Lab 04[[1]](#footnote-1)

Timing Closure, PLL, and DPR

# Purpose

The objectives of this lab are as follows:

* How to make Synopsys Design Constraint
* How to attain timing closure using pipelining
* How to instantiate and simulate a PLL
  + How to use the “lock” output signal of the PLL
  + How to constraint derived clock of the PLL
* How to instantiate and simulate dual-port RAM (DPR)

# Required tools

Altera Quartus Prime Standard Edition.

# Required Documents

[1] Quartus Prime Introduction Using Verilog Designs, Intel FPGA, February 2017.

[2] DE0-CV User Manual, Terasic, May 4, 2015.

# Examples

1. **Synopsys Design Constraint (.SDC) file**

The Synopsys Design Constraint (SDC) format provides a simple and easy method to constrain the clocks in your design. If you don’t create it, a default file will be created with constrain on the clock equal 1 GHz.

* Start a new project by following the steps on pages 6 – 12 of document [1]. Select the right FPGA for the DE0-CV board from “Table 1. DE-series FPGA device names” on page 10 in document [1].
* Add the following source files using the steps on pages 16 – 18 in document [1].
  + - 1. multiplier.vhd
      2. binaryToBCD.vhd
      3. SSD.vhd
      4. shift\_add\_3.vhd
      5. multiplier\_no\_sdc.vhd
         1. This file is top level which integrate multiplier (the same as in hw1), but register the inputs and outputs
* In the Files view, right click on multiplier\_no\_sdc.vhd file and select “Set as Top-Level Entity”.
* Open multiplier\_no\_sdc.qsf and add the following pin assignment lines to it

set\_location\_assignment PIN\_U13 -to a[0]

set\_location\_assignment PIN\_V13 -to a[1]

set\_location\_assignment PIN\_T13 -to a[2]

set\_location\_assignment PIN\_T12 -to a[3]

set\_location\_assignment PIN\_AA14 -to b[0]

set\_location\_assignment PIN\_AA13 -to b[1]

set\_location\_assignment PIN\_AB13 -to b[2]

set\_location\_assignment PIN\_AB12 -to b[3]

set\_location\_assignment PIN\_U21 -to ss3[0]

set\_location\_assignment PIN\_V21 -to ss3[1]

set\_location\_assignment PIN\_W22 -to ss3[2]

set\_location\_assignment PIN\_W21 -to ss3[3]

set\_location\_assignment PIN\_Y22 -to ss3[4]

set\_location\_assignment PIN\_Y21 -to ss3[5]

set\_location\_assignment PIN\_AA22 -to ss3[6]

set\_location\_assignment PIN\_AA20 -to ss2[0]

set\_location\_assignment PIN\_AB20 -to ss2[1]

set\_location\_assignment PIN\_AA19 -to ss2[2]

set\_location\_assignment PIN\_AA18 -to ss2[3]

set\_location\_assignment PIN\_AB18 -to ss2[4]

set\_location\_assignment PIN\_AA17 -to ss2[5]

set\_location\_assignment PIN\_U22 -to ss2[6]

set\_location\_assignment PIN\_Y19 -to ss1[0]

set\_location\_assignment PIN\_AB17 -to ss1[1]

set\_location\_assignment PIN\_AA10 -to ss1[2]

set\_location\_assignment PIN\_Y14 -to ss1[3]

set\_location\_assignment PIN\_V14 -to ss1[4]

set\_location\_assignment PIN\_AB22 -to ss1[5]

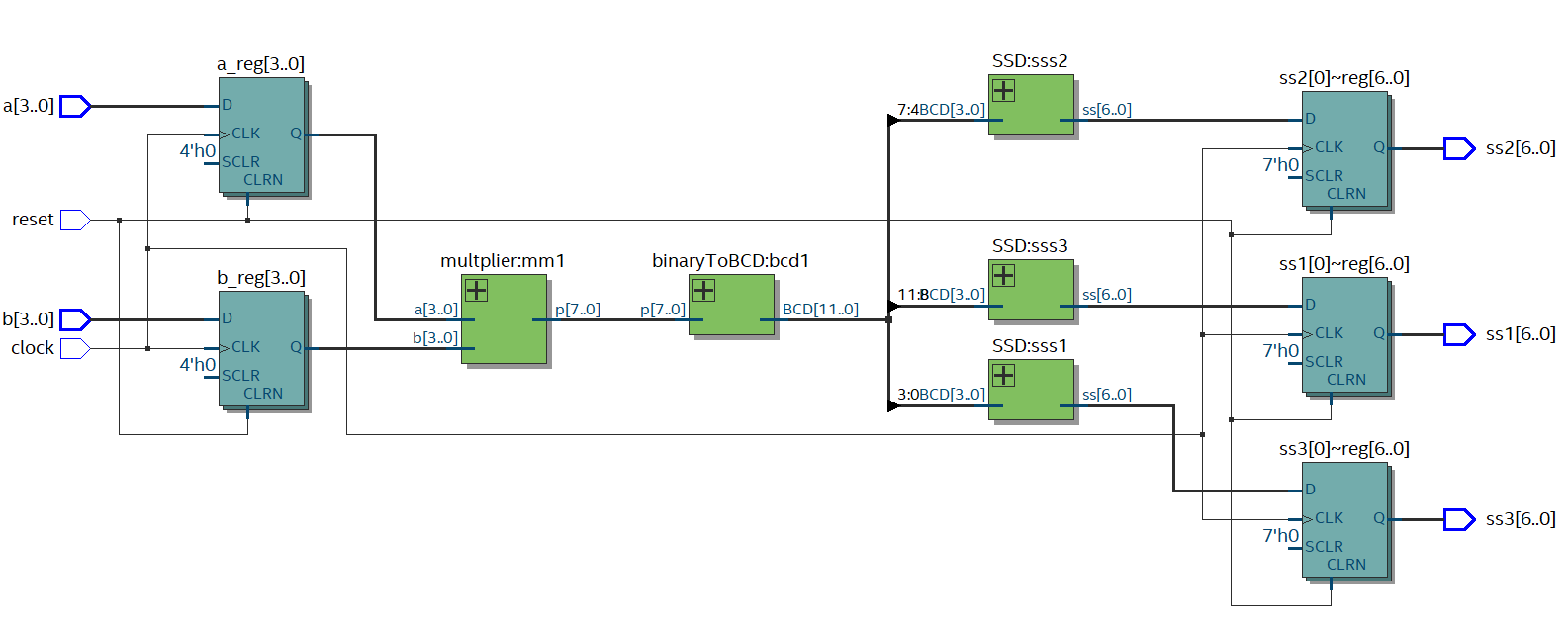
set\_location\_assignment PIN\_AB21 -to ss1[6]

set\_location\_assignment PIN\_M9 -to clock

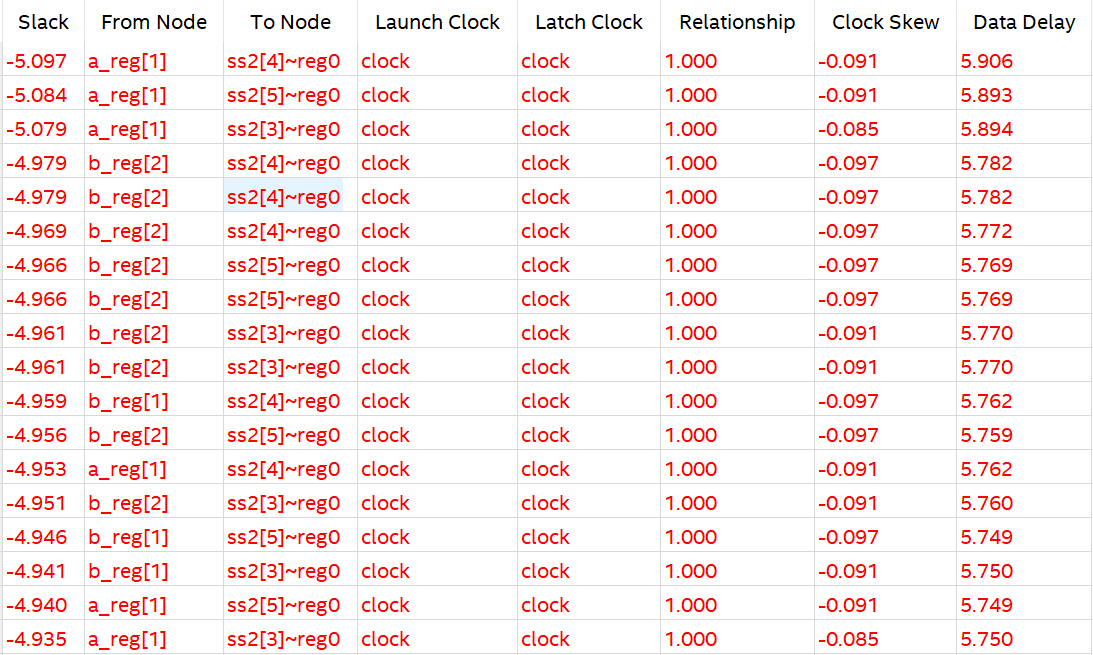
set\_location\_assignment PIN\_U7 -to reset

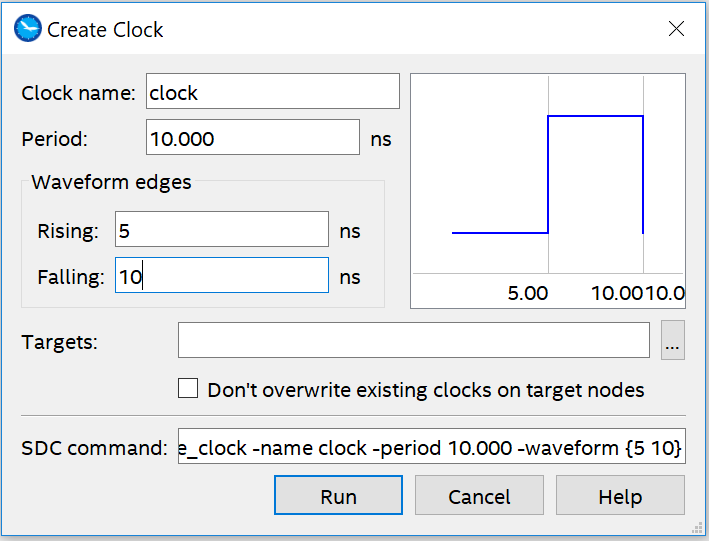
* Compile the multiplier design by following the steps on pages 18 – 21 in document [1].

The design block diagram is shown in the following figure



In this project we don’t create sdc file then quartus will create default one with constrain on clock equal 1 GHz.

* Open Tools🡪TimeQuest Timing Analyzer
* In the Task panel double click on Update Timing Netlist
* In the Task panel double click on Report Fmax Summary to see the max frequency you can work on ( Fmax = 164 MHz)
* To see if there are failing paths, in the Task panel double click on Report Top Failing Paths. It will show you that it has failing paths as the following figure 
* To solve this problem you must create SDC file and enter the constrain on the clock lower than 164 MHz
* To create SDC, in the TimeQuest Timing Analyzer click Constraints🡪Create Clock and enter clock name as clock (the same name in the rtl file), Period to 10 ns (to make clock = 100MHz), Rising = 5 ns and Falling = 10 ns then press Run to create SDC file.



* To write SDC file, from Constraints🡪Write SDC file🡪OK.
* In the task panel, click Update Timing Netlist then click Report Top Failing Paths. You will see that No failing paths found.
* To add created SDC file to your project follow the steps on pages 16 – 18 in document [1]. Then compile the project again by press clrt+L

1. **Pipelining**

If your design doesn’t meet the clock constraint in SDC file you must pipeline you design by divide the design to blocks and register the outputs of each block which are inputs the next block.

* Start a new project by following the steps on pages 6 – 12 of document [1]. Select the right FPGA for the DE0-CV board from “Table 1. DE-series FPGA device names” on page 10 in document [1].
* Add the following source files using the steps on pages 16 – 18 in document [1].
  + - 1. multiplier.vhd
      2. binaryToBCD.vhd
      3. SSD.vhd
      4. Shift\_add\_3.vhd
      5. integrated\_multiplier.vhd as in multiplier\_no\_sdc.vhd
      6. integrated\_multiplier.out.sdc, this file set constrain on the clock to be 200 MHz
* In the Files view, right click on integrated\_multiplier.vhd file and select “Set as Top-Level Entity”.
* Open integrated\_multiplier.qsf and add the following pin assignment lines to it

set\_location\_assignment PIN\_U13 -to a[0]

set\_location\_assignment PIN\_V13 -to a[1]

set\_location\_assignment PIN\_T13 -to a[2]

set\_location\_assignment PIN\_T12 -to a[3]

set\_location\_assignment PIN\_AA14 -to b[0]

set\_location\_assignment PIN\_AA13 -to b[1]

set\_location\_assignment PIN\_AB13 -to b[2]

set\_location\_assignment PIN\_AB12 -to b[3]

set\_location\_assignment PIN\_U21 -to ss3[0]

set\_location\_assignment PIN\_V21 -to ss3[1]

set\_location\_assignment PIN\_W22 -to ss3[2]

set\_location\_assignment PIN\_W21 -to ss3[3]

set\_location\_assignment PIN\_Y22 -to ss3[4]

set\_location\_assignment PIN\_Y21 -to ss3[5]

set\_location\_assignment PIN\_AA22 -to ss3[6]

set\_location\_assignment PIN\_AA20 -to ss2[0]

set\_location\_assignment PIN\_AB20 -to ss2[1]

set\_location\_assignment PIN\_AA19 -to ss2[2]

set\_location\_assignment PIN\_AA18 -to ss2[3]

set\_location\_assignment PIN\_AB18 -to ss2[4]

set\_location\_assignment PIN\_AA17 -to ss2[5]

set\_location\_assignment PIN\_U22 -to ss2[6]

set\_location\_assignment PIN\_Y19 -to ss1[0]

set\_location\_assignment PIN\_AB17 -to ss1[1]

set\_location\_assignment PIN\_AA10 -to ss1[2]

set\_location\_assignment PIN\_Y14 -to ss1[3]

set\_location\_assignment PIN\_V14 -to ss1[4]

set\_location\_assignment PIN\_AB22 -to ss1[5]

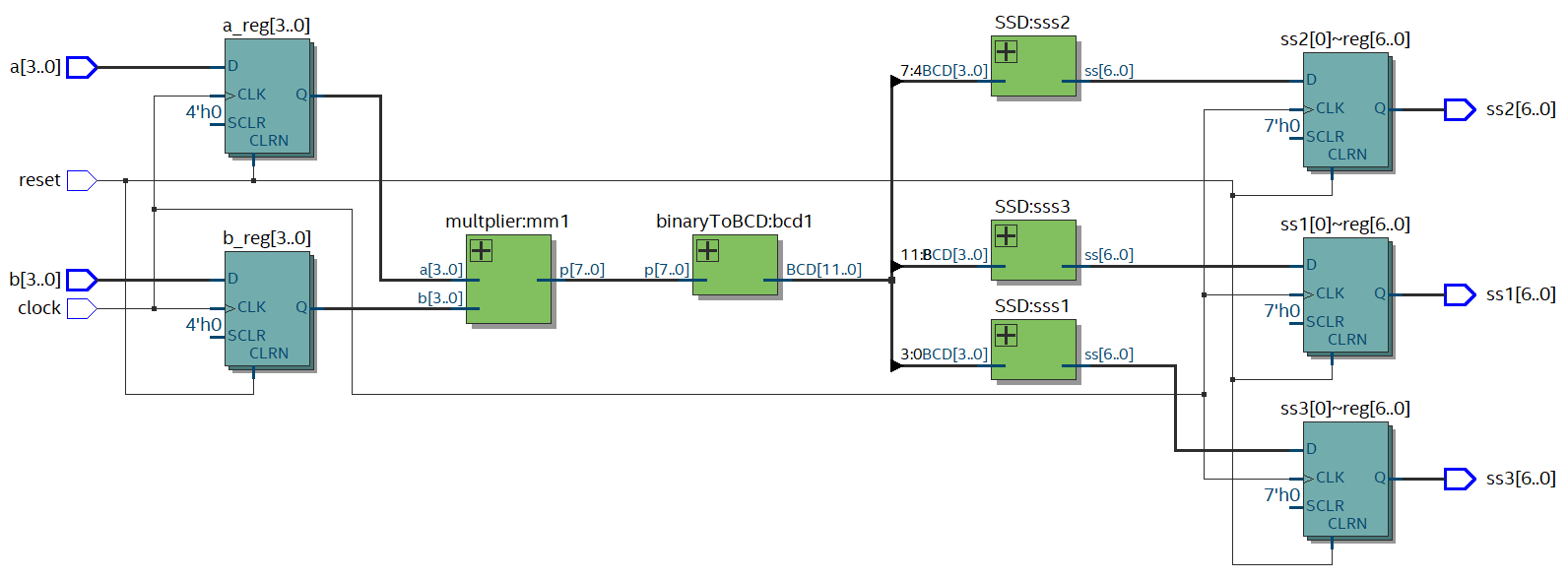
set\_location\_assignment PIN\_AB21 -to ss1[6]

set\_location\_assignment PIN\_M9 -to clock

set\_location\_assignment PIN\_U7 -to reset

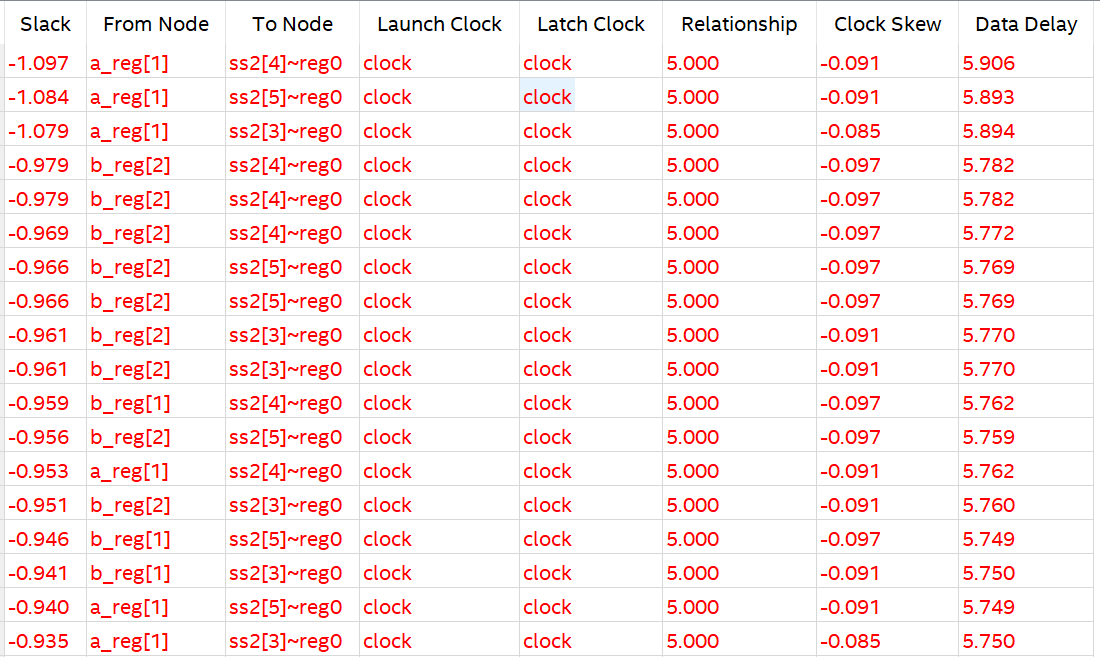
* Compile the multiplier design by following the steps on pages 18 – 21 in document [1].

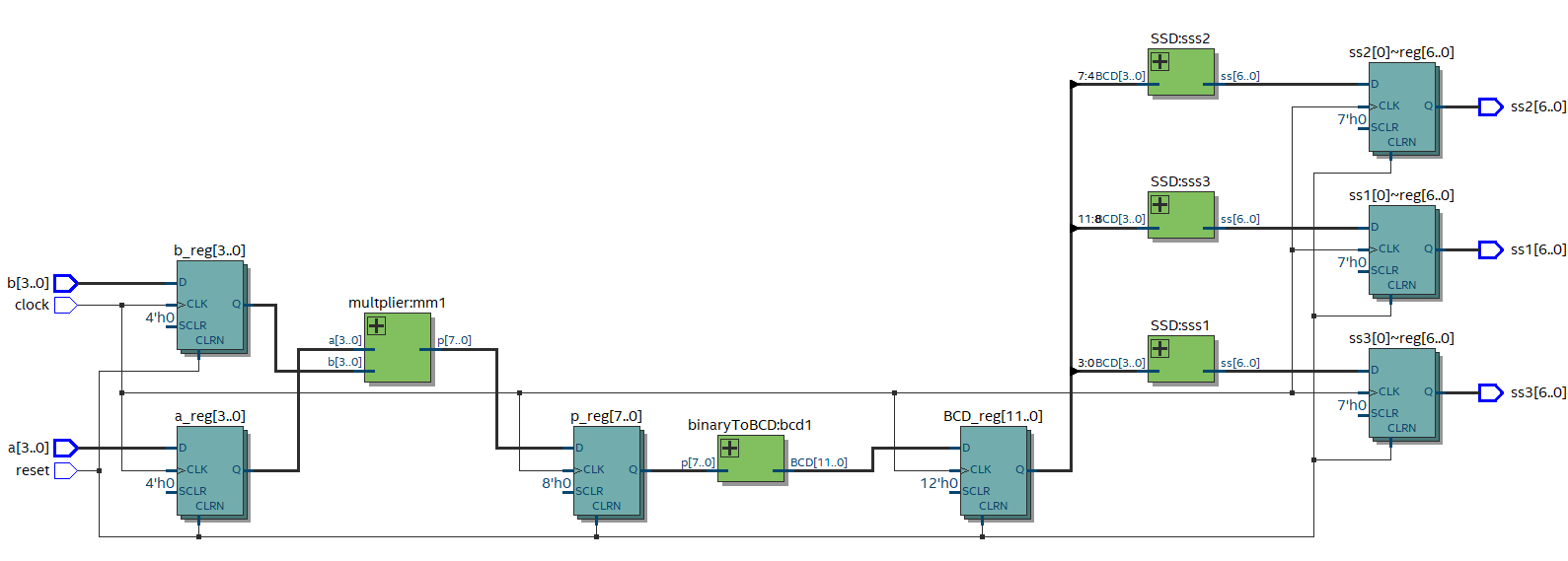
The design block diagram is shown in the following figure



* Open Tools🡪TimeQuest Timing Analyzer
* In the Task panel double click on Update Timing Netlist
* In the Task panel double click on Report Fmax Summary to see the max frequency you can work on ( Fmax = 164 MHz)

To see if there are failing paths, in the Task panel double click on Report Top Failing Paths. It will show you that it has failing paths as the following figure



* To solve this problem, we will pipeline the design as the following figure. 
* To make this you must change the RTL of integrated\_multiplier.vhd to register the results from multiplier, binaryToBCD and SSD
* Compile the design again by press ctrl+L
* Open Tools🡪TimeQuest Timing Analyzer
* In the Task panel double click on Update Timing Netlist
* In the Task panel double click on Report Fmax Summary to see the max frequency you can work on ( Fmax = 259.88 MHz)
* To see if there are failing paths, in the Task panel double click on Report Top Failing Paths. It will show you that No failing paths found.

1. **PLL**

DE0-CV board support 4 clocks, each one are 50 MHz. PLL is IP from Altera which can generate clock with different frequency from reference frequency.

* Start a new project by following the steps on pages 6 – 12 of document [1]. Select the right FPGA for the DE0-CV board from “Table 1. DE-series FPGA device names” on page 10 in document [1].
* Add the following source files using the steps on pages 16 – 18 in document [1].
  + - 1. multiplier.vhd
      2. binaryToBCD.vhd
      3. SSD.vhd
      4. shift\_add\_3.vhd
      5. multiplier\_clock.vhd as in multiplier\_no\_sdc.vhd
      6. integrated\_multiplier.out.sdc, this file set constrain on the clock to be 50 MHz
* In the Files view, right click on multiplier\_clock.vhd file and select “Set as Top-Level Entity”.
* Open multiplier\_clock.qsf and add the following pin assignment lines to it

set\_location\_assignment PIN\_U13 -to a[0]

set\_location\_assignment PIN\_V13 -to a[1]

set\_location\_assignment PIN\_T13 -to a[2]

set\_location\_assignment PIN\_T12 -to a[3]

set\_location\_assignment PIN\_AA14 -to b[0]

set\_location\_assignment PIN\_AA13 -to b[1]

set\_location\_assignment PIN\_AB13 -to b[2]

set\_location\_assignment PIN\_AB12 -to b[3]

set\_location\_assignment PIN\_U21 -to ss3[0]

set\_location\_assignment PIN\_V21 -to ss3[1]

set\_location\_assignment PIN\_W22 -to ss3[2]

set\_location\_assignment PIN\_W21 -to ss3[3]

set\_location\_assignment PIN\_Y22 -to ss3[4]

set\_location\_assignment PIN\_Y21 -to ss3[5]

set\_location\_assignment PIN\_AA22 -to ss3[6]

set\_location\_assignment PIN\_AA20 -to ss2[0]

set\_location\_assignment PIN\_AB20 -to ss2[1]

set\_location\_assignment PIN\_AA19 -to ss2[2]

set\_location\_assignment PIN\_AA18 -to ss2[3]

set\_location\_assignment PIN\_AB18 -to ss2[4]

set\_location\_assignment PIN\_AA17 -to ss2[5]

set\_location\_assignment PIN\_U22 -to ss2[6]

set\_location\_assignment PIN\_Y19 -to ss1[0]

set\_location\_assignment PIN\_AB17 -to ss1[1]

set\_location\_assignment PIN\_AA10 -to ss1[2]

set\_location\_assignment PIN\_Y14 -to ss1[3]

set\_location\_assignment PIN\_V14 -to ss1[4]

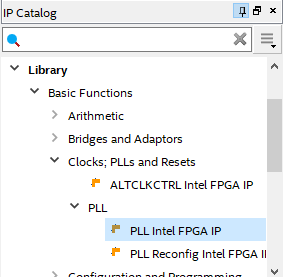
set\_location\_assignment PIN\_AB22 -to ss1[5]

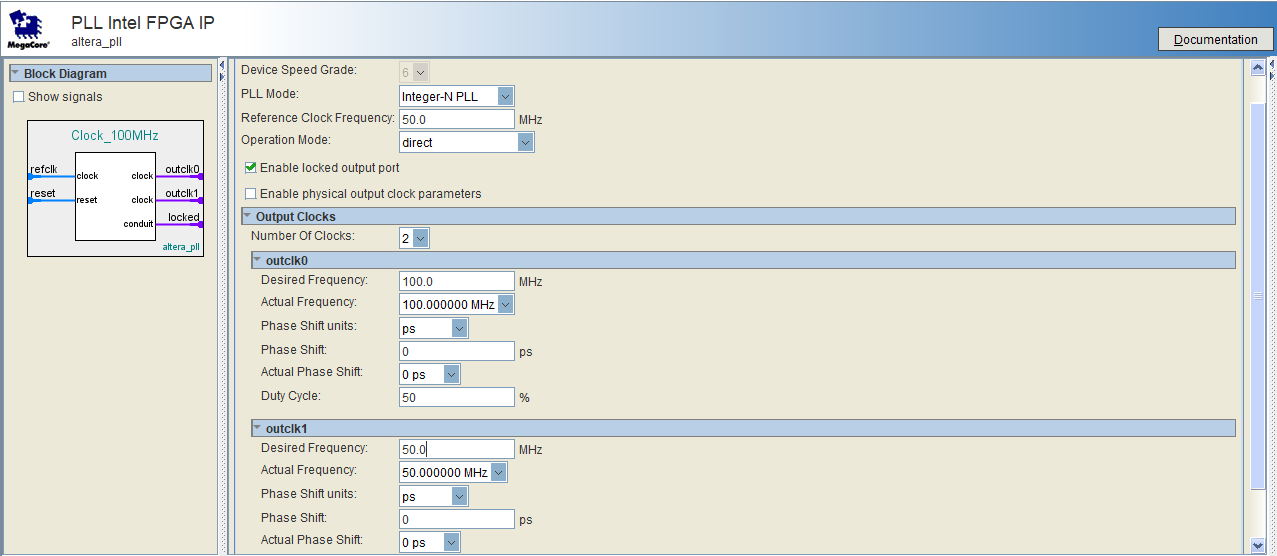
set\_location\_assignment PIN\_AB21 -to ss1[6]

set\_location\_assignment PIN\_M9 -to clock

set\_location\_assignment PIN\_U7 -to reset

* Select Tools🡪IP Catalog
* In the IP Catalog panel, select Library🡪Basic Functions 🡪Clocks; PLLs and Resets🡪PLL🡪 PLL Intel FPGA IP



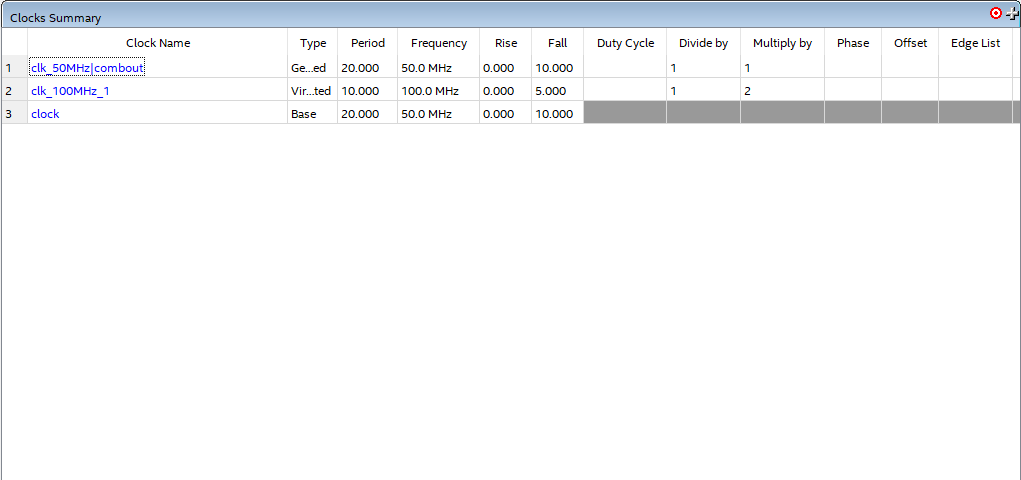
* Then Enter the PLL name as Clock\_100MHz then press OK.
* Enter the Reference Clock Frequency as 50MHz
* Enter Number Of Clocks to 2
* Enter the Desired Frequency of outclk0 as 100 MHz
* Enter the Desired Frequency of outclk1 as 50 MHz 
* Then press Finish
* Edit your muliplier\_clock.vhd to add PLL by instantiate it and read outclk0 and outclk1 only when locked is 1(Outclk0 is clk\_100MHz\_1, outclk1 is clk\_50MHz and refclk is your input clock)(you can find instantiation code in Files(project navigator)🡪Clock\_100MHz.qip🡪Clock\_100MHz.v)
* To make the new two generated clocks view in TimeQuest you need to add the following lines in .sdc file

create\_generated\_clock -source [get\_ports clock] -multiply\_by 1 -name clk\_50MHz

create\_generated\_clock -source [get\_ports clock] -multiply\_by 2 -name clk\_100MHz\_1

Then open Tools🡪 Timing Analyzer🡪Update Timing Netlist

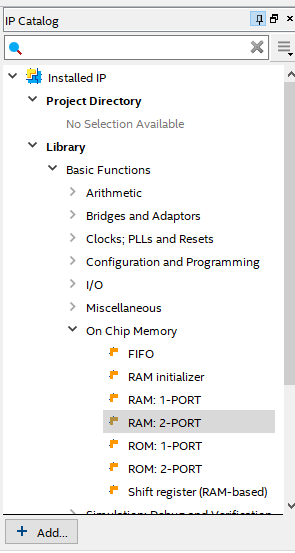
* Report🡪Diagnostic🡪Report Clocks



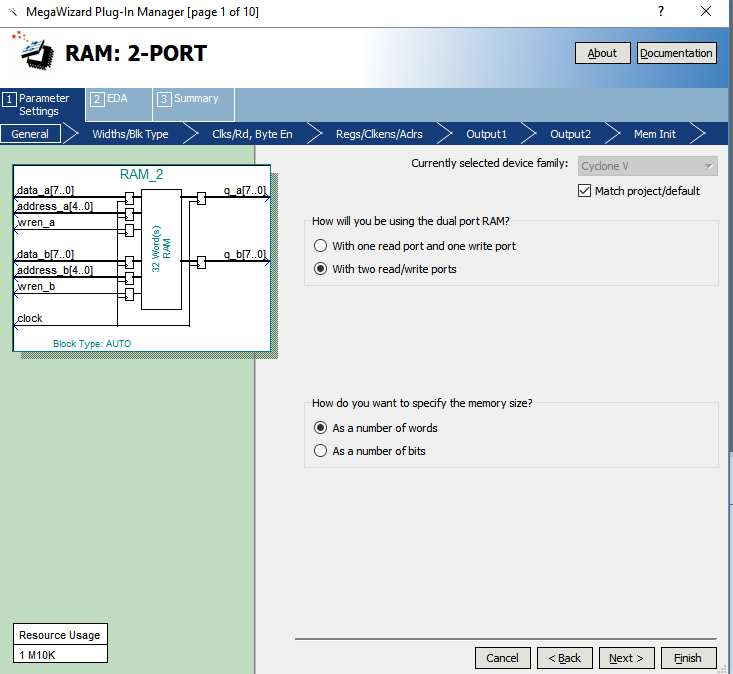
* Then compile again.

1. **Dual port RAM:**

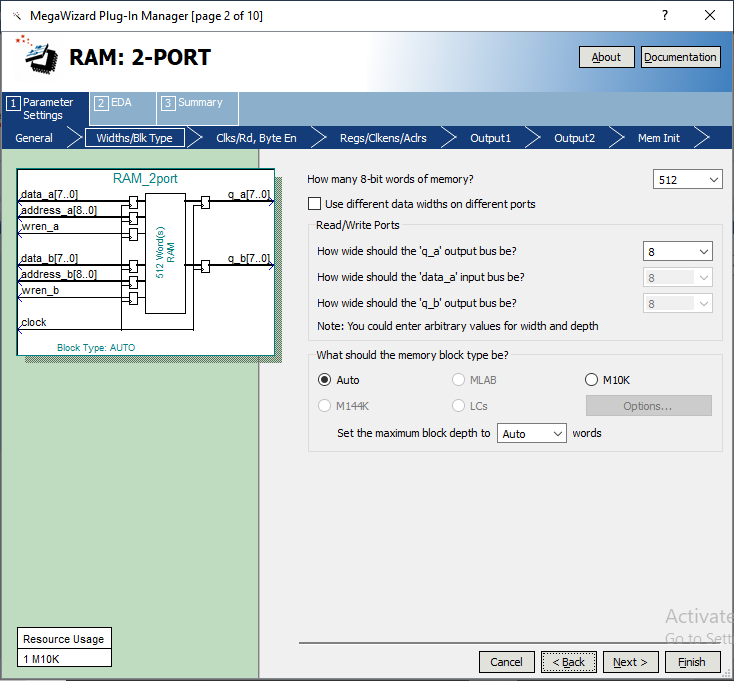
* Select Tools IP Catalog
* In the IP Catalog panel, select Library/ On Chip Memory/ RAM: 2-PORT



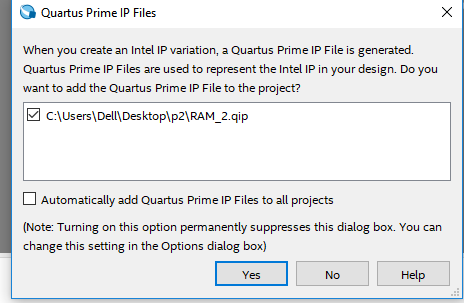
* Then Enter the RAM name as RAM\_2port then press VHDL then press OK.
* Choose “With two read/write ports” then press Next



* Make “How wide should ‘q\_a’ output bus be?” to 8
* Make “How many 1-bit words of memory?” to 512 then press Next



* Choose “Dual clock: use separate clocks for A and B ports” then press Finish
* Then press Finish again
* Then press Yes



* You will also use the PLL with the 2 output clocks. Port a of the DPR will write the incoming data (for simplicity make the data equal to the address). Port A runs at 50 MHz. Port B will read from Port B which runs at 100 MHz. In the ping pong buffer the memory array is divided into 2 equal blocks. You will wait until you fill block A and then fill block B using Port A. You will wait until you fill Block A and then start reading Block A from port B at 100 MHz and then you will read Block B from Port B.
* Below is a code example for a PingPong buffer ready for simulation. Simulate the code and make sure the PLL and DPR is working as intended.

library ieee ;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

----------------------------------------------------

entity PingPong is

port( clock: in std\_logic;

Rst: in std\_logic;

Q: out std\_logic\_vector(7 downto 0)

);

end PingPong;

----------------------------------------------------

architecture behv of PingPong is

component Dual\_Clock is

port (

areset : in std\_logic := 'X'; -- clk

inclk0 : in std\_logic := 'X'; -- reset

c0 : out std\_logic; -- clk

c1 : out std\_logic; -- clk

locked : out std\_logic -- export

);

end component Dual\_Clock;

component RAM\_2port IS

PORT

(

address\_a : IN STD\_LOGIC\_VECTOR (8 DOWNTO 0);

address\_b : IN STD\_LOGIC\_VECTOR (8 DOWNTO 0);

clock\_a : IN STD\_LOGIC := '1';

clock\_b : IN STD\_LOGIC ;

data\_a : IN STD\_LOGIC\_VECTOR (7 DOWNTO 0);

data\_b : IN STD\_LOGIC\_VECTOR (7 DOWNTO 0);

wren\_a : IN STD\_LOGIC := '0';

wren\_b : IN STD\_LOGIC := '0';

q\_a : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0);

q\_b : OUT STD\_LOGIC\_VECTOR (7 DOWNTO 0)

);

END component;

signal Clk\_50MHz,Clk\_100MHz : std\_logic;

signal locked : std\_logic;

signal address\_a : STD\_LOGIC\_VECTOR (8 DOWNTO 0);

signal address\_b : STD\_LOGIC\_VECTOR (8 DOWNTO 0);

signal clock\_a : STD\_LOGIC := '1';

signal clock\_b : STD\_LOGIC ;

signal data\_a : STD\_LOGIC\_VECTOR (7 DOWNTO 0);

signal data\_b : STD\_LOGIC\_VECTOR (7 DOWNTO 0);

signal wren\_a : STD\_LOGIC := '0';

signal wren\_b : STD\_LOGIC := '0';

signal q\_a : STD\_LOGIC\_VECTOR (7 DOWNTO 0);

signal q\_b : STD\_LOGIC\_VECTOR (7 DOWNTO 0);

signal Pre\_Q: std\_logic\_vector(8 downto 0);

begin

Dual\_Clock\_inst : Dual\_Clock PORT MAP (

areset => Rst,

inclk0 => clock,

c0 => Clk\_50MHz,

c1 => Clk\_100MHz,

locked => locked

);

RAM\_2portIns: RAM\_2port

PORT map

(

address\_a => address\_a,

address\_b => address\_b,

clock\_a => Clk\_50MHz,

clock\_b => Clk\_100MHz,

data\_a => data\_a,

data\_b => data\_b,

wren\_a => '1',

wren\_b => '0',

q\_a => q\_a,

q\_b => q\_b

);

process(Clk\_50MHz, Rst)

begin

if Rst = '1' then

Pre\_Q <= (others=>'0');

elsif (Clk\_50MHz='1' and Clk\_50MHz'event) then

if locked = '1' then

Pre\_Q <= Pre\_Q + 1;

end if;

end if;

end process;

-- concurrent assignment statement

address\_a <= Pre\_Q;

address\_b <= not(Pre\_Q(8)) & Pre\_Q(7 downto 0);

data\_a <= Pre\_Q(7 downto 0);

data\_b <= "00000000";

Q <= q\_b;

end behv;

1. Fall 2019 version [↑](#footnote-ref-1)