Solutions to Assignment 1
VHDL Synthesis

The entity should look something like:

```
-- ELEC 464 Assignment 1
-- model of 74’151 8 to 1 multiplexer with
-- complementary outputs and enable
-- Ed Casas, September 10 1996

entity EC151 is
   port (  a, b, c, s : in bit  ;
           d : in bit_vector (7 downto 0) ;
           y, w : out bit ) ;
end EC151 ;
```

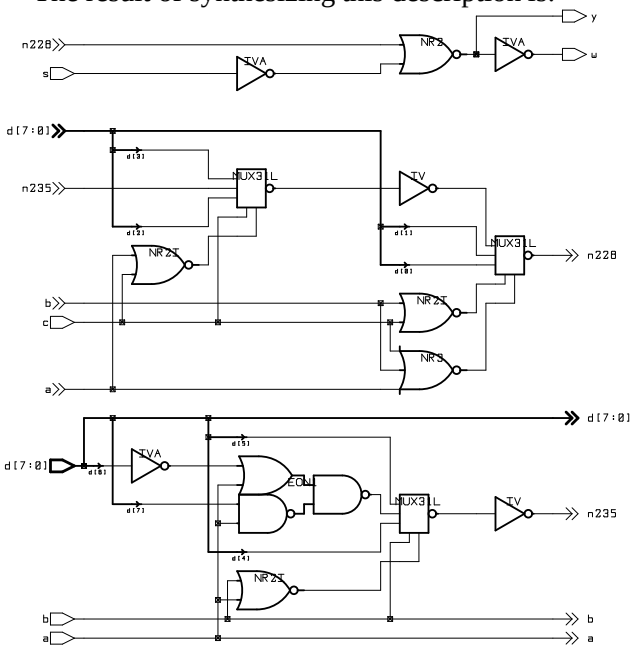
A straighforward architecture written with if/then/else statements might look like:

```
architecture rtl of EC151 is
begin

   process(a,b,c,s,d)
   begin
      if s = ’0’ then
         y <= ’0’ ;
         w <= ’1’ ;
      else
         if    c=’0’ and b=’0’ and a=’0’ then
            y <= d(0) ;
            w <= not d(0) ;
         elsif c=’0’ and b=’0’ and a=’1’ then
            y <= d(1) ;
            w <= not d(1) ;
         elsif c=’0’ and b=’1’ and a=’0’ then
            y <= d(2) ;
            w <= not d(2) ;
         elsif c=’0’ and b=’1’ and a=’1’ then
            y <= d(3) ;
            w <= not d(3) ;
         elsif c=’1’ and b=’0’ and a=’0’ then
            y <= d(4) ;
            w <= not d(4) ;
         elsif c=’1’ and b=’0’ and a=’1’ then
            y <= d(5) ;
            w <= not d(5) ;
         elsif c=’1’ and b=’1’ and a=’0’ then
            y <= d(6) ;
            w <= not d(6) ;
         else
            y <= d(7) ;
            w <= not d(7) ;
         end if ;
      end if ;
   end process ;

end rtl ;
```

The result of synthesizing this description is:



An alternative architecture can be written by deriving the sum-of-products expressions for y and w from a truth table containing all the possible combinations of inputs. The description can be simplified by using a separate process to compute w and y from an intermediate internal signal:

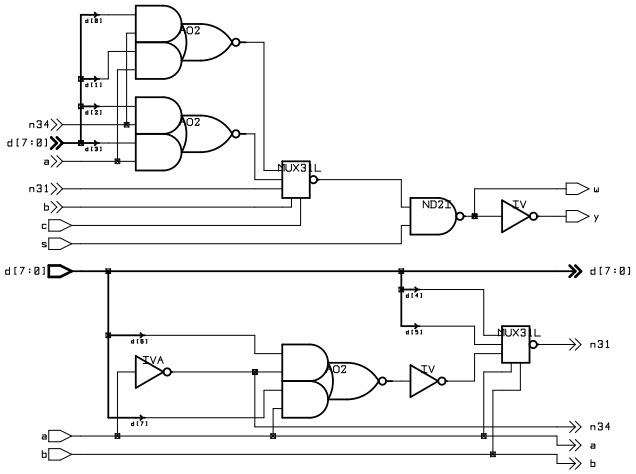
```
architecture rtl of EC151 is
```

```
   process(tmp,s)
   begin
        y <= s and tmp ;
        w <= not ( s and tmp ) ;
   end process ;

end rtl ;
```

Note that the parentheses are required because the `and` and `or` logical operators have the same precedence.

The result of synthesizing this description is:



A third approach is to combine `a`, `b`, and `c` into an internal 3-element `bit_vector` and use it in a case statement:

```
architecture rtl of EC151 is
   -- internal signals:
   signal tmp : bit ;
   signal sel : bit_vector (2 downto 0) ;
begin
```

```
   end

end rt
```

This
the pre
The
that we
values
verted
portabl
express
rent sig

```
library
use syn

archite
   sign
begin

   tmp
   y <=
   w <=

end rt
```

This
cuit as
back o
other c
portabl
IEEE \