四位加法器.实验报告

2018011324

计82

尤艺霖

1.实验要求/目的

1.1.实验目的

掌握组合逻辑电路的基本分析和设计方法

理解半加器和全加器的工作原理并掌握利用全加器组成不同字长加法器的各种方法

学习元件例化的方式进行硬件电路设计

1.2.实验要求

用硬件描述语言实现半加器和全加器的设计

并用全加器实现逐次进位加法器和超前进位加法器

全加器需要用半加器实现

2.思路

使用元件例化的方法，先实现半加器和或门，然后再用这两个实现全加器

对于逐次进位加法器，可以直接使用全加器实现，对于超前进位加法器，还需要写一个进位器电路

具体实现见代码部分

3.代码

为了便于在Jielab上测试，将前五部分的代码全部放入了**fourdigitadder.vhd**

3.1.半加器

最简单的半加器代码，包含一次异或和一次与操作。

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

entity halfadder is
port(
    a,b:in std_logic;
    s,cout:out std_logic
);
end halfadder;

architecture halfadd of halfadder is
```

```
begin
    process(a,b)
    begin
        s<=a xor b;
        cout<=a and b;
    end process;
end halfadd;
```

3.2.或门

用于实现全加器中的或操作。

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

entity orgate is
port(
    a,b:in std_logic;
    s:out std_logic
);
end orgate;

architecture orprocess of orgate is
begin
    process(a,b)
    begin
        s<=a or b;
    end process;
end orprocess;
```

3.3.全加器

使用两次半加器和一次或门，同时计算出**p**和**g**用来给超前进位加法器备用，需要设为**buffer**。

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

entity fulladder is
port(
    a,b,cin:in std_logic;
    s,cout:out std_logic;
    p,g,k:buffer std_logic
);
end fulladder;

architecture fulladd of fulladder is
    component halfadder
    port(
        a,b:in std_logic;
        s,cout:out std_logic
    );
    end component;
    component orgate
    port(
        a,b:in std_logic;
        s:out std_logic
    );
    end component;
begin
    hf0:halfadder port map(a,b,p,g);
    hf1:halfadder port map(p,cin,s,k);
    op0:orgate port map(k,g,cout);
end fulladd;
```

3.4.进位器

按照课本的电路图，接受所有的**g**和**p**，提供所有的进位信息。

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

entity advancedcarrier is
port(
    p,g:in std_logic_vector(3 downto 0);
    cin:in std_logic;
    carry:out std_logic_vector(3 downto 0);
    cout:out std_logic
);
end advancedcarrier;

architecture carryon of advancedcarrier is
begin
    process(p,g)
    begin
        carry(0)<=g(0) or (p(0) and cin);
        carry(1)<=g(1) or (p(1) and g(0)) or (p(1) and p(0) and cin);
        carry(2)<=g(2) or (p(2) and g(1)) or (p(2) and p(1) and g(0)) or (p(2) and p(1) and p(0) and cin);
        cout<=g(3) or (p(3) and g(2)) or (p(3) and p(2) and g(1)) or (p(3) and p(2) and p(1) and g(0)) or (p(3) and p(2) and p(1) and p(0) and cin);
    end process;
end carryon;
```

3.5.四位加法器

将逐次进位和超前进位分别写在了两个**architecture**中，测试时需要注释掉其中一个

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

entity fourdigitadder is
port(
    a,b:in std_logic_vector(3 downto 0);
    cin:in std_logic;
    s:out std_logic_vector(3 downto 0);
    cout:out std_logic
);
end fourdigitadder;

architecture normalcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    signal carry:std_logic_vector(2 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),cout=>carry(0));
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),cout=>carry(1));
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),cout=>carry(2));
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>cout);
end normalcarry;

architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    component advancedcarrier
    port(
        p,g:in std_logic_vector(3 downto 0);
        cin:in std_logic;
        carry:out std_logic_vector(3 downto 0);
        cout:out std_logic
    );
    end component;
    signal p,g,carry:std_logic_vector(3 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),p=>p(0),g=>g(0));
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),p=>p(1),g=>g(1));
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),p=>p(2),g=>g(2));
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),p=>p(3),g=>g(3));
    cry0:advancedcarrier port map(p,g,cin,carry,cout);
end advancedcarry;
```

3.6.TestBench

采用教程中给定的网址生成，填充了若干组测试加法用的测例。

```
-- Testbench created online at:
--   www.doulos.com/knowhow/perl/testbench_creation/
-- Copyright Doulos Ltd
-- SD, 03 November 2002

library IEEE;
use IEEE.Std_logic_1164.all;
use IEEE.Numeric_Std.all;

entity fourdigitadder_tb is
end;

architecture bench of fourdigitadder_tb is

  component fourdigitadder
  port(
    a,b:in std_logic_vector(3 downto 0);
    cin:in std_logic;
    s:out std_logic_vector(3 downto 0);
    cout:out std_logic
  );
  end component;

  signal a,b: std_logic_vector(3 downto 0);
  signal cin: std_logic;
  signal s: std_logic_vector(3 downto 0);
  signal cout: std_logic ;

begin

  uut: fourdigitadder port map ( a    => a,
                                 b    => b,
                                 cin  => cin,
                                 s    => s,
                                 cout => cout );

  stimulus: process
  begin

    -- Put initialisation code here
cin<='0';
a<="0000";
b<="0000";
wait for 100ns;
cin<='1';
a<="1111";
b<="1111";
wait for 100ns;
cin<='0';
a<="1111";
b<="0001";
```

```
wait for 100ns;
cin<='0';
a<="1011";
b<="0010";
wait for 100ns;
cin<='0';
a<="1001";
b<="0001";
wait for 100ns;
cin<='1';
a<="1011";
b<="0010";
wait for 100ns;
cin<='0';
a<="1000";
b<="1000";
wait for 100ns;

    -- Put test bench stimulus code here

    wait;
  end process;


end;
```

4.实验结果

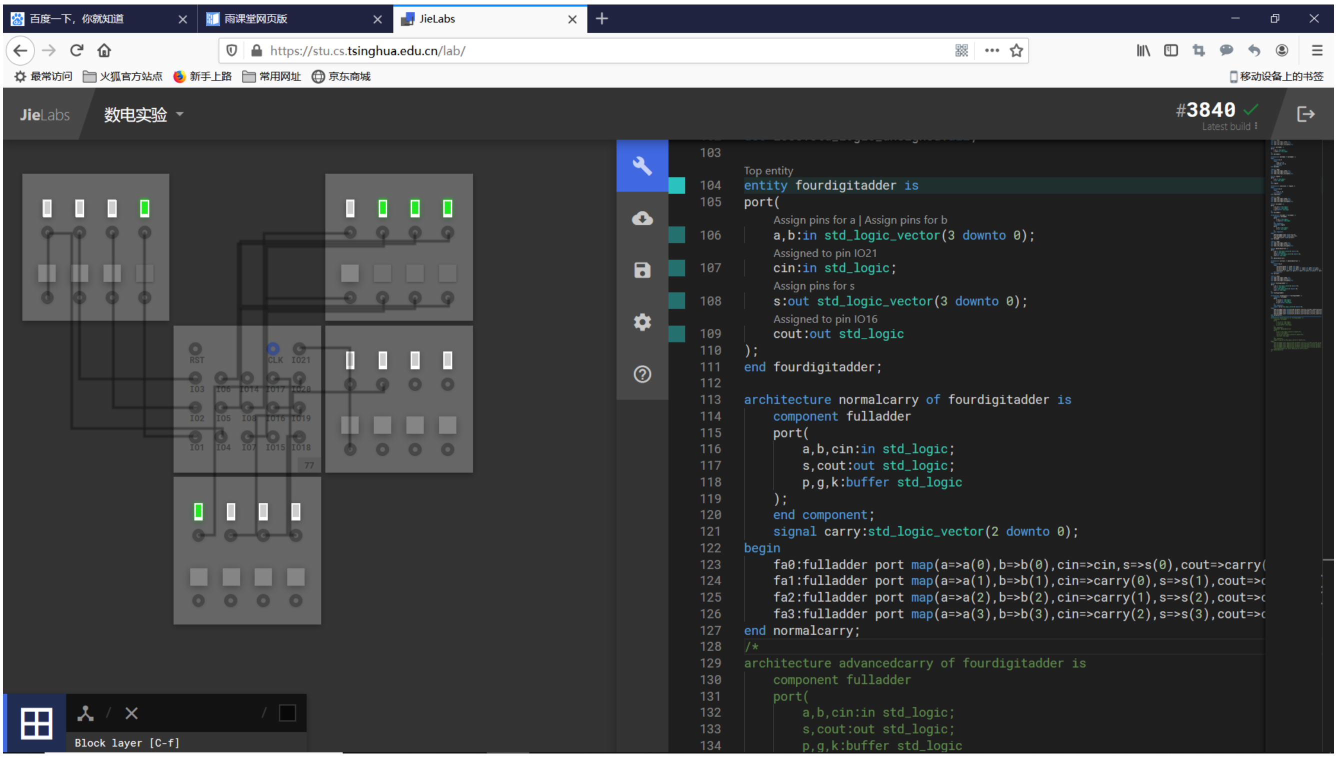
所有的图片都在**screenshot**目录下存储备查

4.1.Jielab实验结果

4.1.1.逐次进位加法器在Jielab

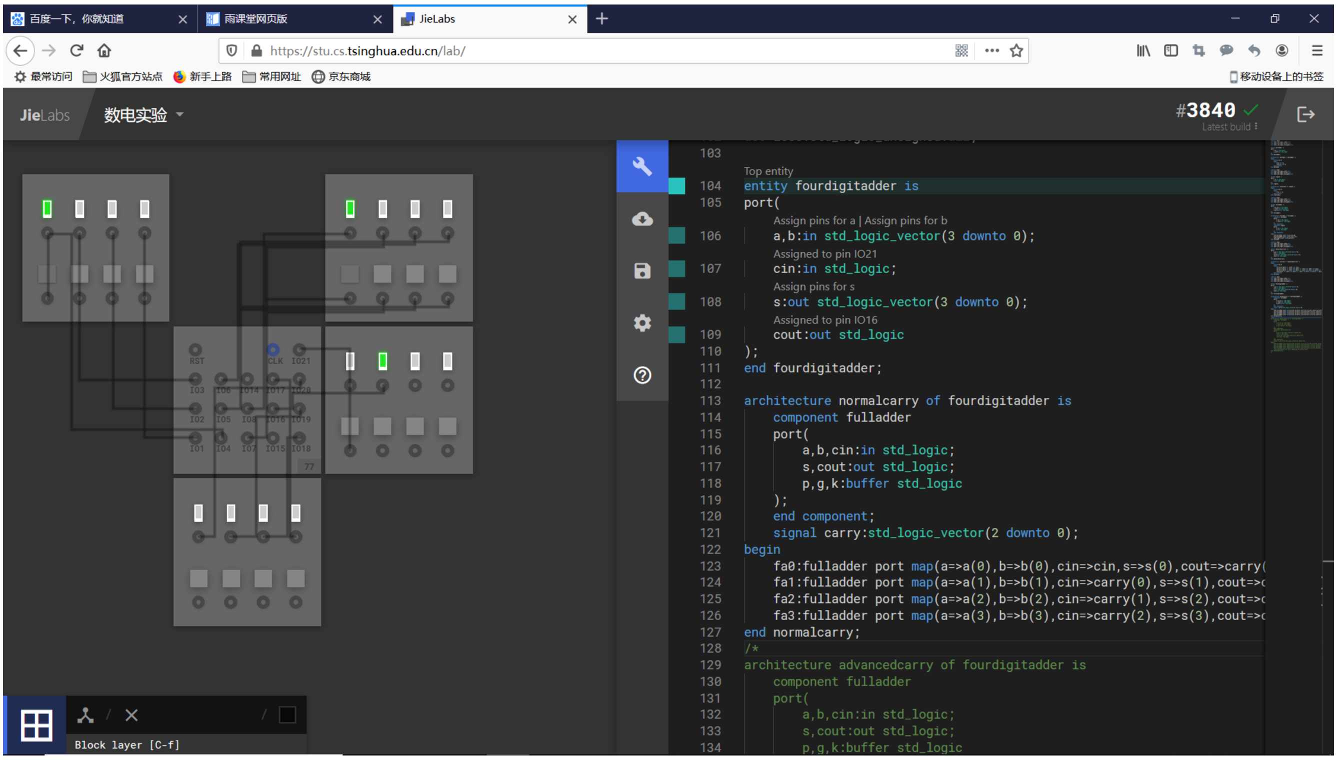
界面截图如下，上端的两个**switch**用来控制输入的数，左侧的**switch**用来控制**cin**和**cout**，下方用来显示输出。

五个测例分别是7+1=8;8+8=16;9+1=10;15+1=16;15+15+1=31：



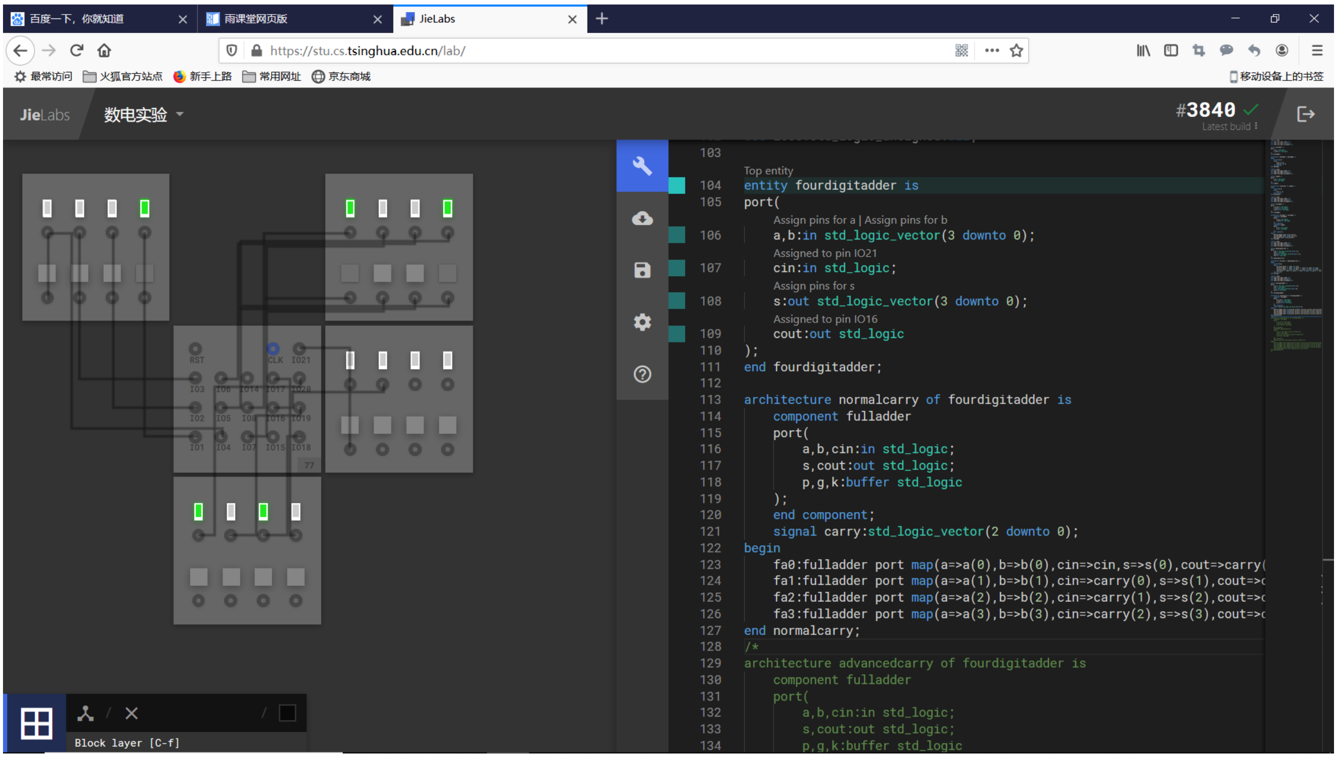
```
Top entity
entity fourdigitadder is
port(
    Assign pins for a | Assign pins for b
    a,b:in std_logic_vector(3 downto 0);
    Assigned to pin IO21
    cin:in std_logic;
    Assign pins for s
    s:out std_logic_vector(3 downto 0);
    Assigned to pin IO16
    cout:out std_logic
);
end fourdigitadder;

architecture normalcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    signal carry:std_logic_vector(2 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),cout=>carry(
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),cout=>c
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),cout=>c
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
/*
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
```



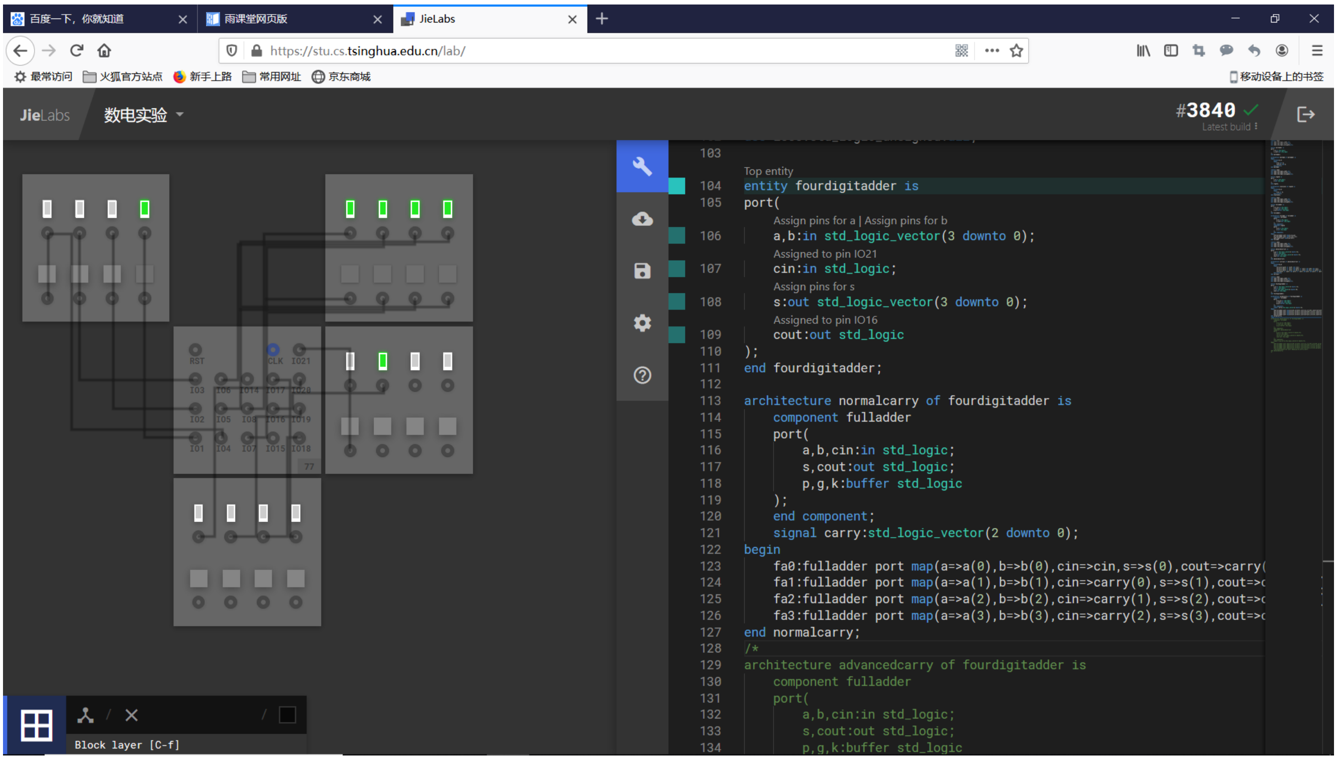
```
Top entity
entity fourdigitadder is
port(
    Assign pins for a | Assign pins for b
    a,b:in std_logic_vector(3 downto 0);
    Assigned to pin IO21
    cin:in std_logic;
    Assign pins for s
    s:out std_logic_vector(3 downto 0);
    Assigned to pin IO16
    cout:out std_logic
);
end fourdigitadder;

architecture normalcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    signal carry:std_logic_vector(2 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),cout=>carry(
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),cout=>c
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),cout=>c
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
/*
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
```



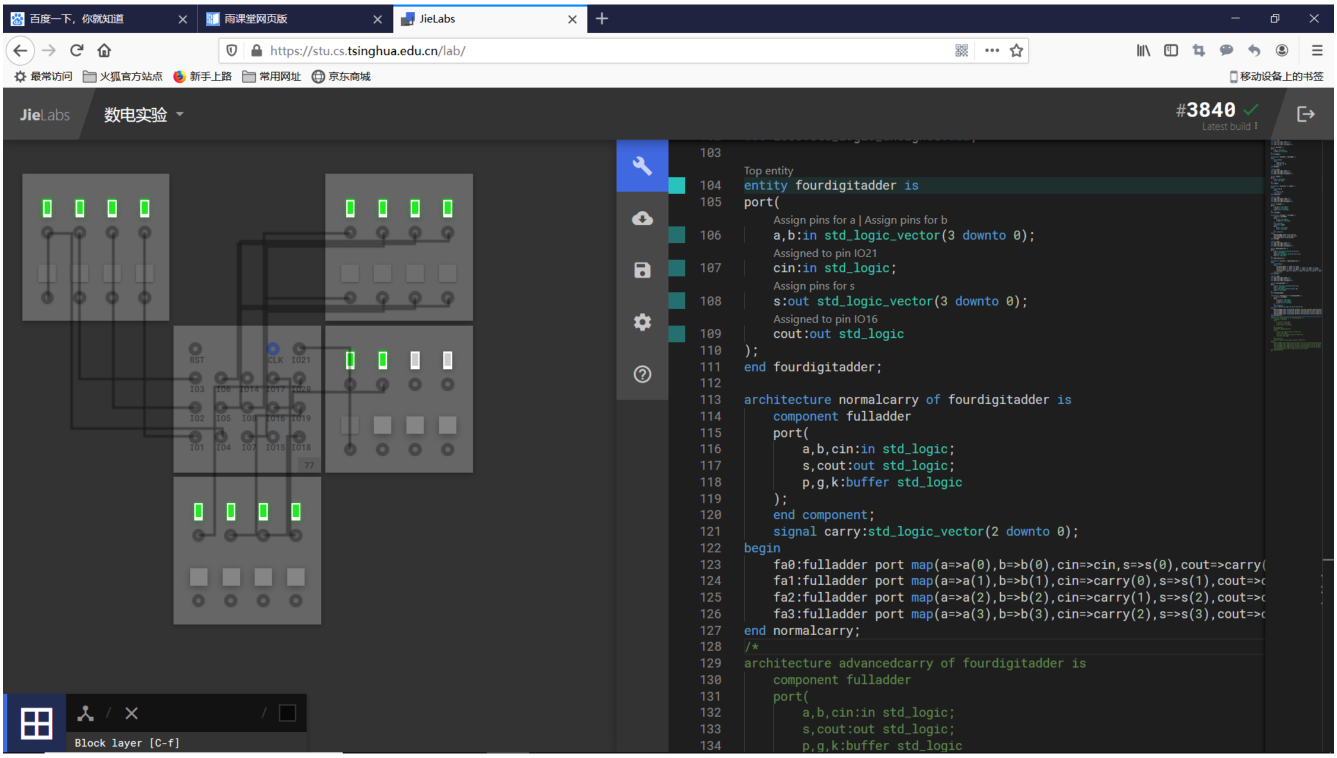
```
Top entity
entity fourdigitadder is
port(
    Assign pins for a | Assign pins for b
    a,b:in std_logic_vector(3 downto 0);
    Assigned to pin IO21
    cin:in std_logic;
    Assign pins for s
    s:out std_logic_vector(3 downto 0);
    Assigned to pin IO16
    cout:out std_logic
);
end fourdigitadder;

architecture normalcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    signal carry:std_logic_vector(2 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),cout=>carry(
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),cout=>c
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),cout=>c
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
/*
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
```



```
Top entity
entity fourdigitadder is
port(
    Assign pins for a | Assign pins for b
    a,b:in std_logic_vector(3 downto 0);
    Assigned to pin IO21
    cin:in std_logic;
    Assign pins for s
    s:out std_logic_vector(3 downto 0);
    Assigned to pin IO16
    cout:out std_logic
);
end fourdigitadder;

architecture normalcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    signal carry:std_logic_vector(2 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),cout=>carry(
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),cout=>c
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),cout=>c
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
/*
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
```



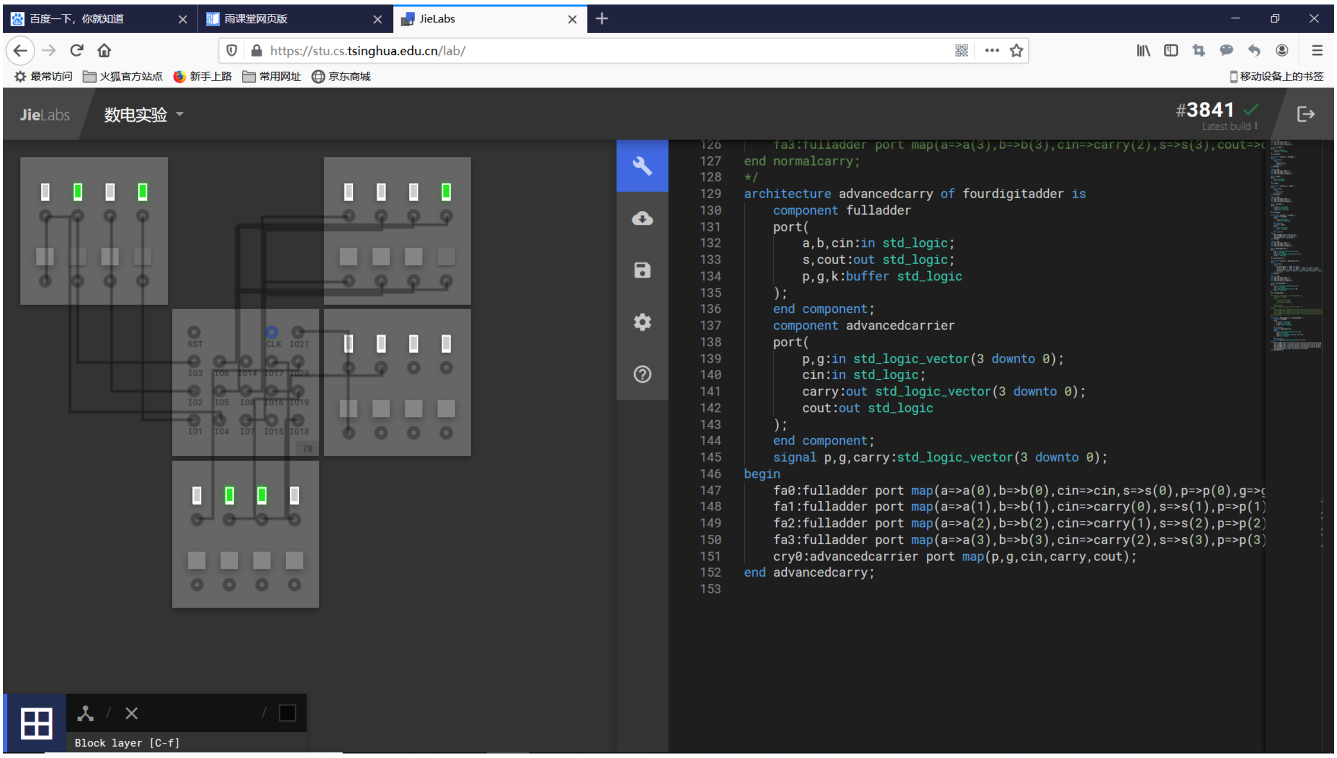
```
Top entity
entity fourdigitadder is
port(
    Assign pins for a | Assign pins for b
    a,b:in std_logic_vector(3 downto 0);
    Assigned to pin IO21
    cin:in std_logic;
    Assign pins for s
    s:out std_logic_vector(3 downto 0);
    Assigned to pin IO16
    cout:out std_logic
);
end fourdigitadder;

architecture normalcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    signal carry:std_logic_vector(2 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),cout=>carry(
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),cout=>c
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),cout=>c
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
/*
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
```

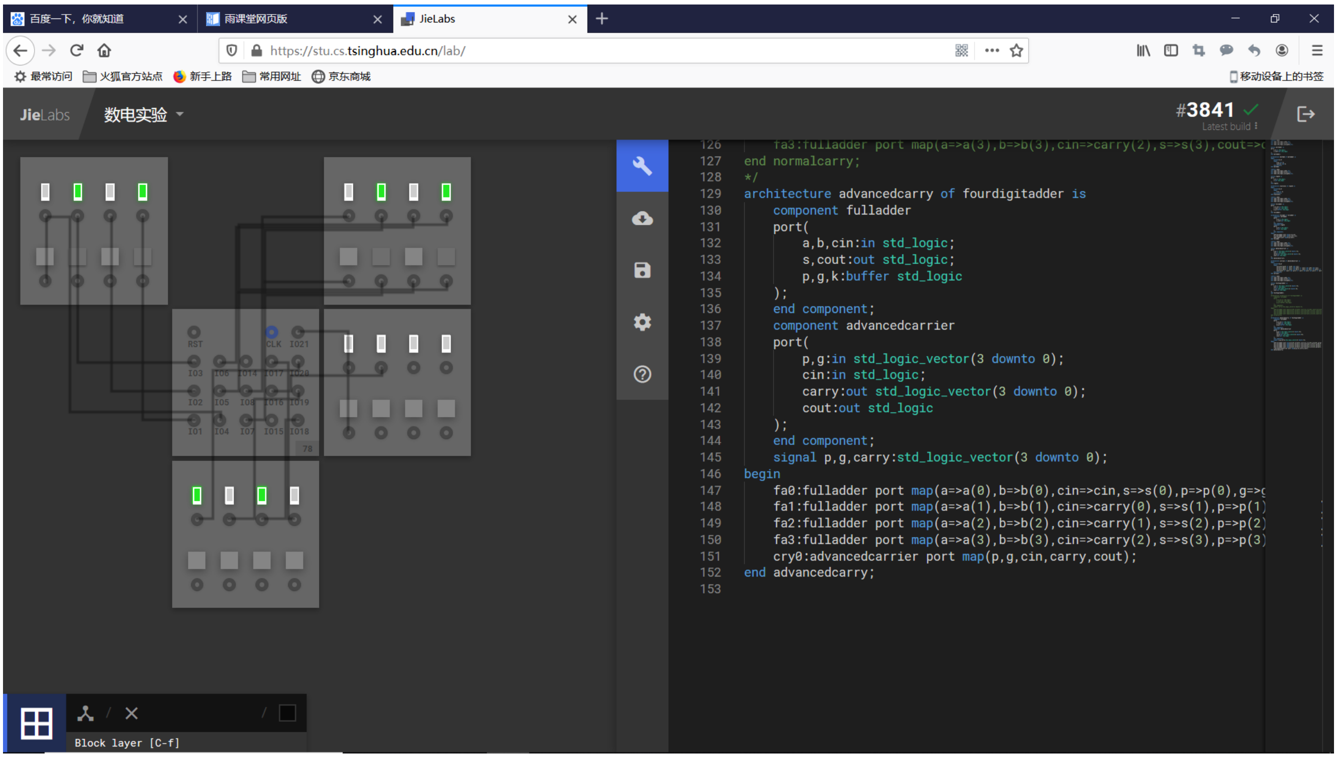
4.1.2.超前进位加法器在Jielab

接线方式同上。

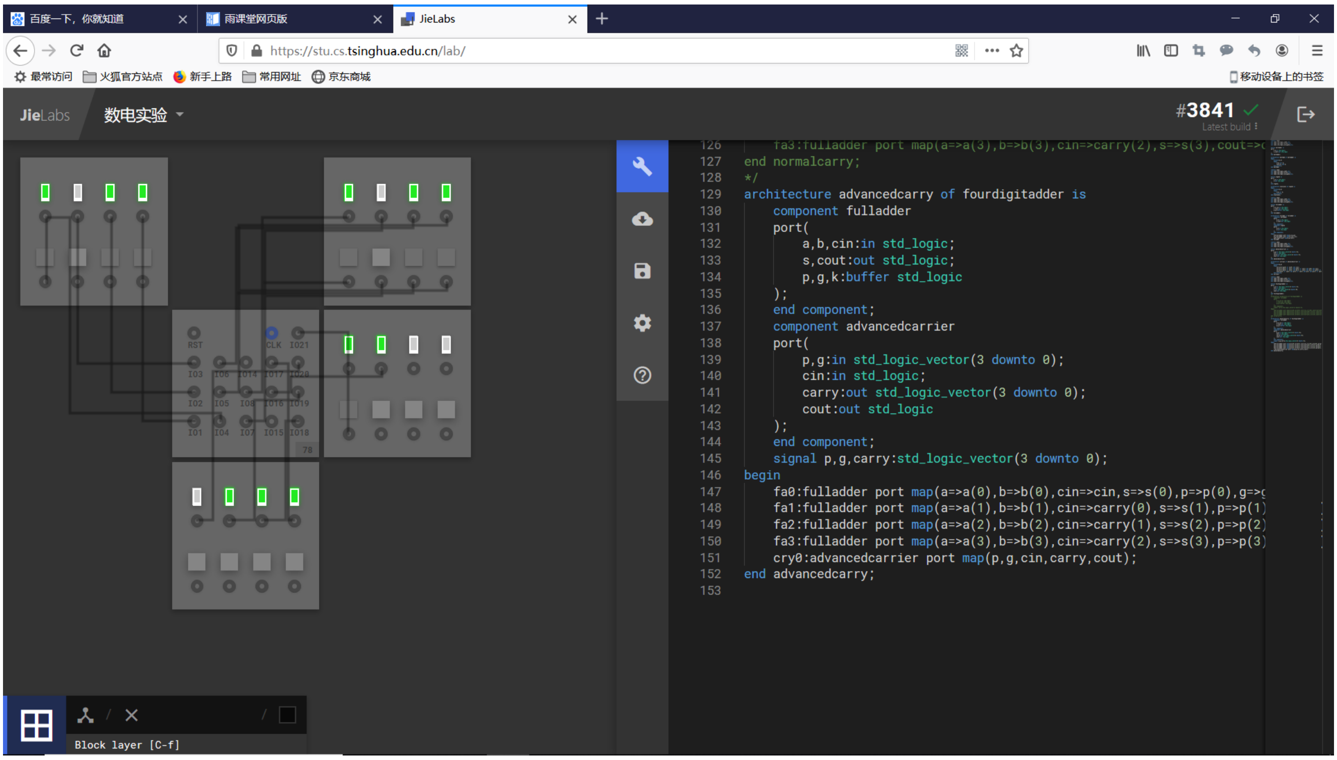
测例分别为5+1=6;5+5=10;11+11+1=23;15+1=16;15+15+1=31：



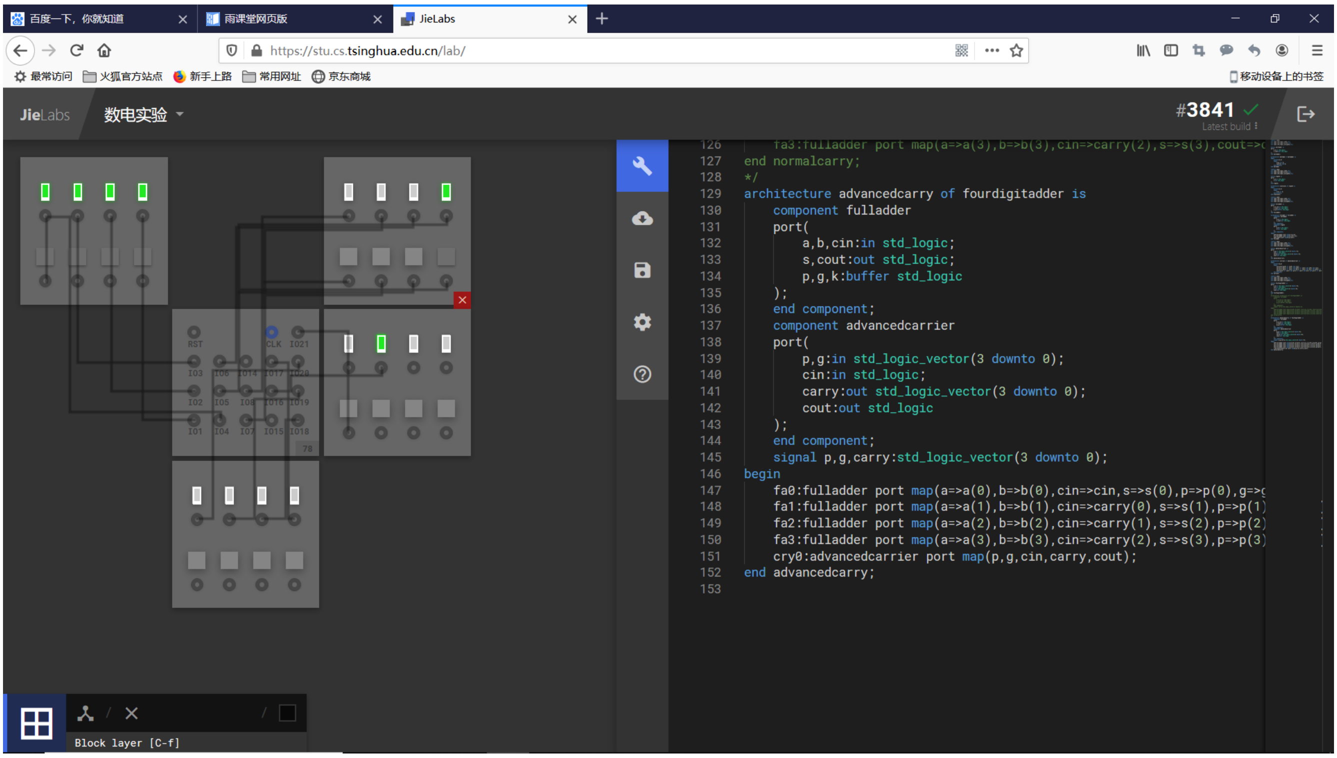
```
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
*/
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    component advancedcarrier
    port(
        p,g:in std_logic_vector(3 downto 0);
        cin:in std_logic;
        carry:out std_logic_vector(3 downto 0);
        cout:out std_logic
    );
    end component;
    signal p,g,carry:std_logic_vector(3 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),p=>p(0),g=>g
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),p=>p(1)
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),p=>p(2)
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),p=>p(3)
    cry0:advancedcarrier port map(p,g,cin,carry,cout);
end advancedcarry;
```



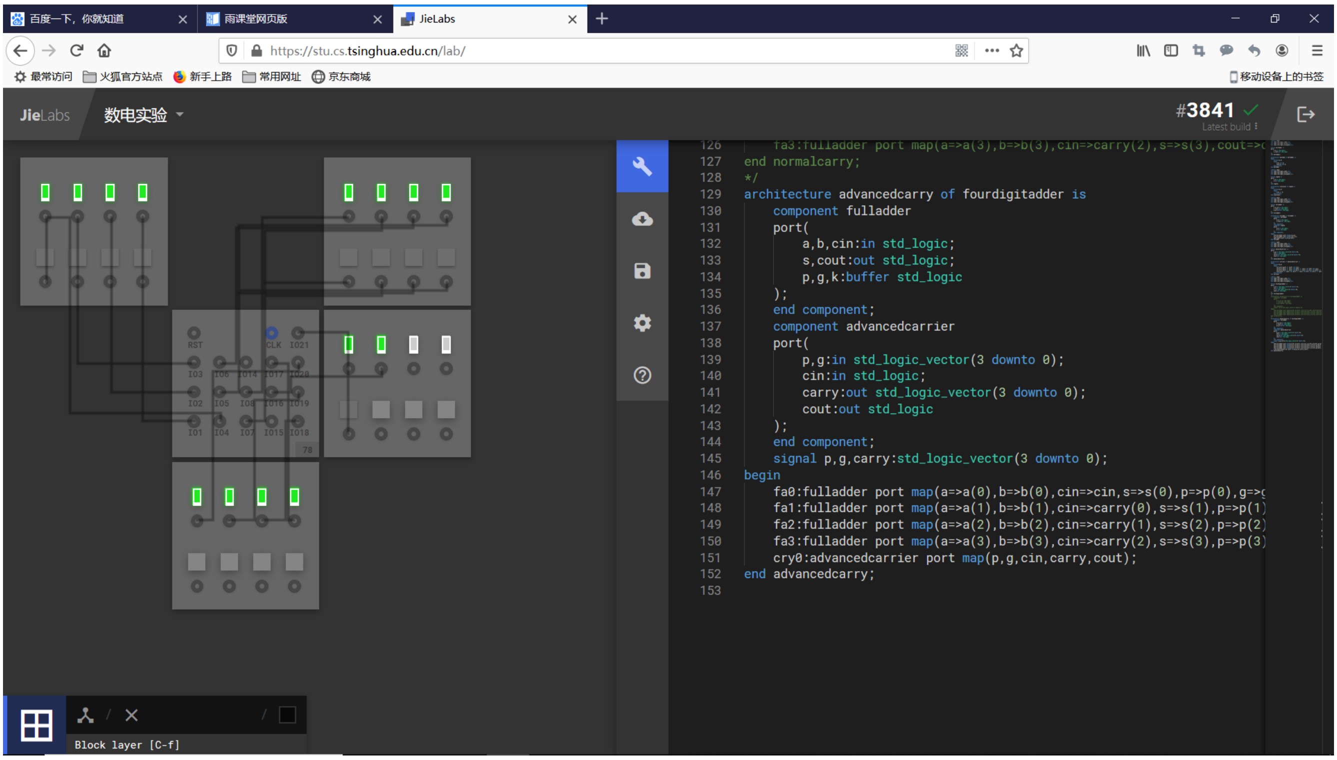
```
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
*/
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    component advancedcarrier
    port(
        p,g:in std_logic_vector(3 downto 0);
        cin:in std_logic;
        carry:out std_logic_vector(3 downto 0);
        cout:out std_logic
    );
    end component;
    signal p,g,carry:std_logic_vector(3 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),p=>p(0),g=>g
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),p=>p(1)
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),p=>p(2)
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),p=>p(3)
    cry0:advancedcarrier port map(p,g,cin,carry,cout);
end advancedcarry;
```



```
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
*/
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    component advancedcarrier
    port(
        p,g:in std_logic_vector(3 downto 0);
        cin:in std_logic;
        carry:out std_logic_vector(3 downto 0);
        cout:out std_logic
    );
    end component;
    signal p,g,carry:std_logic_vector(3 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),p=>p(0),g=>g
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),p=>p(1)
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),p=>p(2)
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),p=>p(3)
    cry0:advancedcarrier port map(p,g,cin,carry,cout);
end advancedcarry;
```



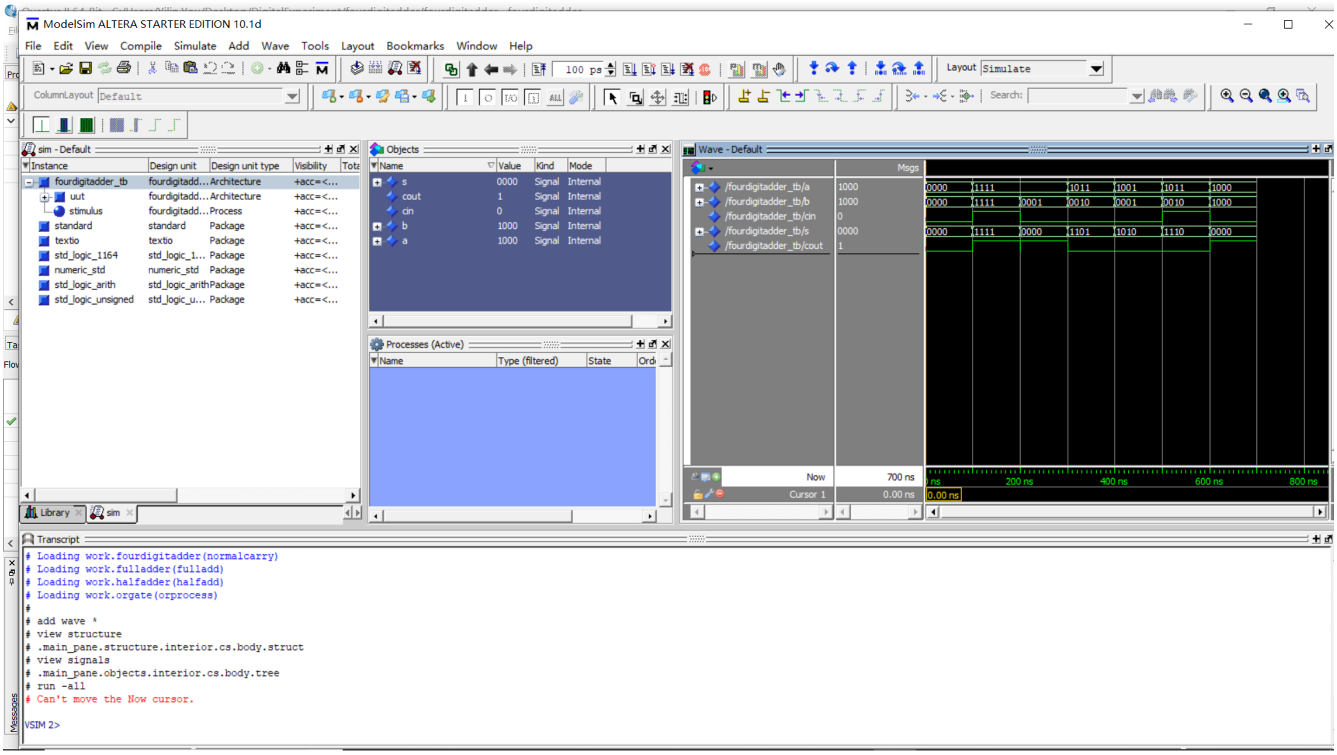
```
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
*/
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    component advancedcarrier
    port(
        p,g:in std_logic_vector(3 downto 0);
        cin:in std_logic;
        carry:out std_logic_vector(3 downto 0);
        cout:out std_logic
    );
    end component;
    signal p,g,carry:std_logic_vector(3 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),p=>p(0),g=>g
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),p=>p(1)
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),p=>p(2)
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),p=>p(3)
    cry0:advancedcarrier port map(p,g,cin,carry,cout);
end advancedcarry;
```



```
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),cout=>c
end normalcarry;
*/
architecture advancedcarry of fourdigitadder is
    component fulladder
    port(
        a,b,cin:in std_logic;
        s,cout:out std_logic;
        p,g,k:buffer std_logic
    );
    end component;
    component advancedcarrier
    port(
        p,g:in std_logic_vector(3 downto 0);
        cin:in std_logic;
        carry:out std_logic_vector(3 downto 0);
        cout:out std_logic
    );
    end component;
    signal p,g,carry:std_logic_vector(3 downto 0);
begin
    fa0:fulladder port map(a=>a(0),b=>b(0),cin=>cin,s=>s(0),p=>p(0),g=>g
    fa1:fulladder port map(a=>a(1),b=>b(1),cin=>carry(0),s=>s(1),p=>p(1)
    fa2:fulladder port map(a=>a(2),b=>b(2),cin=>carry(1),s=>s(2),p=>p(2)
    fa3:fulladder port map(a=>a(3),b=>b(3),cin=>carry(2),s=>s(3),p=>p(3)
    cry0:advancedcarrier port map(p,g,cin,carry,cout);
end advancedcarry;
```

4.2.Modelsim实验结果

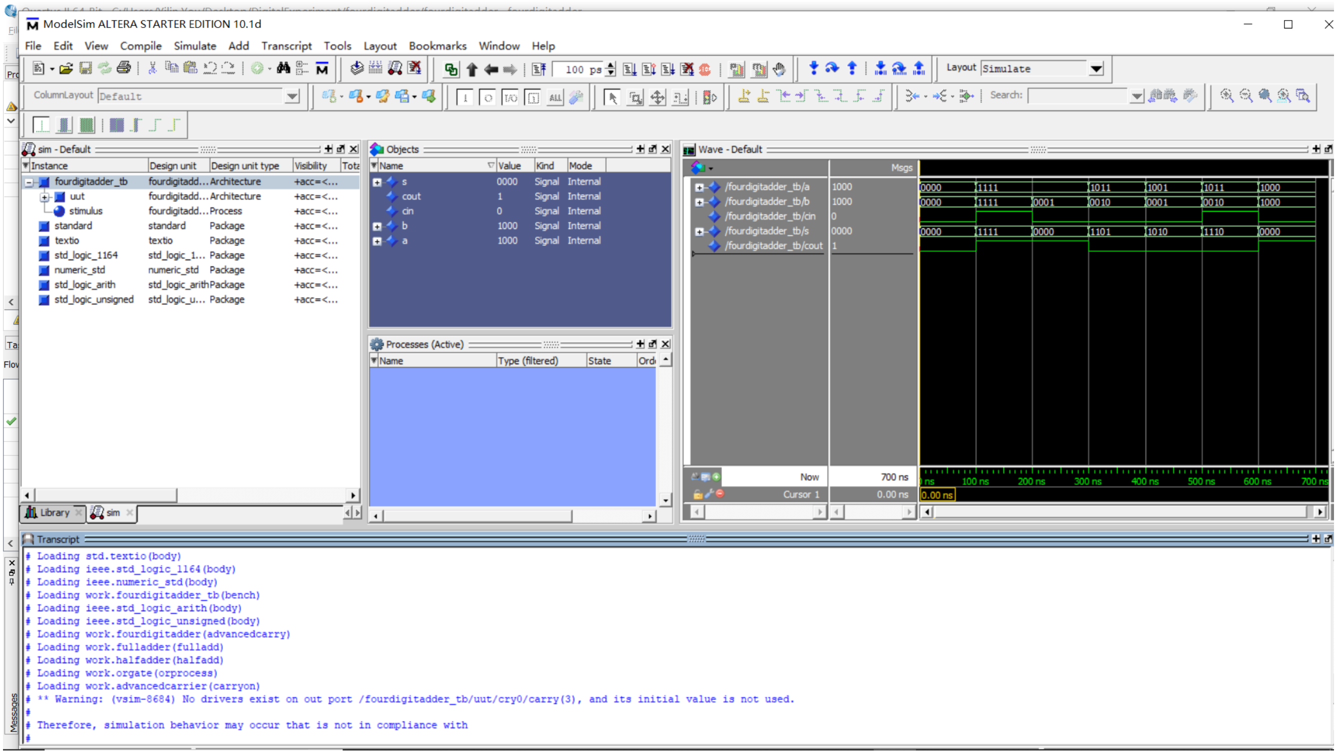
逐次进位加法器的结果（蓝色的字**normalcarry**证明是逐次进位的结构体）：



```
# Loading work.fourdigitadder(normalcarry)
# Loading work.fulladder(fulladd)
# Loading work.halfadder(halfadd)
# Loading work.orgate(orprocess)
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# Can't move the Now cursor.

VSIM 2>
```

超前进位加法器的结果（蓝色的字**advancedcarry**证明是超前进位的结构体）：



```
# Loading std.textio(body)
# Loading ieee.std_logic_1164(body)
# Loading ieee.numeric_std(body)
# Loading work.fourdigitadder_tb(bench)
# Loading ieee.std_logic_arith(body)
# Loading ieee.std_logic_unsigned(body)
# Loading work.fourdigitadder(advancedcarry)
# Loading work.fulladder(fulladd)
# Loading work.halfadder(halfadd)
# Loading work.orgate(orprocess)
# Loading work.advancedcarrier(carryon)
# ** Warning: (vsim-8684) No drivers exist on out port /fourdigitadder_tb/uut/cry0/carry(3), and its initial value is not used.
#
# Therefore, simulation behavior may occur that is not in compliance with
#
```

两个测试的测例都是一样的，分别是：

0+0=0;15+15+1=31;15+1=16;11+2=13;9+1=10;11+2+1=14;8+8=16

5.遇到的问题&总结

通过这次实验，我学习了元件例化的语法，并用其成功实现了两种不同的四位加法器，加深了对加法器的理解，同时也增加了VHDL的代码熟练度和TestBench使用的熟练度。

本次实验最早遇到的问题是测试全加器的时候在**Jielab**上测试发现**buffer**类型的端口接引脚总是红色的，问了助教之后知道这一类型不能接在引脚上，故改用本地拟真测试。之后遇到的困难都是些细节，比如元件例化的语法、连接**Jielab**时网络卡顿等。