Ian Parker

Homework 3

Go through the ModelSim Tutorial. Then, write the VHDL for problems 6.9, 6.46, and 7.10. For each problem, you are to use two of the following coding styles to implement:

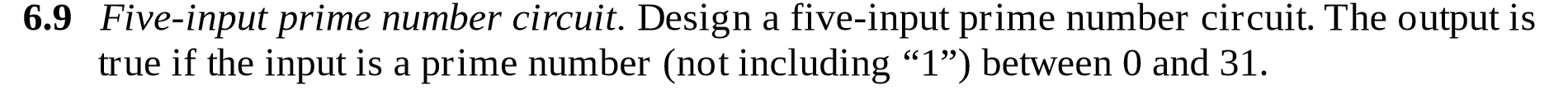
-Dataflow implementation (and/or/not gates)

-Using if-then-else statements

-Using case statements

-Using when-else statements

Between the three problems, you must have used each style at least once to receive full credit. In addition, for each implementation, you must perform several simulations in Modelsim.



**--Using Case Statements**

--Using Case Statements

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity PrimeNumberCircuit1 is

port(number : in bit\_vector(4 downto 0);

f: out bit);

end;

architecture Behavior of PrimeNumberCircuit1 is

begin

process (number)

begin

case number is

when "00000" => f <= '0' ;--0

when "00001"=> f <= '0';

when "00010"=> f <= '1';

when "00011"=> f <= '1';

when "00100"=> f <= '0';

when "00101"=> f <= '1';--5

when "00110"=> f <= '0';

when "00111"=> f <= '1';

when "01000"=> f <= '0';

when "01001"=> f <= '0';

when "01010"=>f <= '0';--10

when "01011"=> f <= '1';

when "01100"=> f <= '0';

when "01101" => f <= '1';

when "01110" => f <= '0';

when "01111" => f <= '0';--15

when "10000" => f <= '0';

when "10001" => f <= '1';

when "10010" => f <= '0';

when "10011" => f <= '1';

when "10100" => f <= '0';--20

when "10101" => f <= '0';

when "10110" => f <= '0';

when "10111" => f <= '1';

when "11000" => f <= '0';

when "11001" => f <= '0';--25

when "11010" => f <= '0';

when "11011" => f <= '0';

when "11100" =>f <= '0';

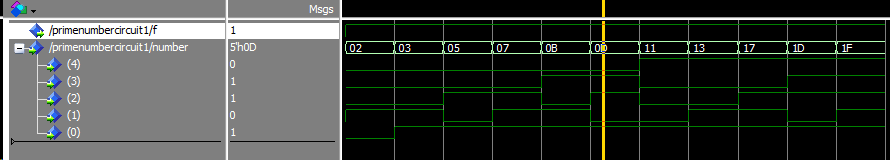
when "11101" => f <= '1';

when "11110" => f <= '0';--30

when "11111" => f <= '1';

end case;

end process;

end Behavior;

**--Using Dataflow Implementation**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity PrimeNumberCircuit2 is

port(A,B,C,D,E : in bit;

F: out bit);

end;

architecture Behavioral of PrimeNumberCircuit2 is

begin

F <= (NOT B AND D AND E)

OR (NOT A AND NOT B AND NOT C AND D)

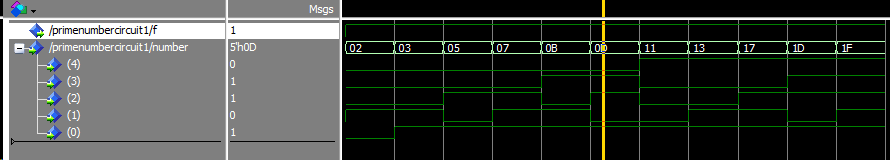
OR (NOT A AND NOT C AND D AND E)

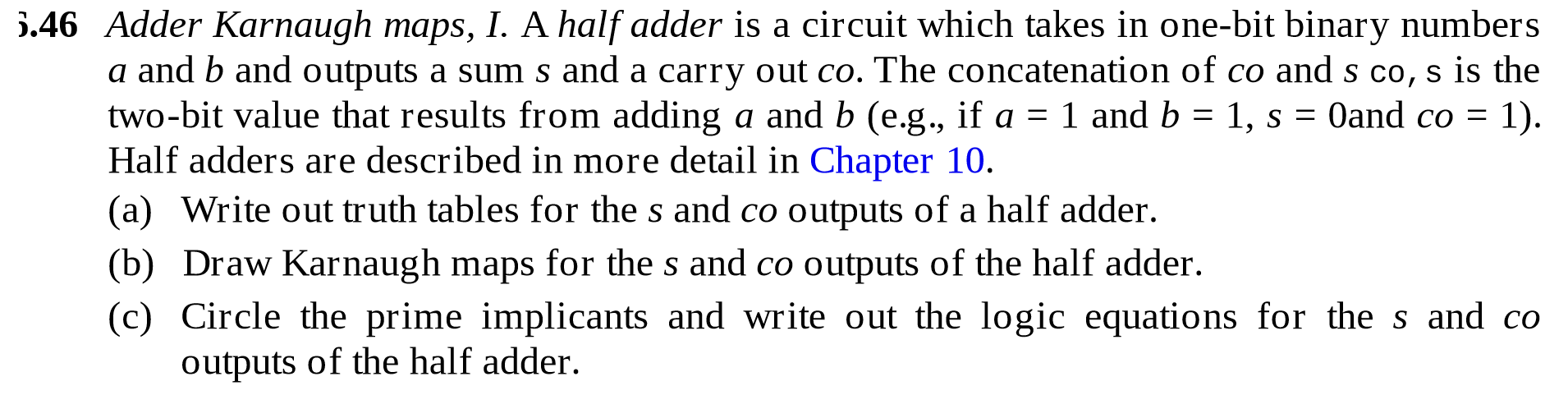
OR (A AND B AND C AND E)

OR (NOT A AND C AND NOT D AND E)

OR (A AND NOT B AND NOT C AND E);

end Behavioral;





**--Using If-Else**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity HalfAdder is

port(A,B : in bit;

SUM, COUT: out bit);

end;

architecture Behavioral of HalfAdder is

begin

process (A,B)

begin

if ((A = '0') and (B = '0')) then

SUM <= '0'; COUT <= '0';

elsif((A = '0') and (B = '1')) then

SUM <= '1'; COUT <= '0';

elsif((A = '1') and (B = '0')) then

SUM <= '1'; COUT <= '0';

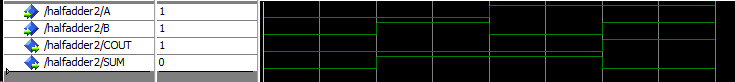
elsif((A = '1') and (B = '1')) then

SUM <= '0'; COUT <= '1';

end if;

end process;

end Behavioral;



**--Using Dataflow Implementation**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity HalfAdder2 is

port(A,B : in bit;

SUM, COUT: out bit);

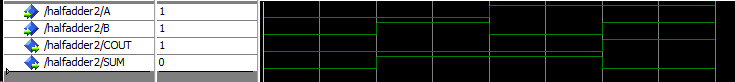
end;

architecture Behavioral of HalfAdder2 is

begin

SUM <= (A XOR B);

COUT<= (A AND B);

End Behavioral; 



**--Using When Else**

--Using When Else

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity Multiples1 is

port( NUMBER : in STD\_LOGIC\_VECTOR(4 downto 0);

F: out STD\_LOGIC);

end;

architecture Behavioral of Multiples1 is

begin

F <= '1' when NUMBER = "00101" else

'1' when NUMBER = "01010" else

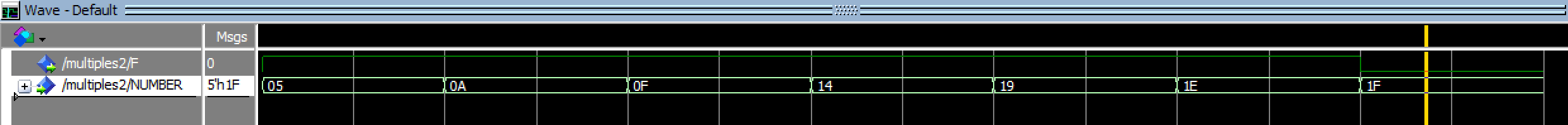
'1' when NUMBER = "01111" else

'1' when NUMBER = "10100" else

'1' when NUMBER = "11001" else

'1' when NUMBER = "11110" ;

End Behavioral;



**--Using If-Else**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity Multiples2 is

port( NUMBER : in STD\_LOGIC\_VECTOR(4 downto 0);

F: out STD\_LOGIC);

end;

architecture Behavioral of Multiples2 is

begin

process(NUMBER)

begin

if (NUMBER = "00101") then F <= '1';

elsif (NUMBER = "01010") then F <= '1';

elsif (NUMBER = "01111") then F <= '1';

elsif (NUMBER = "10100") then F <= '1';

elsif (NUMBER = "11001") then F <= '1';

elsif (NUMBER = "11110") then F <= '1';

else F <= '0';

end if;

end process;

End Behavioral;

