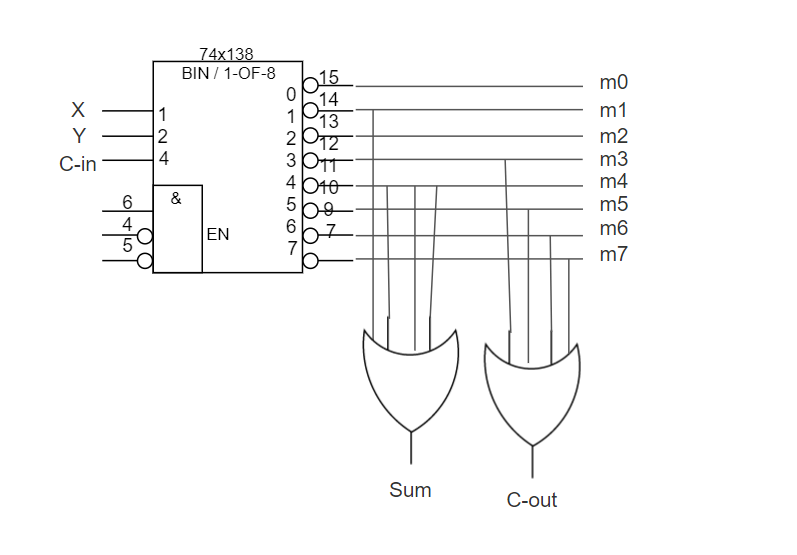
**Question6**

**A**

****

**B**

**Code**

library ieee;

use ieee.std\_logic\_1164.all;

entity full\_adder\_by\_decoder\_3x8 is

port(

inputDecoder : in STD\_LOGIC\_VECTOR(2 downto 0);

sum : out std\_logic;

carryOut : out std\_logic);

end entity;

architecture struct of full\_adder\_by\_decoder\_3x8 is

signal outPutDecoder : STD\_LOGIC\_VECTOR(7 downto 0);

begin

-- to see where m1,m2,m3,... come from, pleas pay attention to answer of section a

outPutDecoder <=("10000000") when (inputDecoder="000") else

("01000000") when (inputDecoder="001") else

("00100000") when (inputDecoder="010") else

("00010000") when (inputDecoder="011") else

("00001000") when (inputDecoder="100") else

("00000100") when (inputDecoder="101") else

("00000010") when (inputDecoder="110") else

("00000001");

process(outPutDecoder) is

variable m0,m1,m2,m3,m4,m5,m6,m7 : STD\_LOGIC;

begin

m0 :='0';

m1 :='0';

m2 :='0';

m3 :='0';

m4 :='0';

m5 :='0';

m6 :='0';

m7 :='0';

case outPutDecoder is

when "10000000" =>

m0 := '1';

when "01000000" =>

m1 := '1';

when "00100000" =>

m2 := '1';

when "00010000" =>

m3 := '1';

when "00001000" =>

m4 := '1';

when "00000100" =>

m5 := '1';

when "00000010" =>

m6 := '1';

when others =>

m7 := '1';

end case;

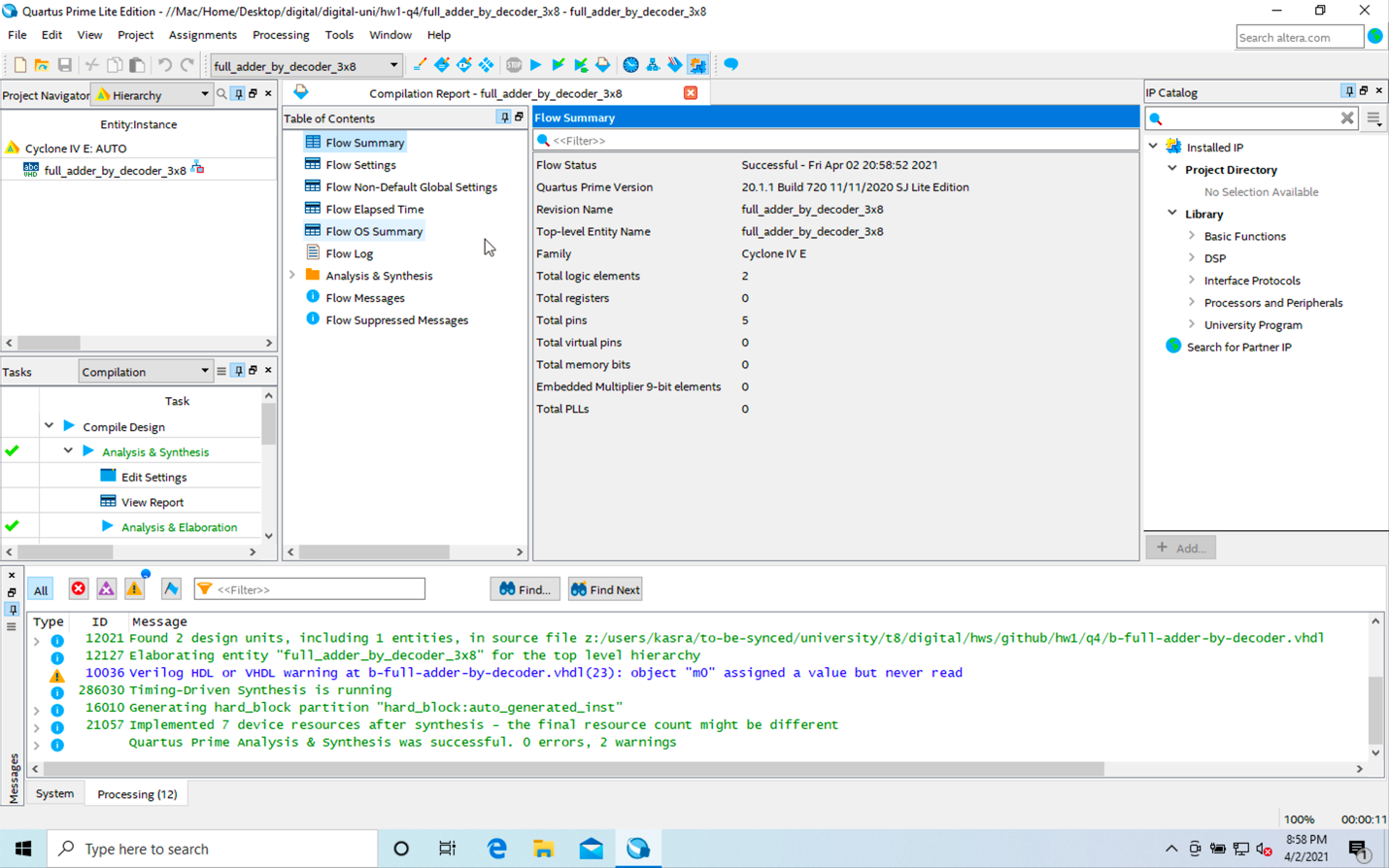
sum <= m1 OR m2 OR m4 OR m7;

carryOut <= m3 OR m5 OR m6 Or m7;

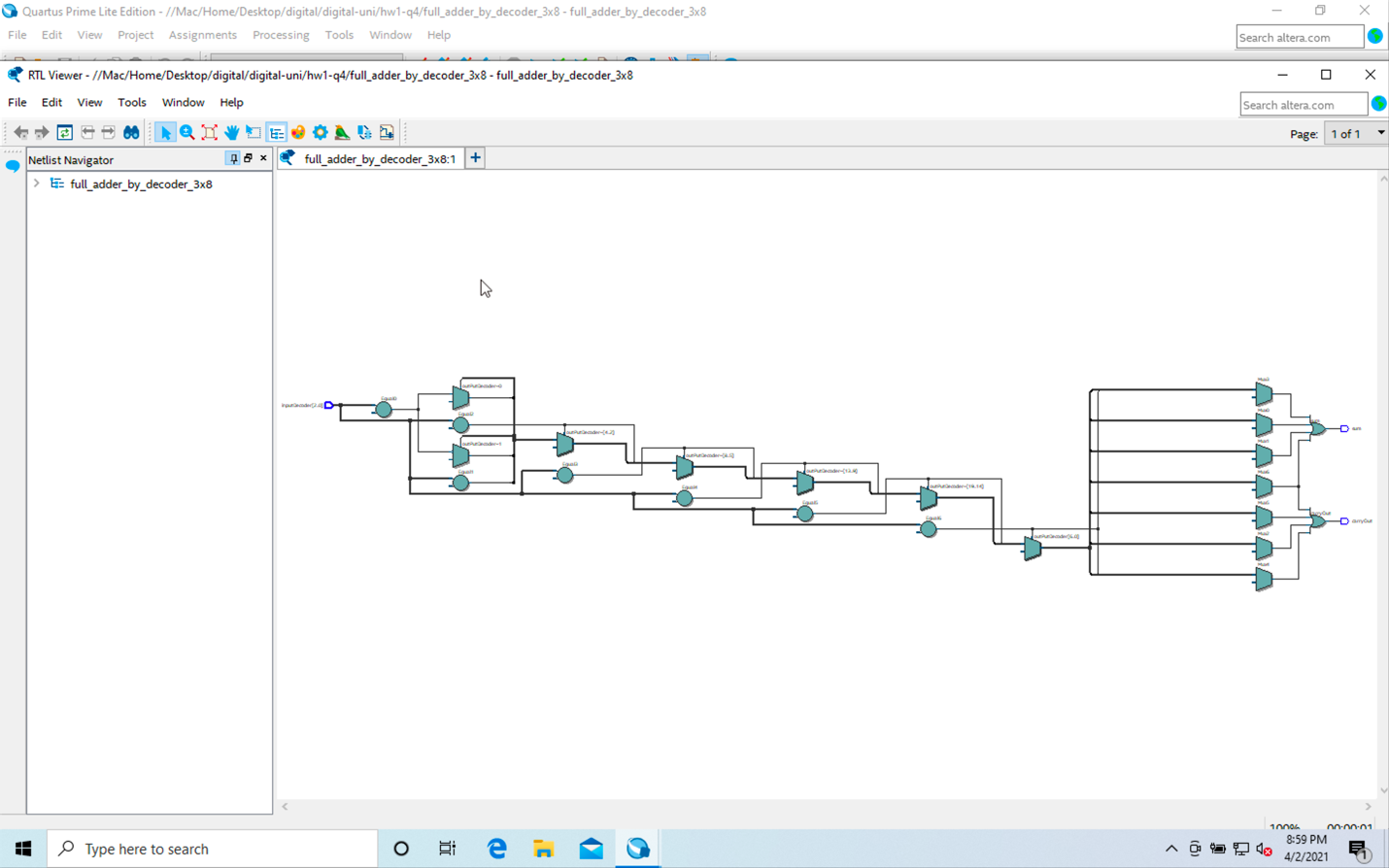
end process;

end architecture;

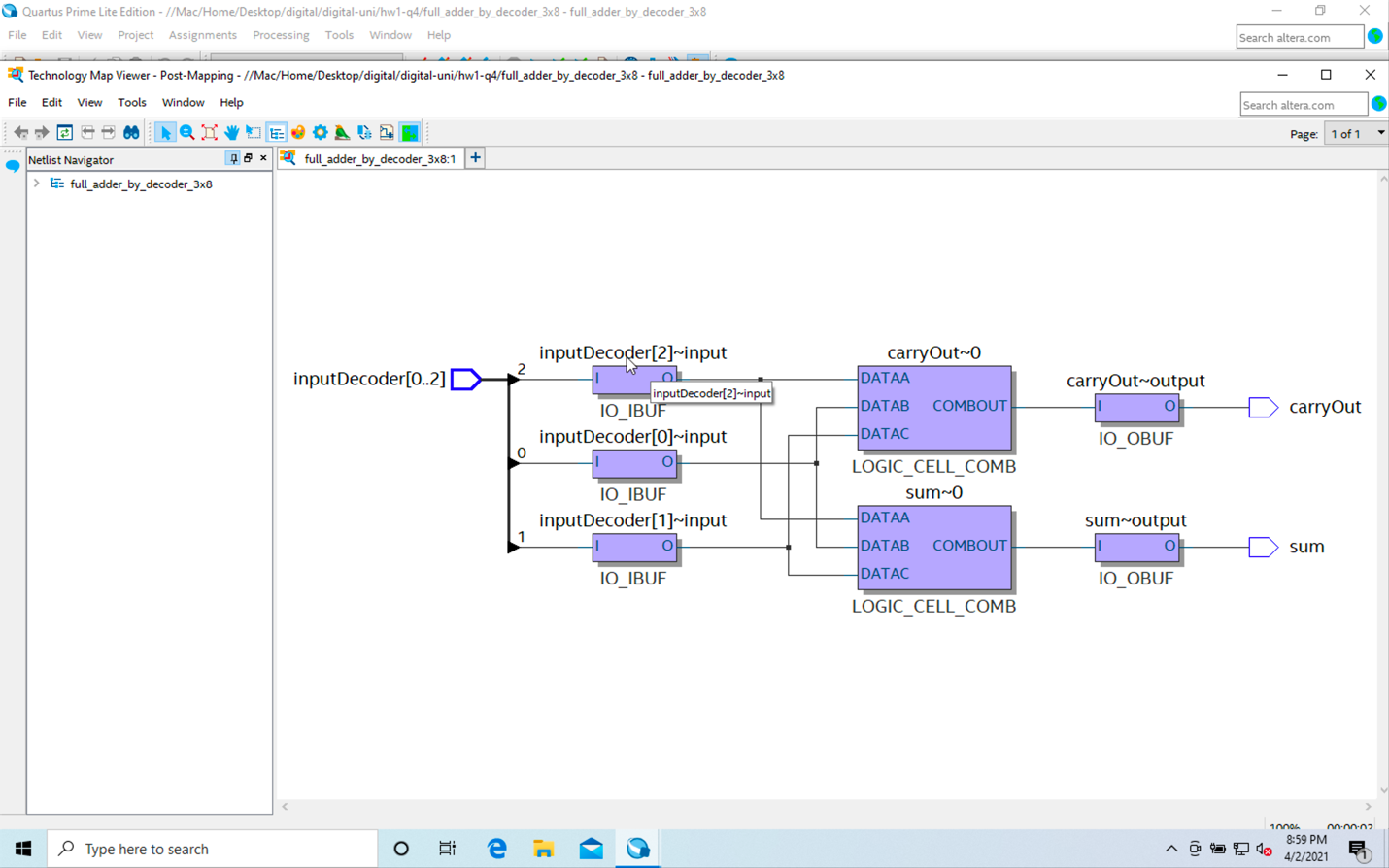
**Compilation report**

****

**RTL**

****

**Post-mapping**

****