Csc 343 Lab

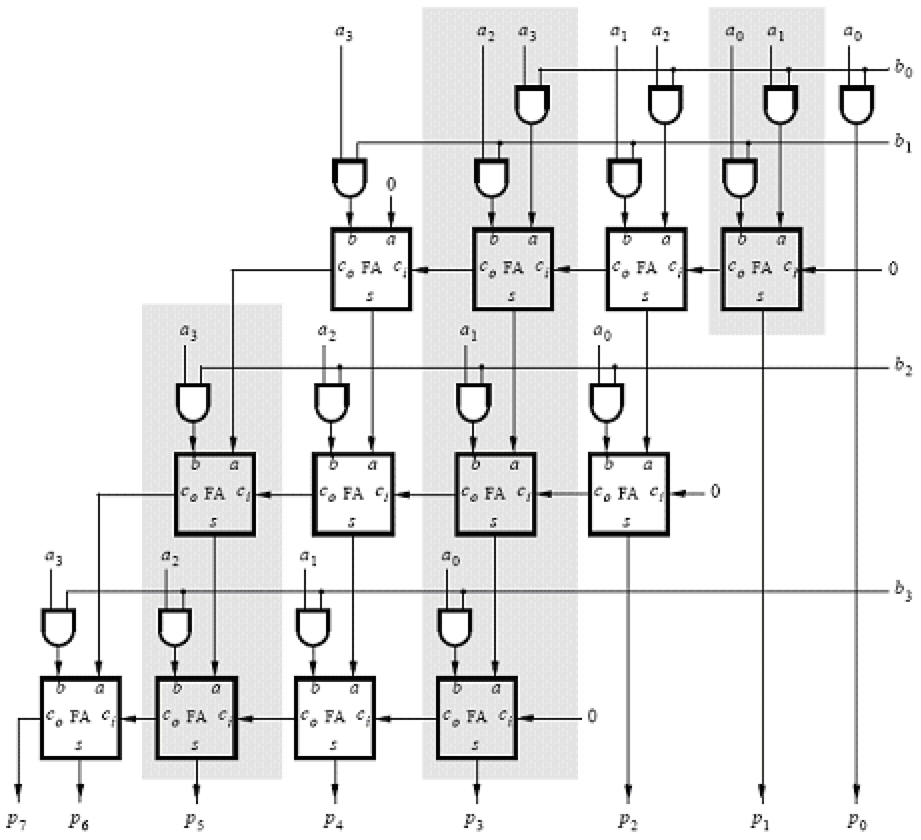
This is a P= A*B multiplier. A and B are 4 bit inputs of the multiplier.
The design is according to the Laboratory Exercise 6 Figure 6.
The components of this multiplier are 12 one-bit adders and 16 AND gates.

one bit adder:

a	b	ci	s	co
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

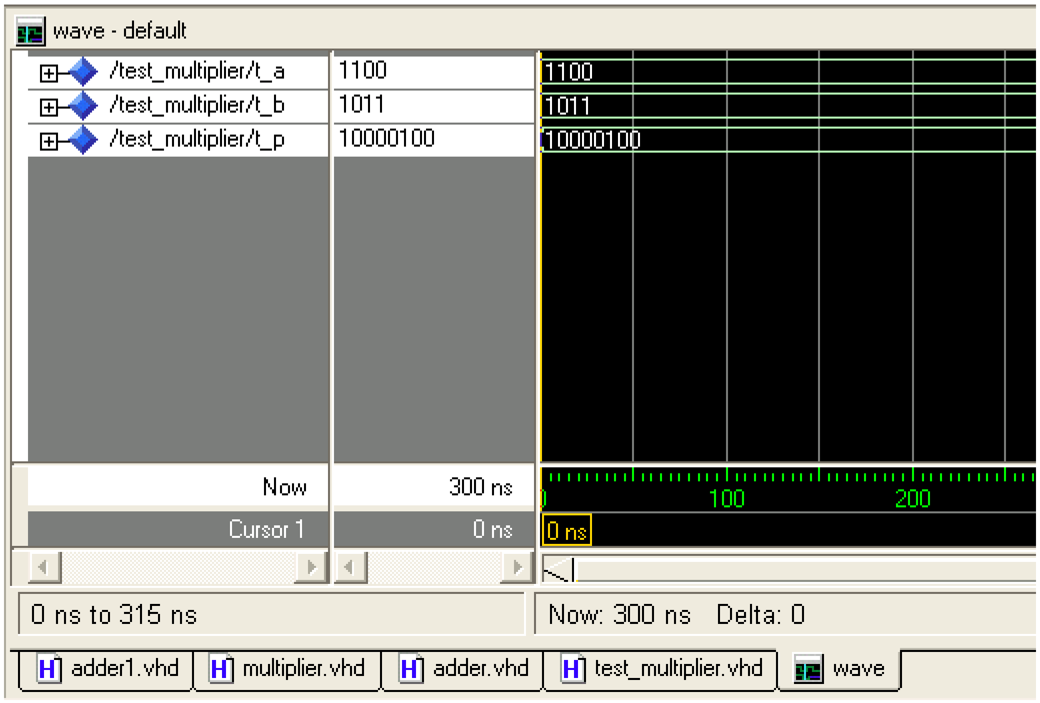
s = a xor b xor ci
co = (a and b) or (a and c) or (b and c)

multiplier:



```
=================multiplier.vhd=======================
library ieee;
use ieee.std_logic_1164.all;
use work.all;

entity multiplier is

port(   A: in std_logic_vector(3 downto 0);
        B: in std_logic_vector(3 downto 0);
        P: out std_logic_vector(7 downto 0)
);

end multiplier;
architecture struct of multiplier is

component adder1 is
port ( a:  in std_logic;
       b:  in std_logic;
       ci: in std_logic;
       co:  out std_logic;
       s:   out std_logic
      );
end component;

component and1 is
port (x:     in std_logic;
      y:     in std_logic;
      f:     out std_logic
      );
end component;

signal ss1: std_logic_vector(7 downto 0);
signal ss2: std_logic_vector(7 downto 0);
signal ss3: std_logic_vector(7 downto 0);
signal ss4: std_logic_vector(3 downto 0);
signal ss5: std_logic_vector(3 downto 0);
signal ss6: std_logic_vector(2 downto 0);


begin

AND_1: and1 port map (x=>B(0), y=>A(0), f=>P(0)   );
AND_2: and1 port map (x=>B(0), y=>A(1), f=>ss1(0) );
AND_3: and1 port map(x=>B(0), y=>A(2), f=>ss1(2) );
AND_4: and1 port map(x=>B(0), y=>A(3), f=>ss1(4) );
```

```
AND_5: and1 port map(x=>B(1), y=>A(0), f=>ss1(1) );
AND_6: and1 port map(x=>B(1), y=>A(1), f=>ss1(3) );
AND_7:  and1 port map(x=>B(1), y=>A(2), f=>ss1(5) );
AND_8:  and1 port map(x=>B(1), y=>A(3), f=>ss1(7) );
AND_9: and1 port map(x=>B(2), y=>A(0), f=>ss2(1) );
AND_10: and1 port map(x=>B(2), y=>A(1), f=>ss2(3) );
AND_11: and1 port map(x=>B(2), y=>A(2), f=>ss2(5) );
AND_12: and1 port map(x=>B(2), y=>A(3), f=>ss2(7) );
AND_13: and1 port map(x=>B(3),  y=>A(0), f=>ss3(1));
AND_14: and1 port map(x=>B(3), y=>A(1), f=>ss3(3));
AND_15: and1 port map(x=>B(3), y=>A(2), f=>ss3(5));
AND_16: and1 port map(x=>B(3), y=>A(3), f=>ss3(7));
ADDER_1: adder1 port map (a=>ss1(0), b=>ss1(1), ci=>'0', s=>P(1), co=>ss4(0));
ADDER_2: adder1 port map(a=>ss1(2), b=>ss1(3), ci=>ss4(0), s=>ss2(0), co=>ss4(1) );
ADDER_3: adder1 port map(a=>ss1(4), b=>ss1(5), ci=>ss4(1), s=>ss2(2), co=>ss4(2) );
ADDER_4: adder1 port map(a=>'0', b=>ss1(7), ci=>ss4(2), s=>ss2(4), co=>ss4(3) );
ADDER_5: adder1 port map(a=>ss2(0), b=>ss2(1), ci=>'0', s=>P(2), co=>ss5(0) );
ADDER_6: adder1 port map(a=>ss2(2), b=>ss2(3), ci=>ss5(0), s=>ss3(0), co=>ss5(1) );
ADDER_7: adder1 port map(a=>ss2(4), b=>ss2(5), ci=>ss5(1), s=>ss3(2), co=>ss5(2));
ADDER_8: adder1 port map(a=>ss4(3), b=>ss2(7), ci=>ss5(2), s=>ss3(4), co=>ss5(3));
ADDER_9:  adder1 port map(a=>ss3(0), b=>ss3(1), ci=>'0', s=>P(3), co=>ss6(0));
ADDER_10:  adder1 port map(a=>ss3(2), b=>ss3(3), ci=>ss6(0), s=>P(4), co=>ss6(1));
ADDER_11: adder1 port map(a=>ss3(4), b=>ss3(5), ci=>ss6(1), s=>P(5), co=>ss6(2));
ADDER_12:  adder1 port map(a=>ss5(3), b=>ss3(7), ci=>ss6(2), s=>P(6), co=>P(7) );
end struct;
```



/test_multiplier/t_a	1100	1100
/test_multiplier/t_b	1011	1011
/test_multiplier/t_p	10000100	10000100
Now	300 ns	
Cursor 1	0 ns	0 ns