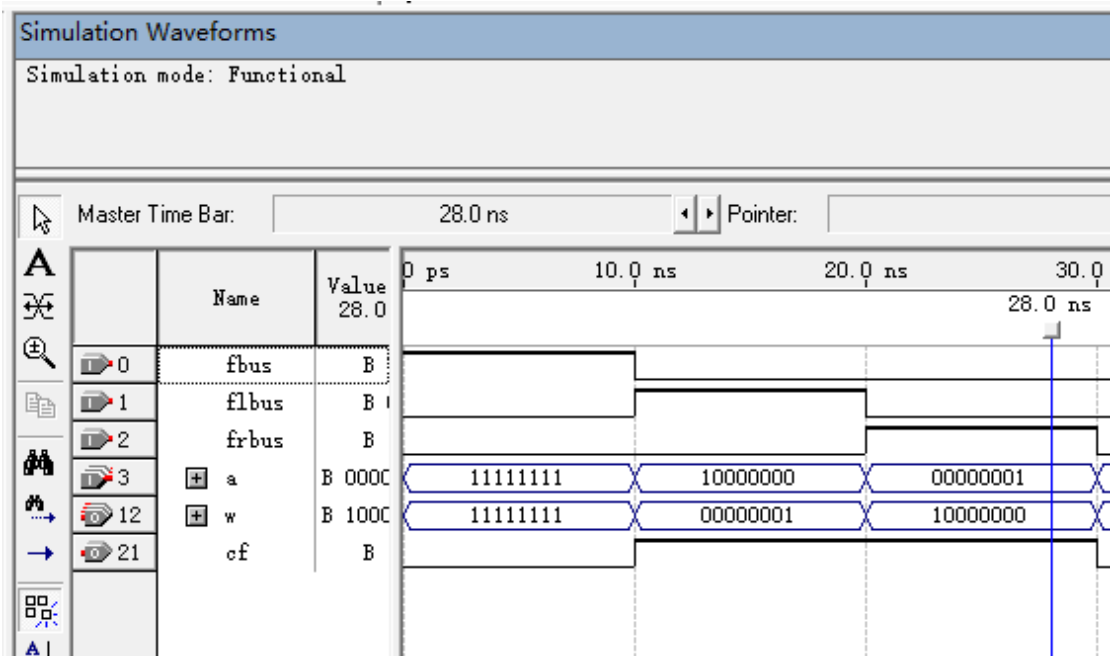
```

编译过程及分析

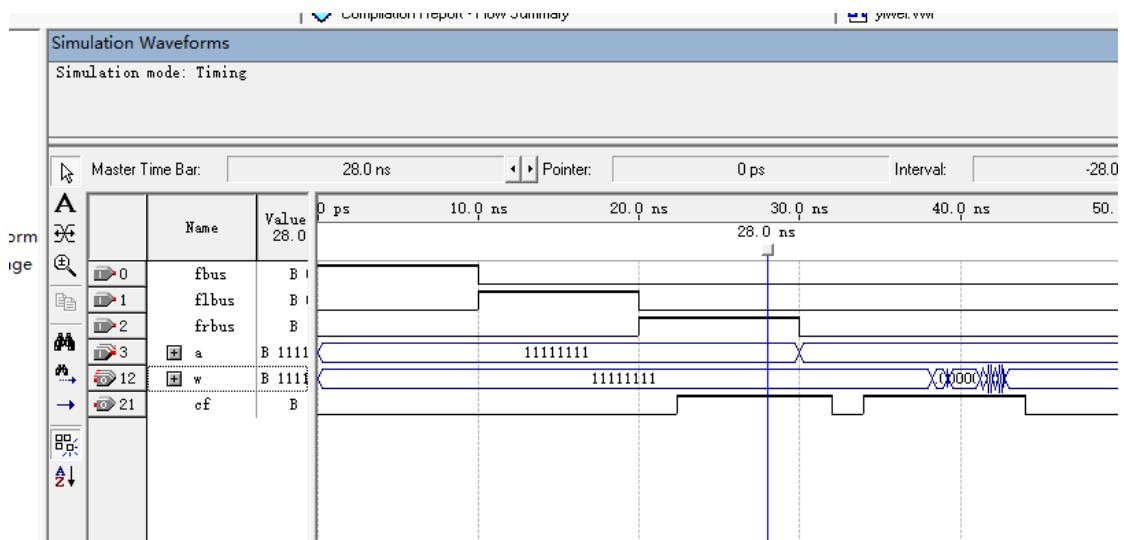
编译过程中，没有报错，仅有三个因管脚未分配造成的警告。

波形仿真

功能仿真



时序仿真



仿真分析

功能仿真和时序仿真没有太大区别，时序仿真容易看出，有 8-9ns 的管脚信号传输的延迟

观察功能仿真图

0-10ns: 简单输出

10-20ns: 左移，将最左位补到最右位，正确

20-30ns: 右移，将最右位补到最左位，正确

五、实验结论

1、思考题

1. 如运算器需要进行溢出判断，请问运算器要如何修改？

另外设置一位输出，进行溢出的判断，主要在加法中功能中使用，若发生进位溢出，则将所有数据还原/改为高阻态，reset 状态

2. 运算器要能支持有符号数和无符号 8 位操作数的运算，应该如何设计？

可以增加补码的运算，将补码运算和功能的选择联系起来即可。

2、实验总结与实验心得

本次实验过程中，主要的问题出在了读题方面，尤其是串行加法器，使用递归+clk 实现的方法被否决了，原因是实验要求使用寄存器和触发器实现，导致二次实验浪费时间同时，在第一次实验中，加法器的默认第一个进位，我一开始默认为 0，但是实际上它可能不是 0，所以后来增加了新的进位输入

移位逻辑中，一开始以为移走后，相应位补 0，最后发现应该补移走位，进行相应修改。

主要问题就是出在了读题和审题方面，浪费大量不必要的时间，下次多加注意