Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

1 Introduction

The objective of this final test project is to evaluate student ability to use LPM modules to design and simulate digital computational circuit. This project is based on Intel API as described in section 3 and 4.

2 What to do

2.1 Create SRAM using LPM of 16 words each 32 bits.
2.2 Make sure that you can read and write to this SRAM.
2.3 Create in VHDL add/sub digital circuit using LPM as described in Section 3 and 4. You have to modify the code to handle 32 bit word addition and subtraction.

2.4 The circuit preforms:
2.4.1 Addition of two integers X1+ X2
2.4.2 Subtraction of two integers X1-X2
2.4.3 Cumulative sum X1+x2+x3+x4
2.4.4 Cumulative subtract X1-x2-x3-X4
2.5 To demonstrate in simulation load integers from SRAM to registers A and B to perform the operations 2.4.1, 2.4.2, 2.4.3,2.4.4 You have to specify control signals Sel and AddSub accordingly.
2.6 Demonstrate overflow.
2.7 Simulate all functionality using **MODELSIM.**
2.8 Submit report and presentation

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

3 Adder/Subtractor Using Library of Parameterized Modules in VHDL

The LPMs in the IP Catalog are general in structure and they can be configured to suit a specific application by specifying the values of various parameters. We will use the *lpm_add_sub* module to simplify our adder/subtractor circuit defined in Figures 1 and 2. The augmented circuit is given in Figure 3. The *lpm_add_sub* module, instantiated under the name *megaddsub*, replaces the adder circuit as well as the XOR gates that provide the input *H* to the adder. Since arithmetic overflow is one of the outputs that the LPM provides, it is not necessary to generate this output with a separate XOR gate.

To implement this adder/subtractor circuit, create a new directory named *tutorial_lpm*, and then create a project *addersubtractor2*. Choose the same device as we previously selected (Refer to Table 1) to allow a direct comparison of implemented designs.

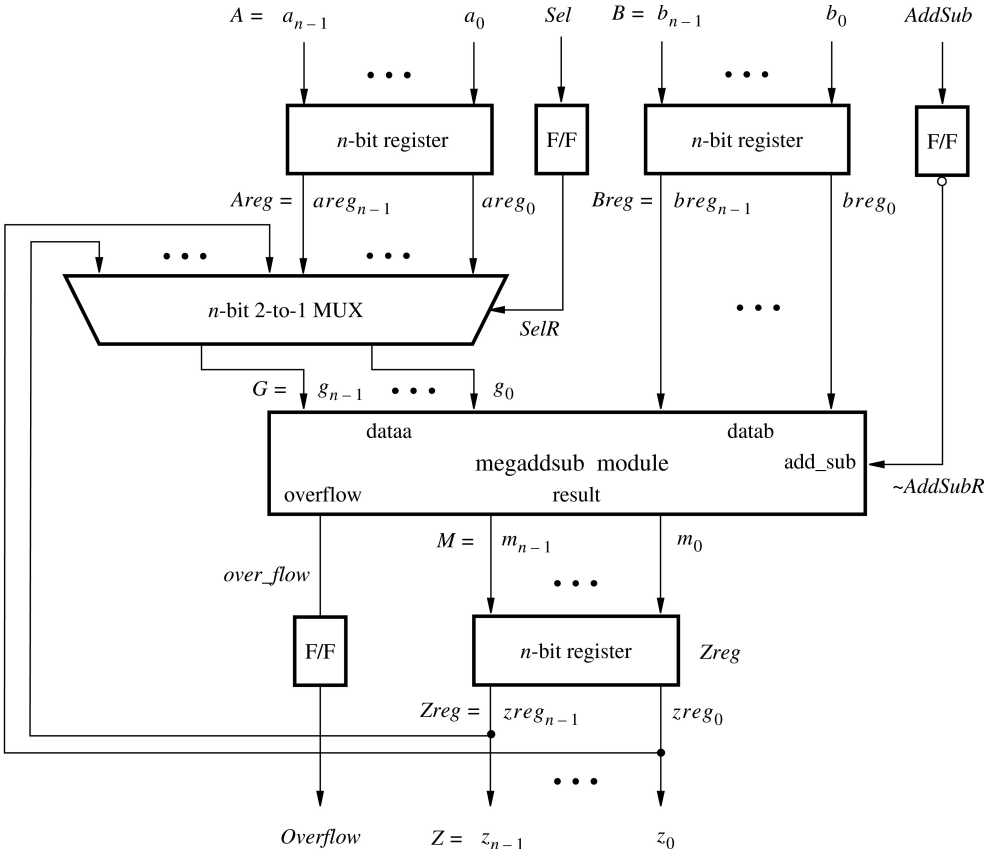


Figure 3. The augmented adder/subtractor circuit.

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

The new design will include the desired LPM subcircuit specified as a VHDL component that will be instantiated in the top-level VHDL design entity. The VHDL component for the LPM subcircuit is generated by using a wizard as follows:

1. Select Tools > IP Catalog, which opens the IP Catalog window in Figure 4.

2. In the IP Catalog panel, expand Library > Basic Functions > Arithmetic and double-click on LPM_ADD_SUB

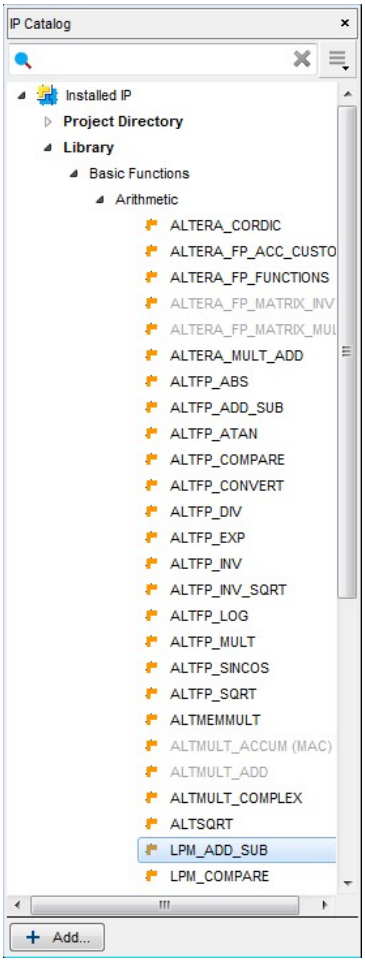


Figure 4. Choose an LPM.

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

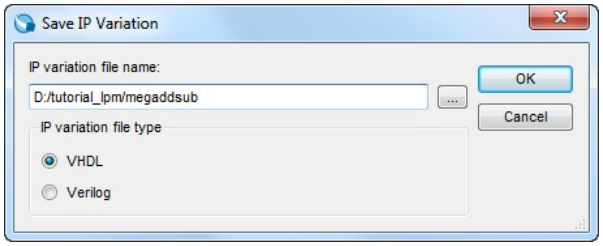
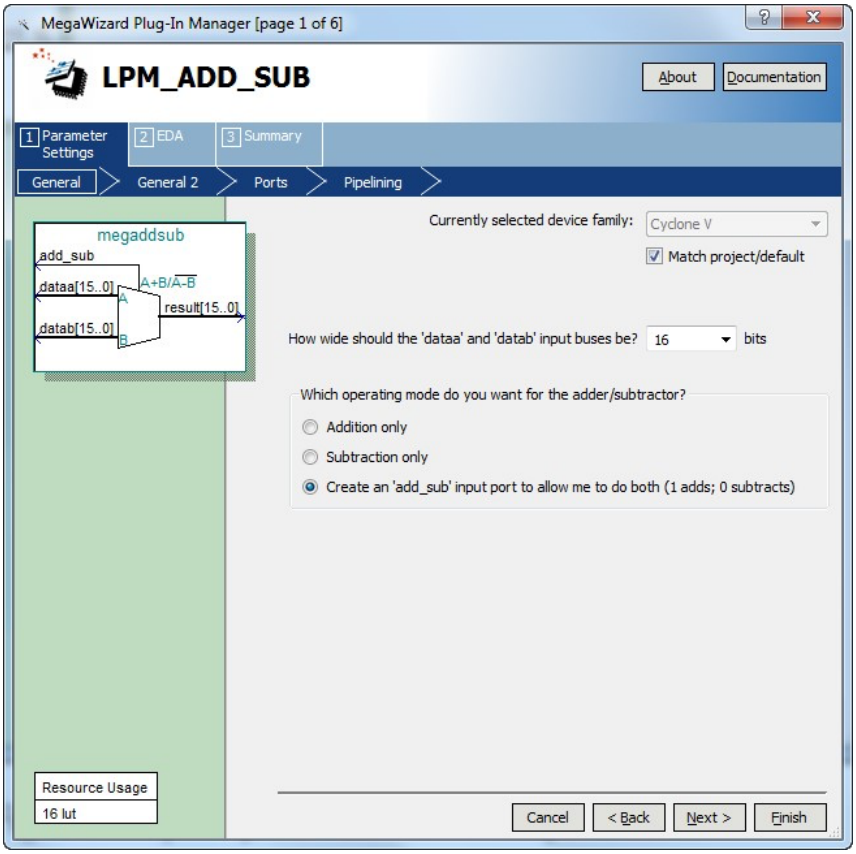


Figure 5. Create an LPM from the available library.

3. In the pop-up box shown in Figure 5, choose VHDL as the type of output file that should be created. The output file must be given a name; choose the name *megaddsub.vhd* and indicate that the file should be placed in the directory *tutorial_lpm* as shown in the figure. Press OK.



Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

Figure 6. Specify the size of data inputs.

4. In the box in Figure 6 specify that the width of the data inputs is 16 bits. Also, specify the operating mode in which one of the ports allows performing both addition and subtraction of the input operand, under the control of the *add_sub* input. A symbol for the resulting LPM is shown in the top left corner. Note that if *add_sub* = 1 then *result* = *A +B*; otherwise, *result* = *A −B*. This interpretation of the control input and the operation performed is different from our original design in Figures 1 and 2, which we have to account for in the modified design. Observe that we have included this change in the circuit in Figure 3. Click Next.

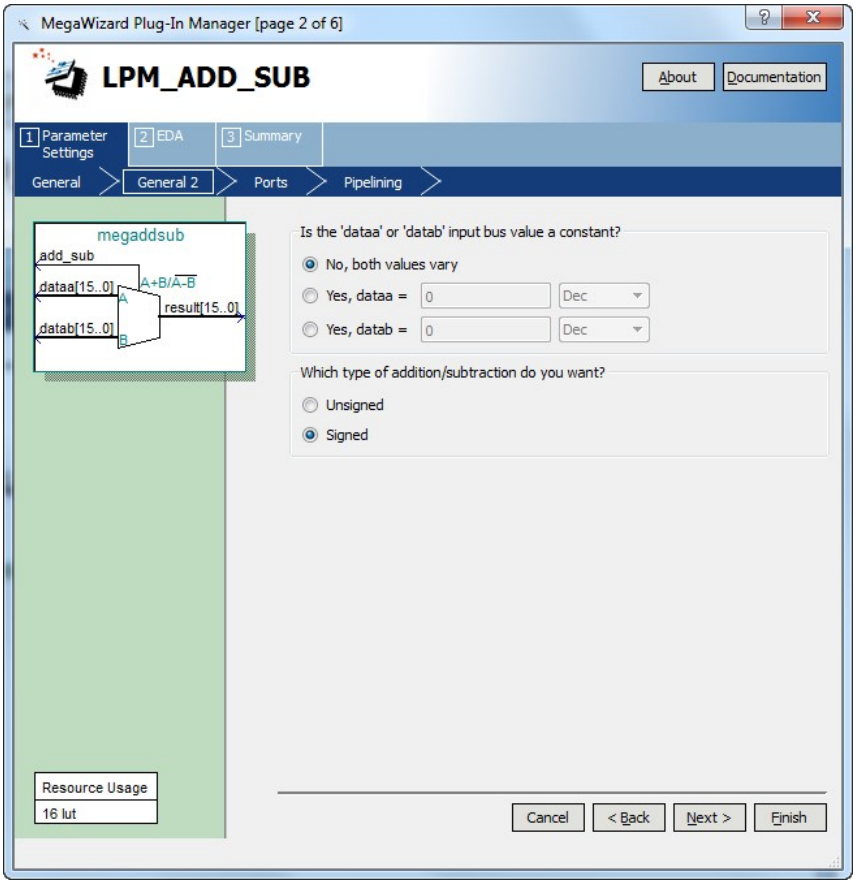


Figure 7. Further specification of inputs.

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

---

5. In the box in Figure 7, specify that the values of both inputs may vary and select Signed for the type of addition/subtraction. Click Next.

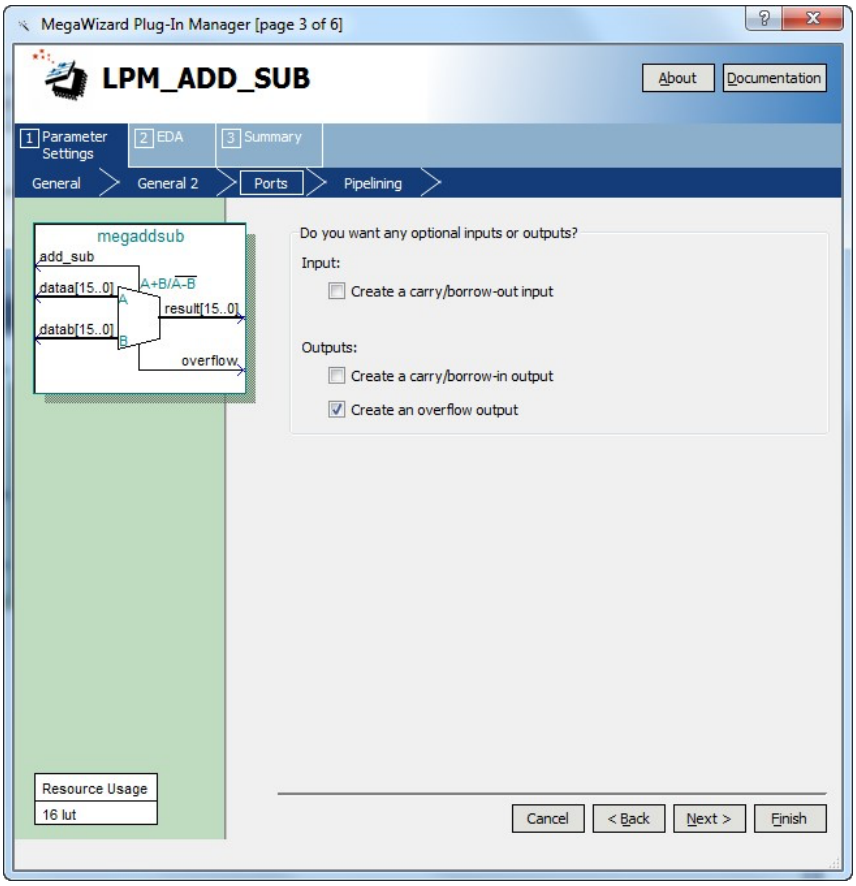


Figure 8. Specify the Overflow output.

6. The box in Figure 8 allows the designer to indicate optional inputs and outputs that may be specified. Since we need the overflow signal, make the Create an overflow output choice and press Next.

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.

The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

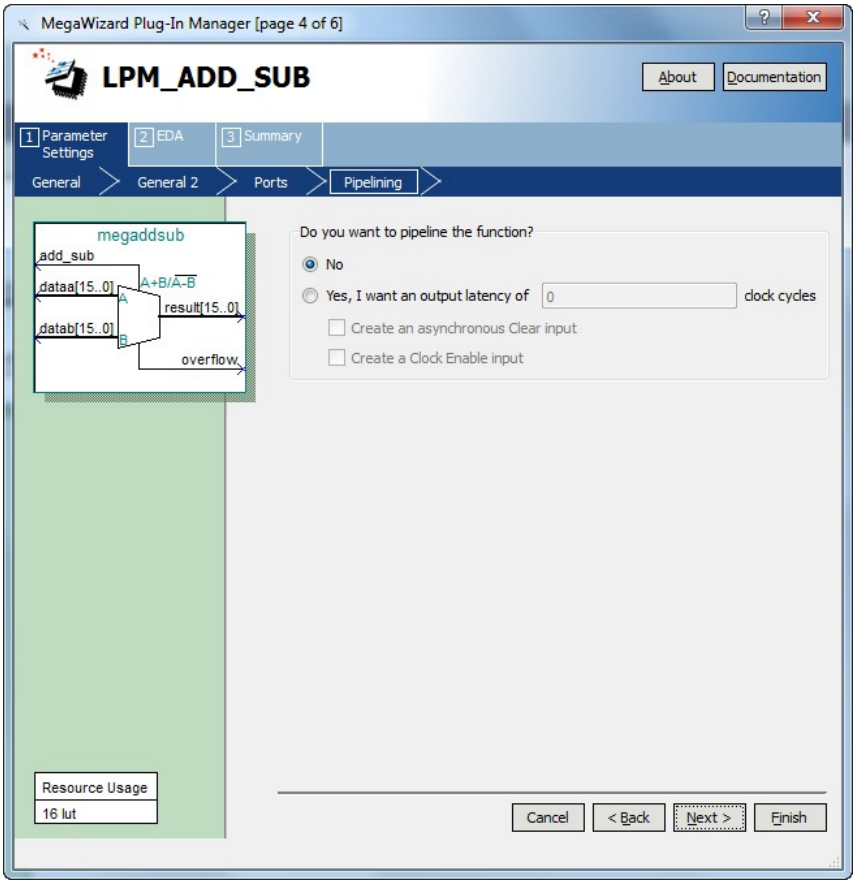


Figure 9. Refuse the pipelining option.

7. In the box in Figure 9 say No to the pipelining option and click Next.
8. Figure 10 shows the simulation model files needed to simulate the generated design. Press Next to proceed to the final page.

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.

The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

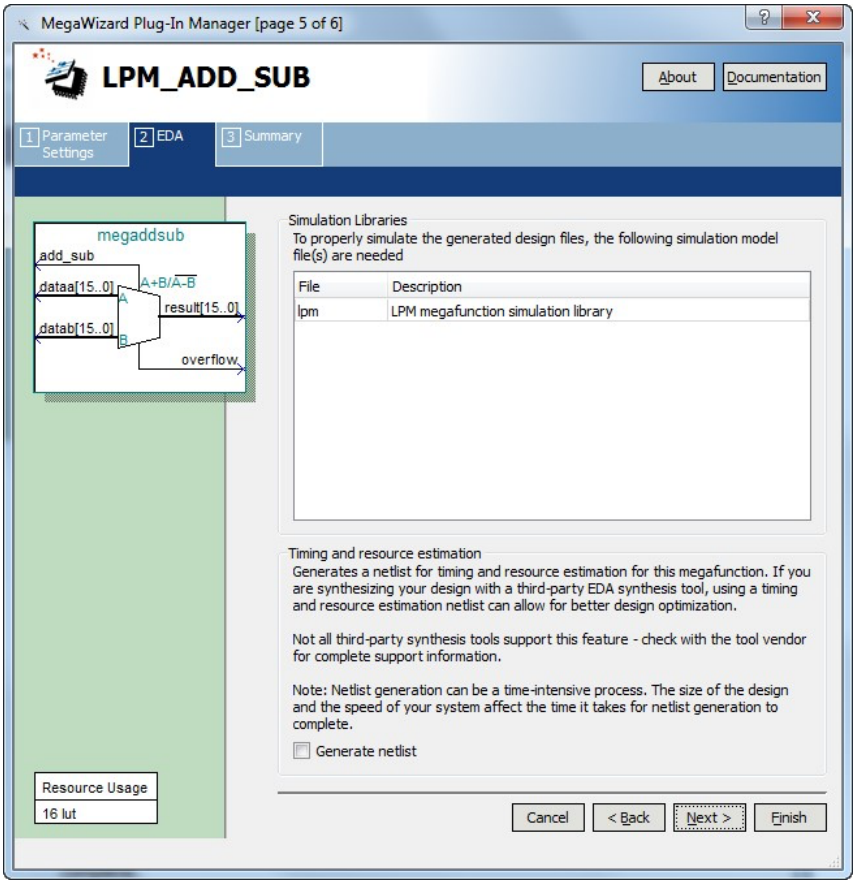


Figure 10. Simulation model files.

9. Figure 11 gives a summary which shows the files that the wizard will create. Press Finish to complete the process.

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.

The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

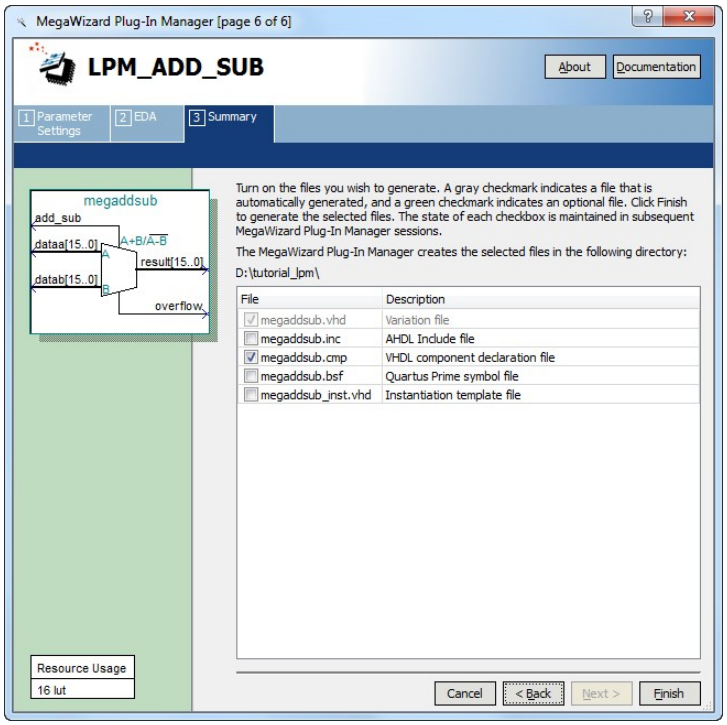


Figure 11. Files created by the wizard.

10. The box in Figure 12 may pop up. If it does, make sure to press No, since adding the newly generated files to the project is not needed when using VHDL (in fact, this may cause compilation errors).

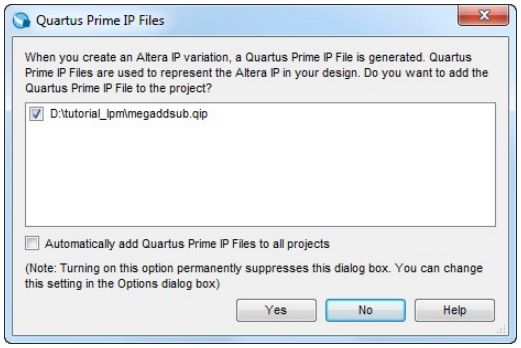


Figure 12. Do not add the new files to the project.

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

4 Augmented Circuit with an LPM

We will use the file *megaddsub.vhd* in our modified design. Figure 13 depicts the VHDL code in this file; note that we have not shown the comments in order to keep the figure small.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY lpm;
USE lpm.all;
ENTITY megaddsub IS
    PORT ( add_sub : IN STD_LOGIC ; dataa : IN STD_LOGIC_VECTOR (15
            DOWNTO 0); datab : IN STD_LOGIC_VECTOR (15 DOWNTO
            0); overflow : OUT STD_LOGIC;
            result      :      OUT STD_LOGIC_VECTOR (15 DOWNTO 0) );
END megaddsub;
ARCHITECTURE SYN OF megaddsub IS
     SIGNAL sub_wire0 :     STD_LOGIC ;
     SIGNAL sub_wire1 :       STD_LOGIC_VECTOR (15 DOWNTO 0);
    COMPONENT lpm_add_sub
    GENERIC   (   lpm_direction   :   STRING;
            lpm_hint        :        STRING;
            lpm_representation  :  STRING;
            lpm_type : STRING; lpm_width
            : NATURAL );
    PORT ( dataa : IN STD_LOGIC_VECTOR (15 DOWNTO 0); add_sub : IN
            STD_LOGIC ; datab : IN STD_LOGIC_VECTOR (15 DOWNTO
            0); overflow : OUT STD_LOGIC ;
            result  :  OUT  STD_LOGIC_VECTOR  (15  DOWNTO  0)  );  END
    COMPONENT;

BEGIN
    overflow <= sub_wire0; result <=
    sub_wire1(15 DOWNTO 0);
    lpm_add_sub_component :
    lpm_add_sub
    GENERIC MAP ( lpm_direction => "UNUSED", lpm_hint =>
            "ONE_INPUT_IS_CONSTANT=NO,CIN_USED=NO",
```

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

```
                lpm_representation => "SIGNED", lpm_type => "LPM_ADD_SUB",
                lpm_width => 16 )
        PORT MAP ( dataa => dataa,
                add_sub => add_sub,
                datab => datab,
                overflow => sub_wire0,
                result => sub_wire1 );
END SYN;
```

Figure 13. VHDL code for the ADD_SUB LPM.

The modified VHDL code for the adder/subtractor design is given in Figure 14. It incorporates the code in Figure 13 as a component. Put this code into a file *addersubtractor2.vhd* under the directory *tutorial_lpm*. The key differences between this code and Figure 2 are:

- The statements that define the *over_flow* signal and the XOR gates (along with the signal H) are no longer needed.
- The *adderk* entity, which specifies the adder circuit, is replaced by *megaddsub* entity. Note that the *dataa* and *datab* inputs shown in Figure 6 are driven by the *G* and *Breg* vectors, respectively.
- *AddSubR* signal is specified to be the inverted version of the *AddSub* signal to conform with the usage of this control signal in the LPM.

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

-- Top-level entity
ENTITY addersubtractor2 IS
        GENERIC ( n :      INTEGER := 16 ) ;
        PORT (A, B                          :      IN STD_LOGIC_VECTOR(n-1 DOWNTO 0) ;
                Clock, Reset, Sel, AddSub   :    IN STD_LOGIC ;
                Z                           :      BUFFER STD_LOGIC_VECTOR(n-1 DOWNTO 0) ;
                Overflow                    :    OUT STD_LOGIC ) ;
END addersubtractor2 ;

ARCHITECTURE Behavior OF addersubtractor2 IS
        SIGNAL G, M, Areg, Breg, Zreg :        STD_LOGIC_VECTOR(n-1 DOWNTO 0) ;
        SIGNAL SelR, AddSubR, over_flow :      STD_LOGIC ;
      COMPONENT mux2to1
```

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

```
        GENERIC ( k :      INTEGER := 8 ) ;
        PORT ( V, W    :  IN        STD_LOGIC_VECTOR(k-1 DOWNTO 0) ;
               Selm    :    IN STD_LOGIC ;
               F       :      OUT STD_LOGIC_VECTOR(k-1 DOWNTO 0) ) ;
    END COMPONENT ;
    COMPONENT megaddsub
        PORT (add_sub : IN STD_LOGIC ; dataa, datab : IN STD_LOGIC_VECTOR(15
    DOWNTO 0) ; result : OUT STD_LOGIC_VECTOR(15 DOWNTO 0) ; overflow :
    OUT STD_LOGIC ) ; END COMPONENT ;
BEGIN
-- Define flip-flops and registers

... continued in Part b
```

Figure 14. VHDL code for the circuit in Figure 3 (Part *a*)

```
    PROCESS ( Reset, Clock )
    BEGIN
        IF Reset = '1' THEN
            Areg <= (OTHERS => '0'); Breg <= (OTHERS => '0');
            Zreg <= (OTHERS => '0'); SelR <= '0'; AddSubR <= '0'; Overflow <= '0';
        ELSIF Clock'EVENT AND Clock = '1' THEN
            Areg <= A; Breg <= B; Zreg <= M;
            SelR <= Sel; AddSubR <= NOT AddSub; Overflow <= over_flow; END IF ;
    END PROCESS ;
--    Define    combinational    circuit
    nbit_addsub: megaddsub
        PORT MAP ( AddSubR, G, Breg, M, over_flow ) ; multiplexer:
    mux2to1
        GENERIC MAP ( k => n )
        PORT MAP ( Areg, Z, SelR, G ) ;
    Z <= Zreg ;
END Behavior;
-- k-bit 2-to-1 multiplexer
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY mux2to1 IS
        GENERIC ( k :      INTEGER := 8 ) ;
        PORT ( V, W        : IN STD_LOGIC_VECTOR(k-1 DOWNTO 0) ;
               Selm     : IN STD_LOGIC ;
               F          : OUT STD_LOGIC_VECTOR(k-1 DOWNTO 0) ) ;
```

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

```
END mux2to1 ;
ARCHITECTURE Behavior OF mux2to1 IS
BEGIN
    PROCESS ( V, W, Selm )
    BEGIN
        IF Selm = '0' THEN F
            <= V ;
        ELSE
            F <= W ; END
        IF ;
    END PROCESS ;
END Behavior ;
-- 16-bit adder/subtractor LPM created by the MegaWizard
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY lpm;
USE lpm.lpm_components.all;
ENTITY megaddsub IS
     PORT ( add_sub    :    IN STD_LOGIC ;

... continued in Part c
```

Figure 14. VHDL code for the circuit in Figure 3 (Part *b*).

```
            dataa : IN STD_LOGIC_VECTOR (15 DOWNTO 0); datab : IN
            STD_LOGIC_VECTOR (15 DOWNTO 0); result : OUT
            STD_LOGIC_VECTOR (15 DOWNTO 0); overflow : OUT
            STD_LOGIC );
END megaddsub;
ARCHITECTURE SYN OF megaddsub IS
     SIGNAL sub_wire0 :    STD_LOGIC ;
     SIGNAL sub_wire1 :      STD_LOGIC_VECTOR (15 DOWNTO 0);
    COMPONENT lpm_add_sub
    GENERIC ( lpm_width : NATURAL;
            lpm_direction    :    STRING;
            lpm_type : STRING; lpm_hint :
            STRING );
    PORT ( dataa : IN STD_LOGIC_VECTOR (15 DOWNTO 0); add_sub : IN
            STD_LOGIC ; datab : IN STD_LOGIC_VECTOR (15 DOWNTO
            0); overflow : OUT STD_LOGIC ;
            result : OUT STD_LOGIC_VECTOR (15 DOWNTO 0) ); END
    COMPONENT;
BEGIN
```

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.
The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

```
        overflow <= sub_wire0; result <=
        sub_wire1(15 DOWNTO 0);
        lpm_add_sub_component :
        lpm_add_sub
        GENERIC MAP ( lpm_width => 16, lpm_direction
                => "UNUSED", lpm_type =>
                "LPM_ADD_SUB",
                lpm_hint => "ONE_INPUT_IS_CONSTANT=NO,CIN_USED=NO" )
        PORT MAP ( dataa => dataa,
                add_sub => add_sub,
                datab => datab,
                overflow => sub_wire0,
                result => sub_wire1 );
END SYN;
```

Figure 14. VHDL code for the circuit in Figure 3 (Part *c*).

Ensure *addersubtractor2.vhd* has been included in the project. To do so, select Project > Add/Remove Files in Project to reach the window in Figure 15. If the file *addersubtractor2.vhd* is not already listed as being included in the project, browse for the available files by clicking the button ... to reach the window in Figure 16. Select the file *addersubtractor2.vhd* and click Open, which returns to the window in Figure 15. Click Add to include the file and then click OK. Now, the modified design can be compiled and simulated in the usual way.

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.

The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

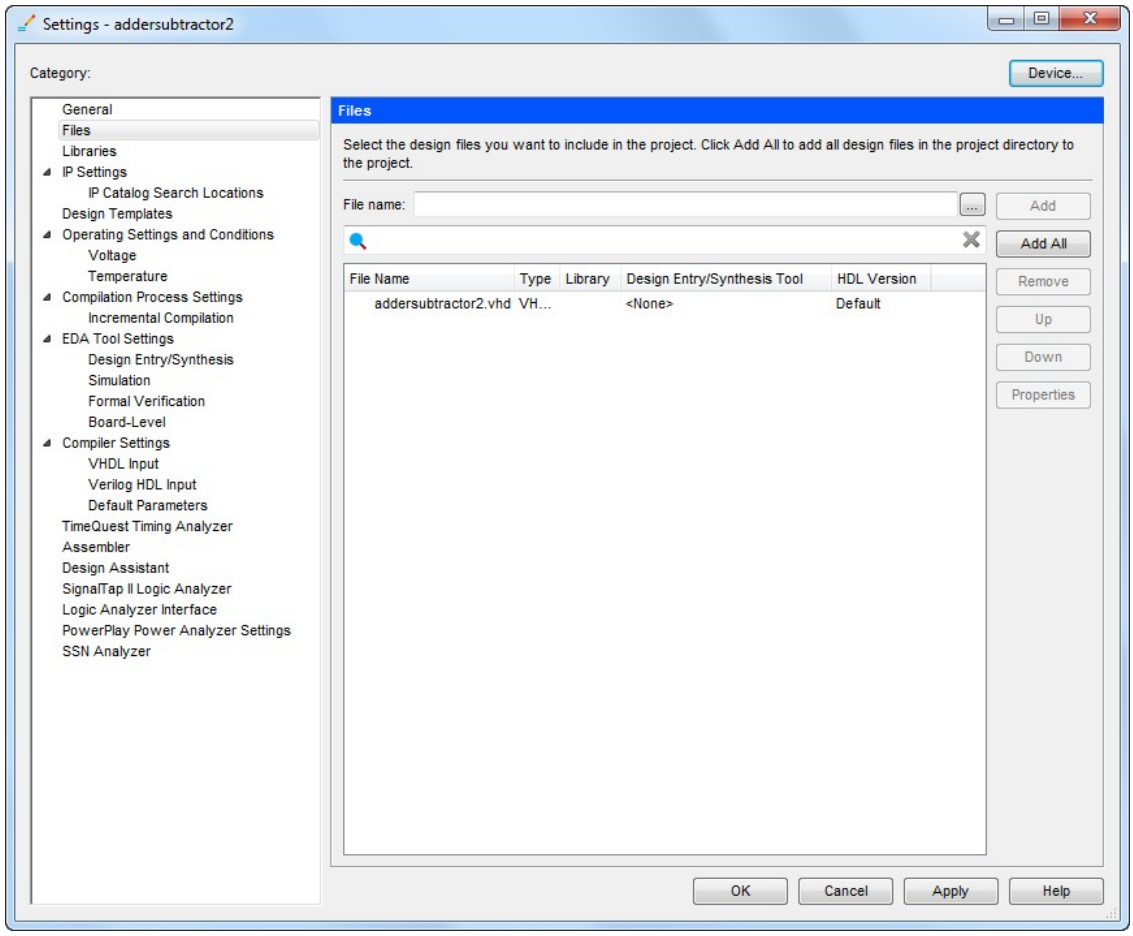


Figure 15. Inclusion of the new file in the project.

Cs 343 Final project test Spring 2023

Please submit to TA your report and presentation by 12:00 PM, May 15, 2023.

The demonstration and presentation will start during class hours 12:30 -1:45 PM in the lab, and will resume at 2:00 PM. Please send to TA your name if you are interested to demo at 12:30 PM. Everyone has to make a presentation and demo.

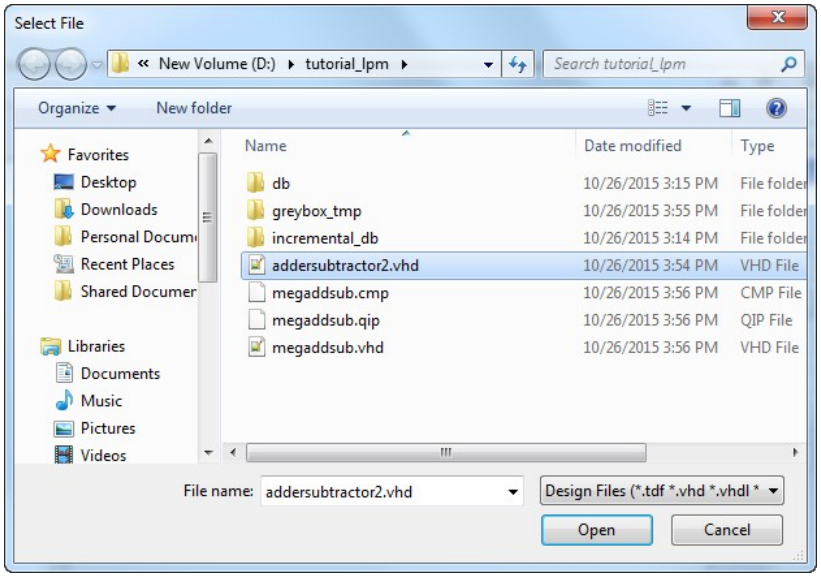


Figure 16. Specify the *addersubtractor2.vhd* file.