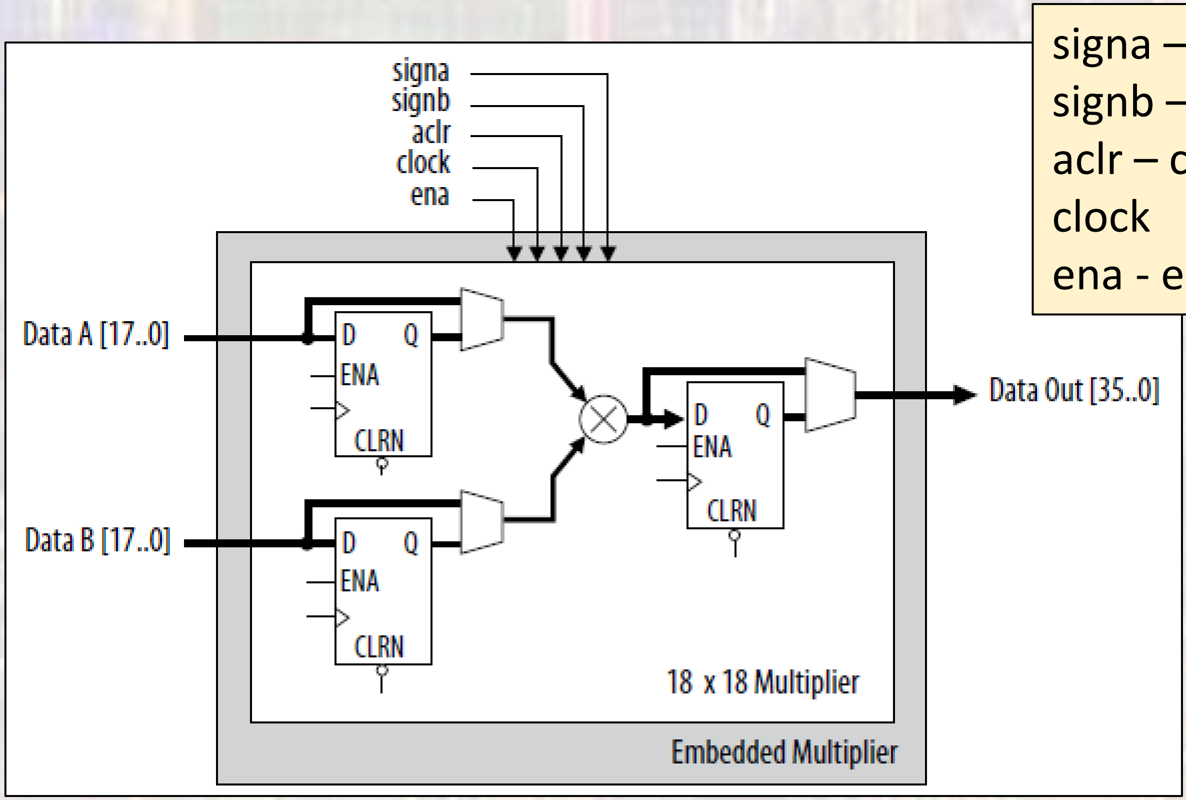
MAX 10 Multiplier

Last updated 7/20/23

MAX10 Multiplier

- MAX10 Fixed Block - Multiplier
  - 144 - 18 x 18 multipliers

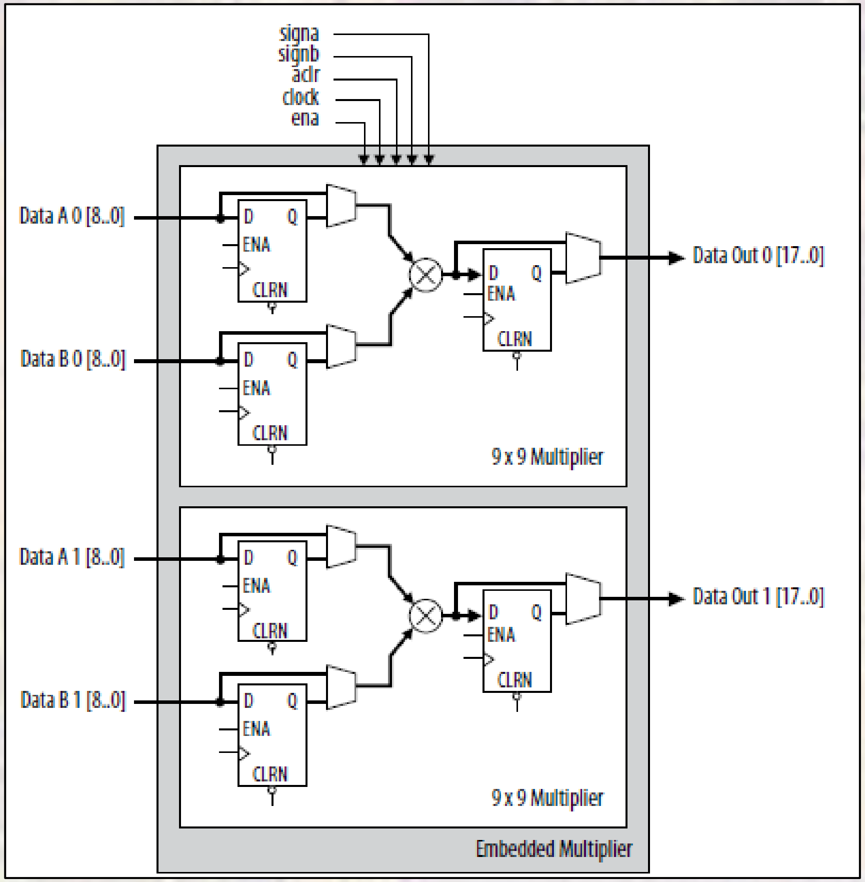


signa – A is signed
signb – B is signed
aclr – clear registers
clock
ena - enable

Src: MAX 10 Device Handbook

MAX10 Multiplier

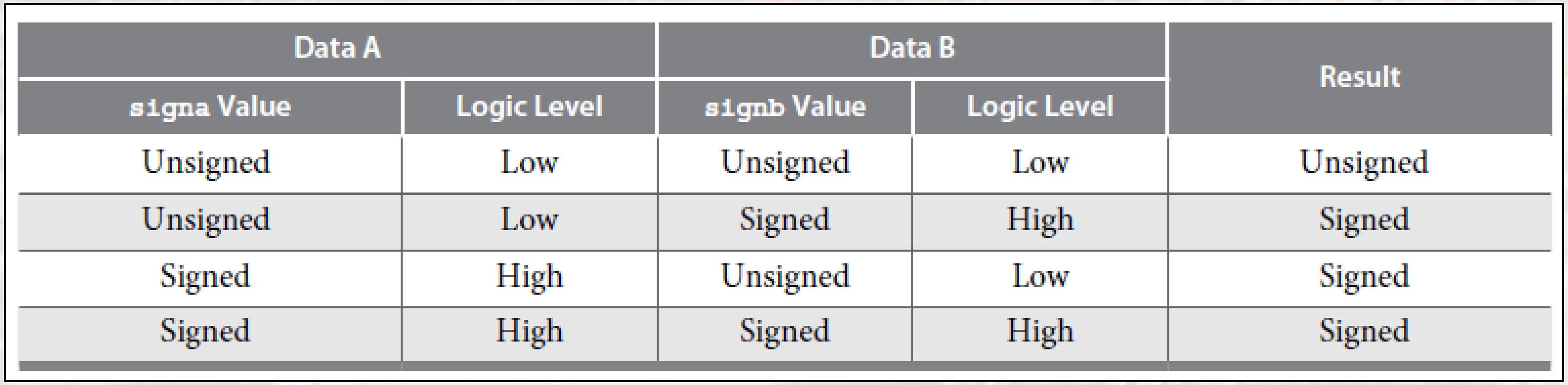
- MAX10 Fixed Block - Multiplier
  - Configured as 2 – 9 x 9 multipliers



Src: MAX 10 Device Handbook

MAX10 Multiplier

- Signals
  - Sign conventions



Data A		Data B		Result
`signa` Value	Logic Level	`signb` Value	Logic Level	
Unsigned	Low	Unsigned	Low	Unsigned
Unsigned	Low	Signed	High	Signed
Signed	High	Unsigned	Low	Signed
Signed	High	Signed	High	Signed

Src: MAX 10 Device Handbook

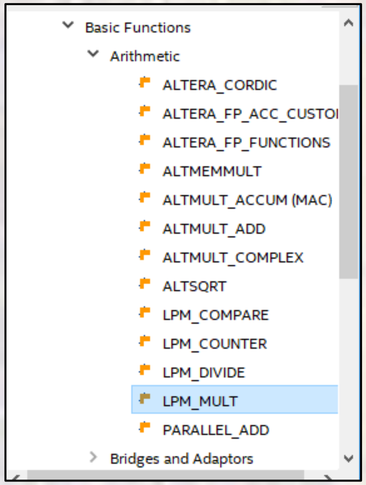
MAX10 Multiplier

- IP Catalog
  - lpm_mult – standard multiplier
  - altmult_add – uses LEs to implement the adders in multiply/add function
  - altmult_accum –uses LEs to implement the accumulator in a multiply/accumulate function
  - altmult_complex – complex multiplier
  - altmemmult – memory based multiplier
    - Part of the memory library

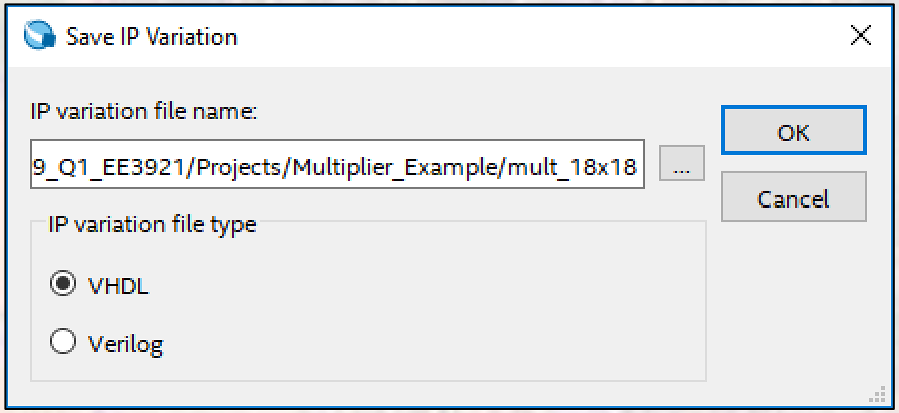
MAX10 Multiplier

- LPM_MULT - configuration
  - Note – no signa or signb in this implementation

    just an option for signed or unsigned multiplcation



- Basic Functions
  - Arithmetic
    - ALTERA_CORDIC
    - ALTERA_FP_ACC_CUSTOI
    - ALTERA_FP_FUNCTIONS
    - ALTMEMMULT
    - ALTMULT_ACCUM (MAC)
    - ALTMULT_ADD
    - ALTMULT_COMPLEX
    - ALTSQRT
    - LPM_COMPARE
    - LPM_COUNTER
    - LPM_DIVIDE
    - LPM_MULT
    - PARALLEL_ADD
  - Bridges and Adaptors



Save IP Variation

IP variation file name: 9_Q1_EE3921/Projects/Multiplier_Example/mult_18x18

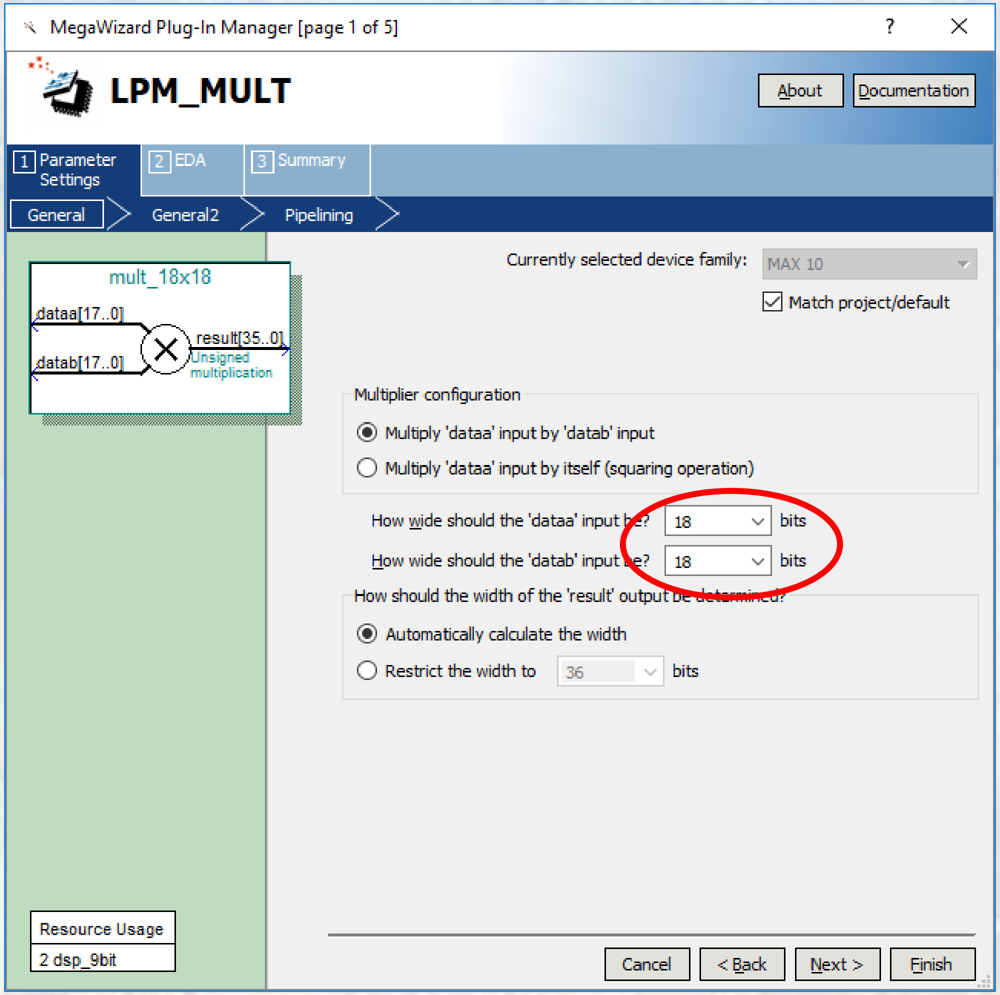
IP variation file type

● VHDL

○ Verilog

MAX10 Multiplier

- LPM_MULT - configuration



MegaWizard Plug-In Manager [page 1 of 5]

LPM_MULT

Resource Usage
2 dsp_9bit

Currently selected device family: MAX 10

☑ Match project/default

Multiplier configuration

● Multiply 'dataa' input by 'datab' input

○ Multiply 'dataa' input by itself (squaring operation)

How wide should the 'dataa' input be? 18 bits

How wide should the 'datab' input be? 18 bits

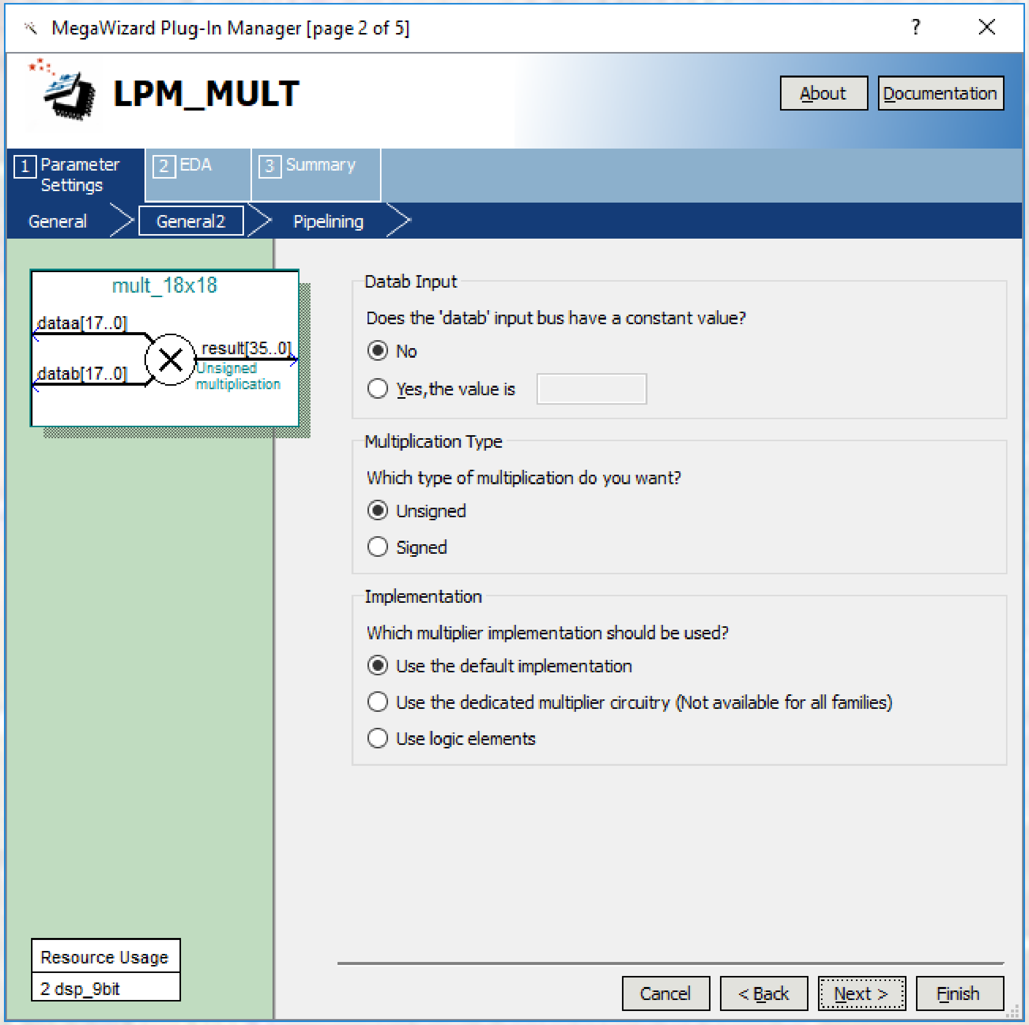
How should the width of the 'result' output be determined?

● Automatically calculate the width

○ Restrict the width to 36 bits

MAX10 Multiplier

- LPM_MULT - configuration



MegaWizard Plug-In Manager [page 2 of 5]

Datab Input

Does the 'datab' input bus have a constant value?

● No

○ Yes, the value is

Multiplication Type

Which type of multiplication do you want?

● Unsigned

○ Signed

Implementation

Which multiplier implementation should be used?

● Use the default implementation

○ Use the dedicated multiplier circuitry (Not available for all families)

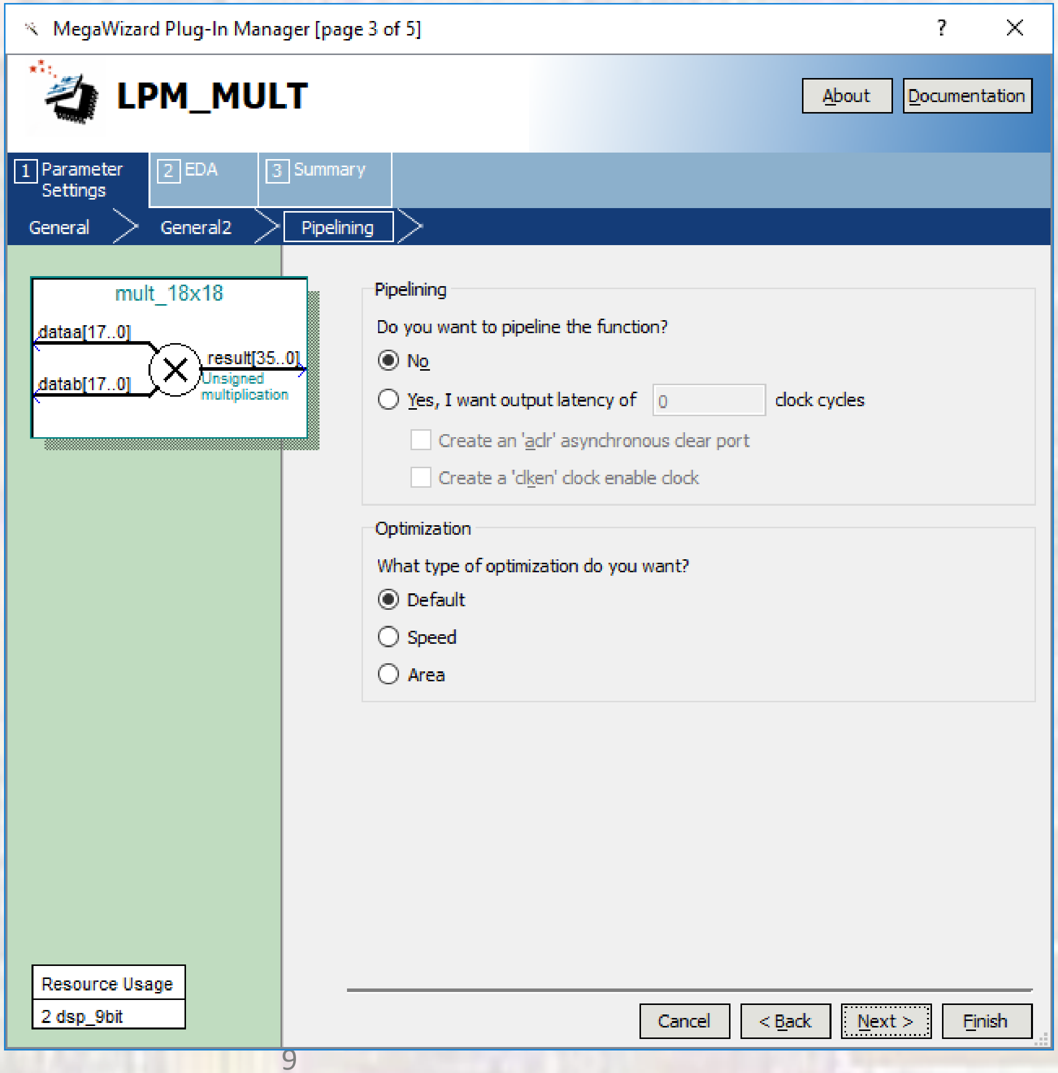
○ Use logic elements

Resource Usage

2 dsp_9bit

MAX10 Multiplier

- LPM_MULT - configuration



MegaWizard Plug-In Manager [page 3 of 5]

LPM_MULT

Pipelining

Do you want to pipeline the function?

● No

○ Yes, I want output latency of 0 clock cycles

□ Create an 'aclr' asynchronous clear port

□ Create a 'clken' clock enable clock

Optimization

What type of optimization do you want?

● Default

○ Speed

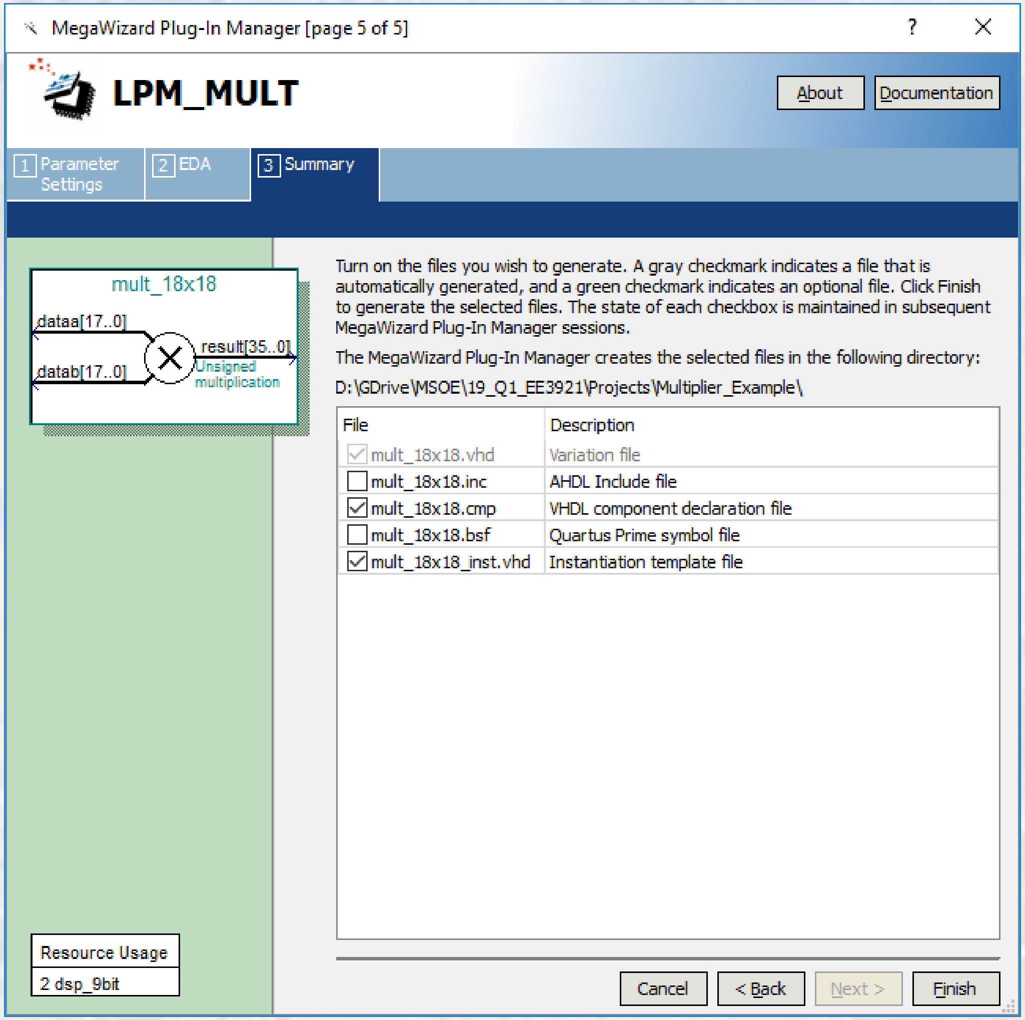
○ Area

Resource Usage

2 dsp_9bit

MAX10 Multiplier

- LPM_MULT - configuration



MegaWizard Plug-In Manager [page 5 of 5]

Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a green checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.

The MegaWizard Plug-In Manager creates the selected files in the following directory:

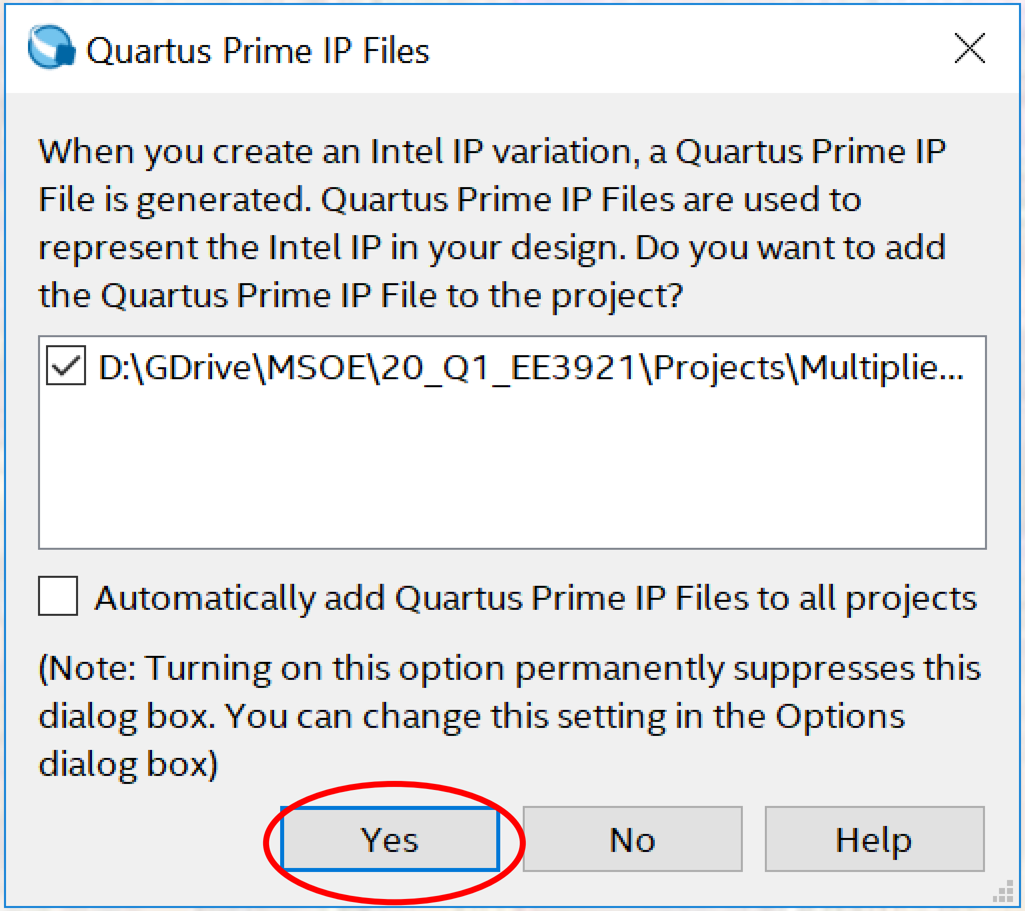
D:\GDrive\MSOE\19_Q1_EE3921\Projects\Multiplier_Example\

File	Description
☑ mult_18x18.vhd	Variation file
□ mult_18x18.inc	AHDL Include file
☑ mult_18x18.cmp	VHDL component declaration file
□ mult_18x18.bsf	Quartus Prime symbol file
☑ mult_18x18_inst.vhd	Instantiation template file

Resource Usage
2 dsp_9bit

MAX10 Multiplier

- LPM_MULT - configuration



Quartus Prime IP Files

When you create an Intel IP variation, a Quartus Prime IP File is generated. Quartus Prime IP Files are used to represent the Intel IP in your design. Do you want to add the Quartus Prime IP File to the project?

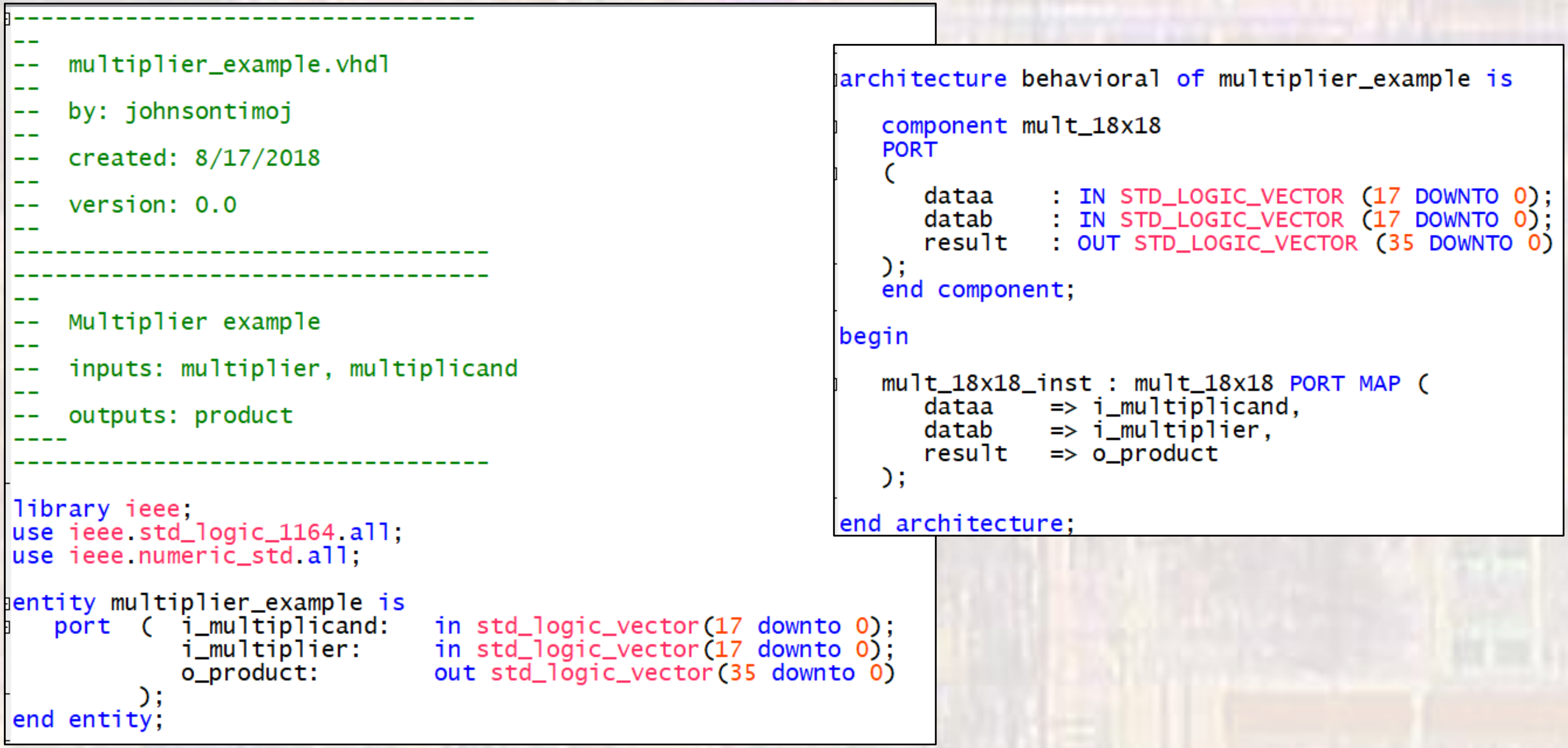
☑ D:\GDrive\MSOE\20_Q1_EE3921\Projects\Multiplie...

□ Automatically add Quartus Prime IP Files to all projects

(Note: Turning on this option permanently suppresses this dialog box. You can change this setting in the Options dialog box)

MAX10 Multiplier

- LPM_MULT - example

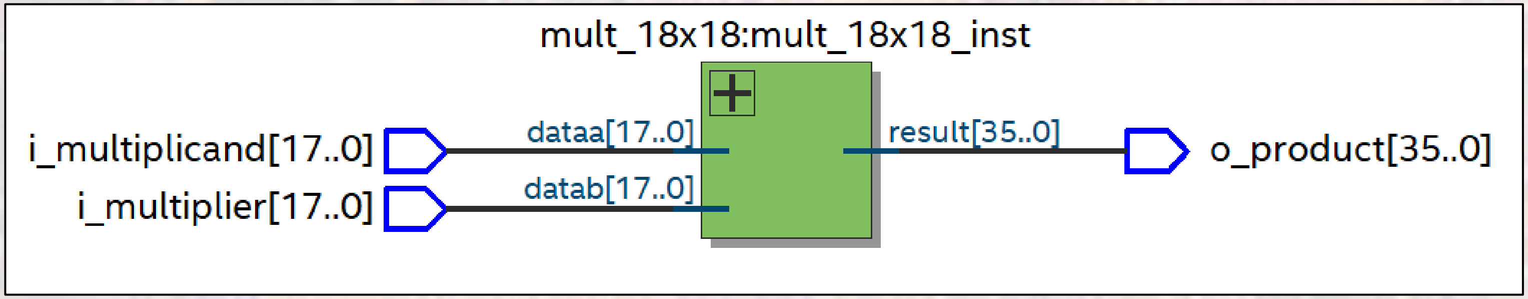


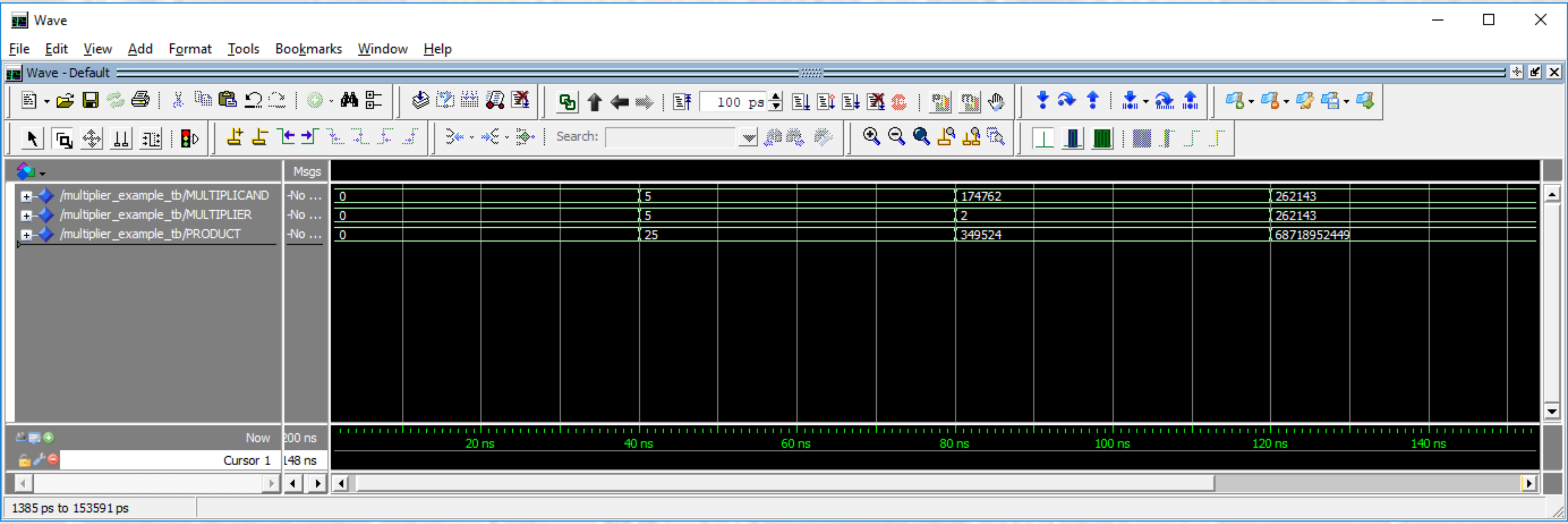
```
---------------------------------
--
--   multiplier_example.vhdl
--
--   by: johnsontimoj
--
--   created: 8/17/2018
--
--   version: 0.0
--
----------------------------------
----------------------------------
--
--   Multiplier example
--
--   inputs: multiplier, multiplicand
--
--   outputs: product
----
----------------------------------

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity multiplier_example is
   port   (  i_multiplicand:    in std_logic_vector(17 downto 0);
             i_multiplier:      in std_logic_vector(17 downto 0);
             o_product:         out std_logic_vector(35 downto 0)
          );
end entity;
```

```
architecture behavioral of multiplier_example is

   component mult_18x18
   PORT
   (
      dataa     : IN STD_LOGIC_VECTOR (17 DOWNTO 0);
      datab     : IN STD_LOGIC_VECTOR (17 DOWNTO 0);
      result    : OUT STD_LOGIC_VECTOR (35 DOWNTO 0)
   );
   end component;

begin

   mult_18x18_inst : mult_18x18 PORT MAP (
      dataa     => i_multiplicand,
      datab     => i_multiplier,
      result    => o_product
   );

end architecture;
```

MAX10 Multiplier

- LPM_MULT - example





```
/multiplier_example_tb/MULTIPLICAND   -No ...   0   5    174762   262143
/multiplier_example_tb/MULTIPLIER     -No ...   0   5    2        262143
/multiplier_example_tb/PRODUCT        -No ...   0   25   349524   68718952449

Now       200 ns
Cursor 1  148 ns

1385 ps to 153591 ps
```

MAX10 Multiplier

- Soft Multipliers
  - Created by HDL code