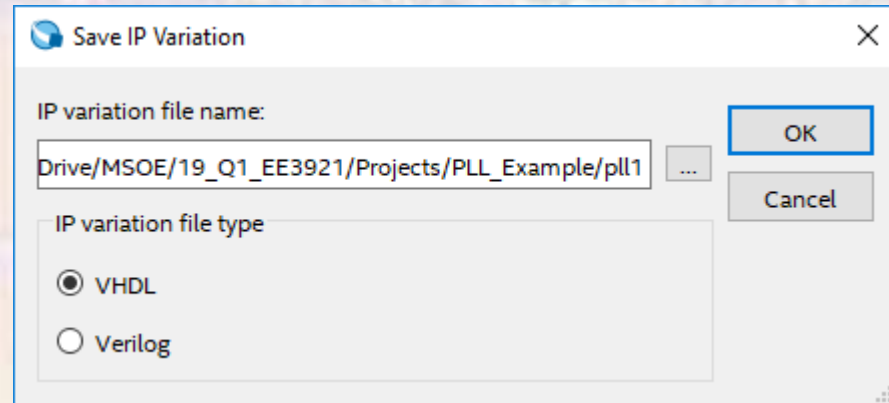
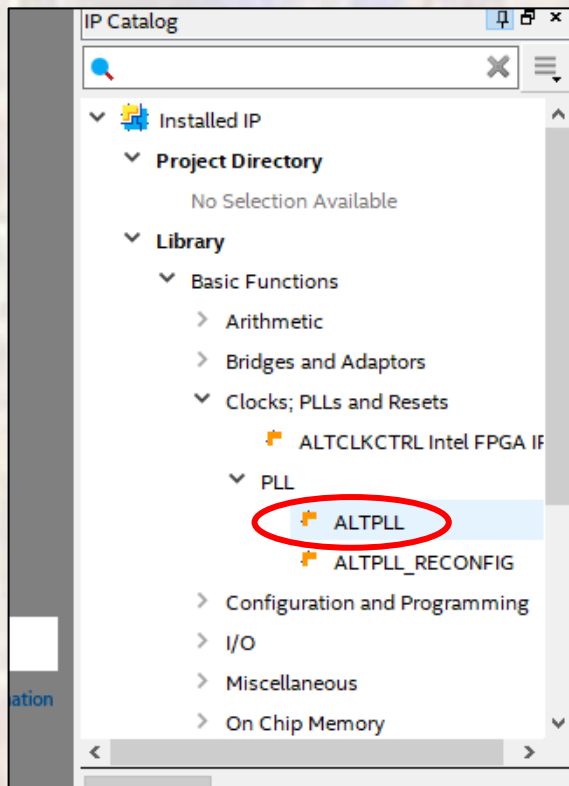


# MAX 10 PLL MegaWizard

Last updated 7/20/23

# MAX10 PLL MegaWizard

- MegaFunction AltPLL
  - Library element



# MAX10 PLL MegaWizard

- MegaFunction AltPLL - configuration

MegaWizard Plug-In Manager [page 1 of 12]

**ALTPLL** About Documentation

1 Parameter Settings 2 PLL Reconfiguration 3 Output Clocks 4 EDA 5 Summary

General/Modes Inputs/Lock Bandwidth/SS Clock switchover

Currently selected device family: MAX 10

Match project/default

Able to implement the requested PLL

General

Which device speed grade will you be using? Any

Use military temperature range devices only

What is the frequency of the inclk0 input? 50,000 MHz

Set up PLL in LVDS mode Data rate: Not Available Mbps

PLL Type

Which PLL type will you be using?

Fast PLL  Enhanced PLL  Select the PLL type automatically

Operation Mode

How will the PLL outputs be generated?

Use the feedback path inside the PLL

In normal mode

In source-synchronous compensation Mode

In zero delay buffer mode

Connect the fbimic port (bidirectional)

With no compensation

Create an 'fbim' input for an external feedback (External Feedback Mode)

Which output clock will be compensated for? c0

Cancel < Back Next > Finish

# MAX10 PLL MegaWizard

- MegaFunction AltPll - configuration

MegaWizard Plug-In Manager [page 2 of 12]

**ALTPLL** About Documentation

1 Parameter Settings 2 PLL Reconfiguration 3 Output Clocks 4 EDA 5 Summary

General/Modes > **Inputs/Lock** > Bandwidth/SS > Clock switchover >

pll1

inclk0 areset pfdena

inclk0 frequency: 50.000 MHz  
Operation Mode: Normal

Clk	Ratio	Ph (dg)	DC (%)
c0	1/50	0.00	50.00
c1	1/100	18.00	25.00

c0 c1 locked

MAX 10

Able to implement the requested PLL

Optional Inputs

- Create an 'pllena' input to selectively enable the PLL
- Create an 'areset' input to asynchronously reset the PLL
- Create an 'pfdena' input to selectively enable the phase/frequency detector

Lock Output

- Create 'locked' output
- Enable self-reset on loss lock

Advanced Parameters

Using these parameters is recommended for advanced users only

- Create output file(s) using the 'Advanced' PLL parameters

- Configurations with output dock(s) that use cascade counters are not supported

Cancel < Back **Next >** Finish

# MAX10 PLL MegaWizard

- MegaFunction AltPll - configuration

MegaWizard Plug-In Manager [page 3 of 12]

**ALTPLL** [About] [Documentation]

1 Parameter Settings 2 PLL Reconfiguration 3 Output Clocks 4 EDA 5 Summary

General/Modes > Inputs/Lock > Bandwidth/SS > Clock switchover >

pll1

incik0 areset pfdena c0 locked

incik0 frequency: 50,000 MHz  
Operation Mode: Normal

Clk	Ratio	Ph. (deg)	DC (%)
c0	1/1	0.00	50.00

MAX 10

Able to implement the requested PLL

**Spread Spectrum**  
The spread spectrum feature allows for a modulation of the PLL clock frequency. The range of the clock frequency deviation is determined by the 'down spread' while 'modulation frequency' controls their period.

Use spread spectrum feature and

Set down spread to 0.500 percent

Set modulation frequency to 50,000 KHz

**Bandwidth**  
A lower bandwidth will result in better jitter rejection and less drift at the expense of a slower PLL.

How would you like to specify bandwidth?

Auto  
 Preset: Low  
 Custom:

Set bandwidth to 1.00

Actual achieved bandwidth

More Details >>

MegaWizard Plug-In Manager [page 4 of 12]

**ALTPLL** [About] [Documentation]

1 Parameter Settings 2 PLL Reconfiguration 3 Output Clocks 4 EDA 5 Summary

General/Modes > Inputs/Lock > Bandwidth/SS > Clock switchover >

pll1

incik0 areset pfdena c0 locked

incik0 frequency: 50,000 MHz  
Operation Mode: Normal

Clk	Ratio	Ph. (deg)	DC (%)
c0	1/1	0.00	50.00

MAX 10

Able to implement the requested PLL

**Clock Switchover**

Create an 'incik1' input for a second input dock

What is the frequency of the 'incik1' input? 100,000 MHz

**Input Clock Switch**

Create a 'clkswitch' input to manually select between the input docks (The 'clkswitch' input will behave as an input clock selection control input)

Allow PLL to automatically control the switching between input docks (The 'clkswitch' input will behave as a manual override control input)

Create a 'clkswbad' input to dynamically control the switching between input docks (Form the input clock switchover after 1 input dock cycles)

Create an 'activedock' output to indicate the input dock being used (incik0 is being used/ 1 incik1 is being used)

Create a 'clkbad' output for each input dock (input dock is toggling/ 1 input dock is not toggling)

[Cancel] [Back] [Next >] [Finish]

MegaWizard Plug-In Manager [page 5 of 12]

**ALTPLL** [About] [Documentation]

1 Parameter Settings 2 PLL Reconfiguration 3 Output Clocks 4 EDA 5 Summary

General/Modes > Inputs/Lock > Bandwidth/SS > Clock switchover >

pll1

incik0 areset pfdena c0 locked

incik0 frequency: 50,000 MHz  
Operation Mode: Normal

Clk	Ratio	Ph. (deg)	DC (%)
c0	1/1	0.00	50.00

MAX 10

Able to implement the requested PLL

**Dynamic Reconfiguration**

Create optional inputs for dynamic reconfiguration

Used for non-phase (e.g. frequency, duty cycle, bandwidth, etc.) reconfiguration  
- Note: Reconfiguration with cascaded counters may not work correctly

**Initial Configuration File**

Use the following initial configuration file to initialize the altpll\_reconfig megafunction (Valid file formats are the Hexadecimal (Intel-format) [.hex] and the Memory Initialization File)

File name: \apl1.mif [Browse ...]

**Additional Configuration File**

You may create additional configuration file(s) for the current PLL settings. These files may be used to initialize the altpll\_reconfig megafunction.

To create a configuration file, enter a valid file name and press the 'Generate A Configuration File' button (Valid file formats are the Hexadecimal (Intel-format) [.hex] and the Memory Initialization File [.mif]).

File name: [Browse ...]

[Generate a Configuration File]

**Dynamic Phase Reconfiguration**

Enable phase shift step resolution

Create optional inputs for dynamic phase reconfiguration

[Cancel] [Back] [Next >] [Finish]

# MAX10 PLL MegaWizard

- MegaFunction AltPll – configuration 1

MegaWizard Plug-In Manager [page 6 of 12]

**ALTPLL** [About] [Documentation]

1 Parameter Settings | 2 PLL Reconfiguration | 3 Output Clocks | 4 EDA | 5 Summary

clk c0 > clk c1 > clk c2 > clk c3 > clk c4

**c0 - Core/External Output Clock**  
Able to implement the requested PLL

Use this clock

Clock Tap Settings

Requested Settings	Actual Settings
1.00000000 MHz	1.000000
1	1
50	50
0.00 deg	0.00
50.00	50.00

Per Clock Feasibility Indicators  
c0 c1 c2 c3 c4

Cancel < Back Next > Finish

**1MHz clk**  
**50% duty cycle**



# MAX10 PLL MegaWizard

- MegaFunction AltPll – configuration 2

**500KHz clk**  
**100ns ph shift**  
**25% duty cycle**

**c1 - Core/External Output Clock**  
Able to implement the requested PLL

Use this clock

**Requested Settings**

Requested Settings	Actual Settings
100.00000000 MHz	0.500000
1	1
100	100
100.00 ns	100.00
25.00	25.00

**Actual Settings**

Description	Val
Primary dock VCO frequency (MHz)	30..
Modulus for M counter	6

**Per Clock Feasibility Indicators**

c0 c1 c2 c3 c4

# MAX10 PLL MegaWizard

- MegaFunction AltPll - configuration

MegaWizard Plug-In Manager [page 12 of 12]

**ALTPLL** [About] [Documentation]

1 Parameter Settings | 2 PLL Reconfiguration | 3 Output Clocks | 4 EDA | 5 Summary

pll1

inclk0  
areset  
pfdena

inclk0 frequency: 50.000 MHz  
Operation Mode: Normal

Clk	Ratio	Ph (dg)	DC (%)
c0	1/50	0.00	50.00
c1	1/100	18.00	25.00

locked

MAX 10

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\PLL\_Example\

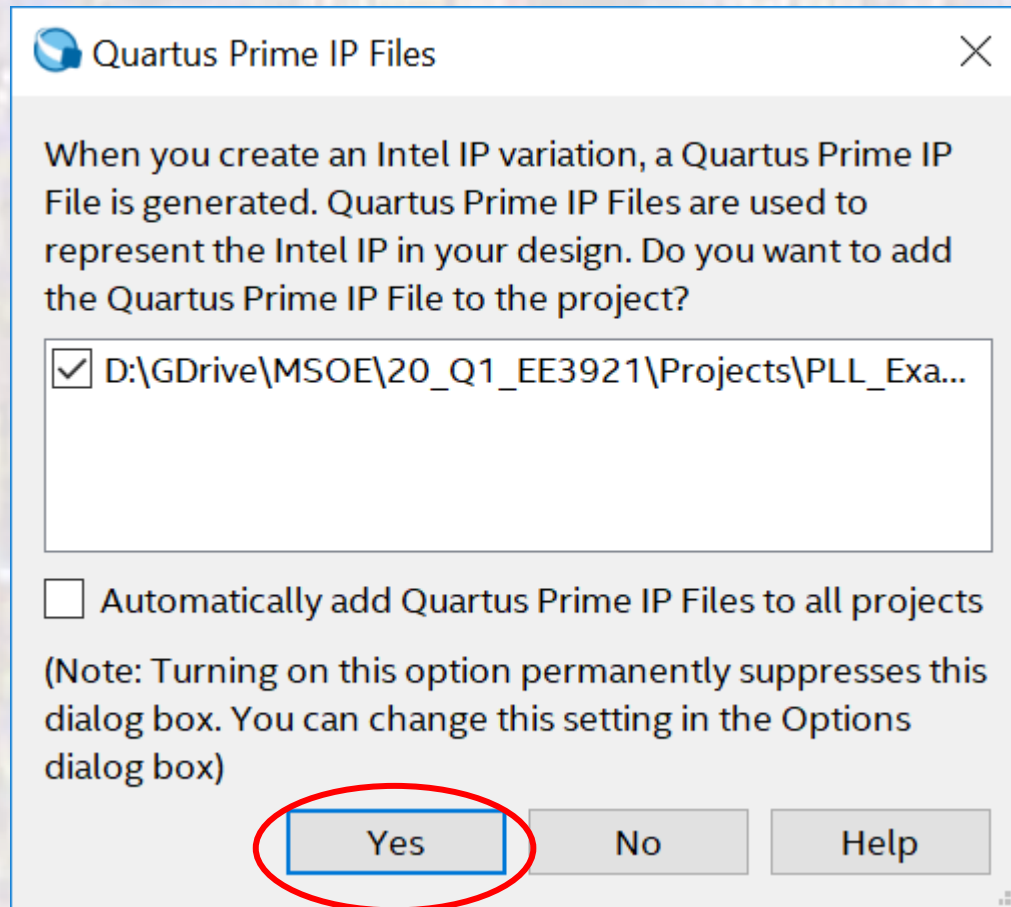
File	Description
<input checked="" type="checkbox"/> pll1.vhd	Variation file
<input checked="" type="checkbox"/> pll1.ppf	PinPlanner ports PPF file
<input type="checkbox"/> pll1.inc	AHDL Include file
<input checked="" type="checkbox"/> pll1.cmp	VHDL component declaration file
<input type="checkbox"/> pll1.bsf	Quartus Prime symbol file
<input checked="" type="checkbox"/> pll1_inst.vhd	Instantiation template file

[Cancel] [Back] [Next >] [Finish]



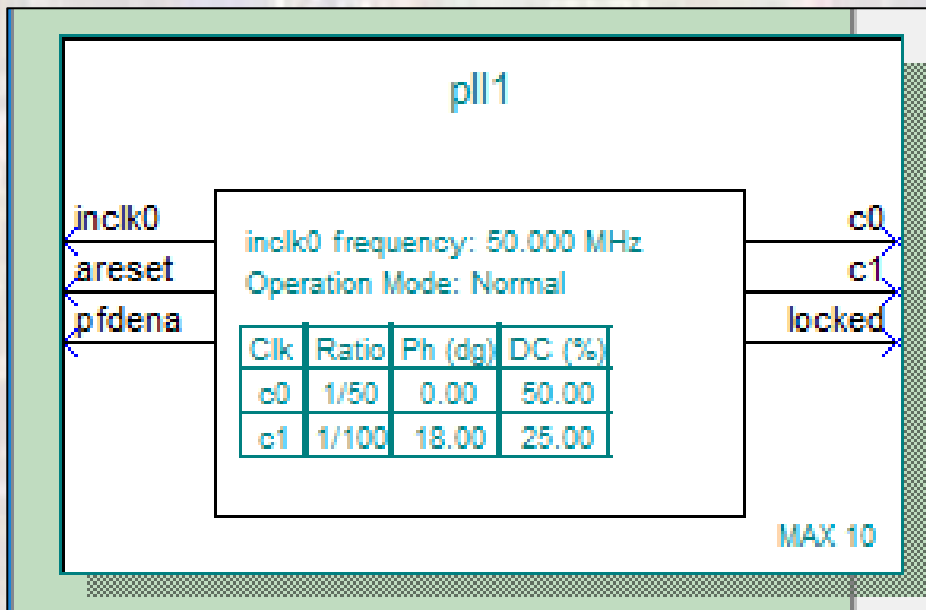
# MAX10 PLL MegaWizard

- MegaFunction AltPll - configuration



# MAX10 PLL MegaWizard

- MegaFunction AltPll
  - Component information



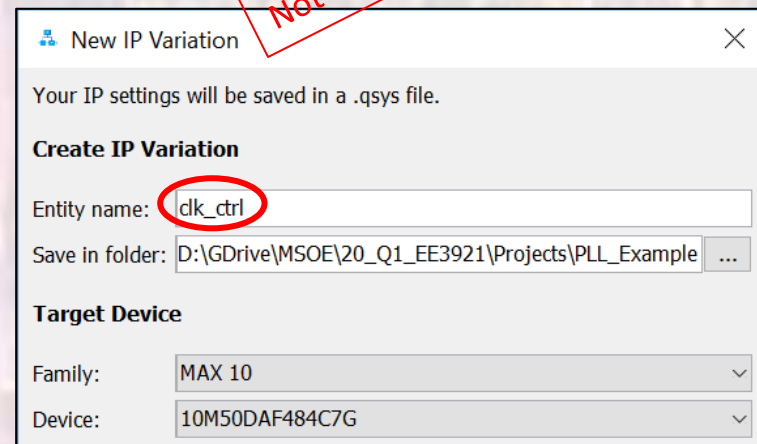
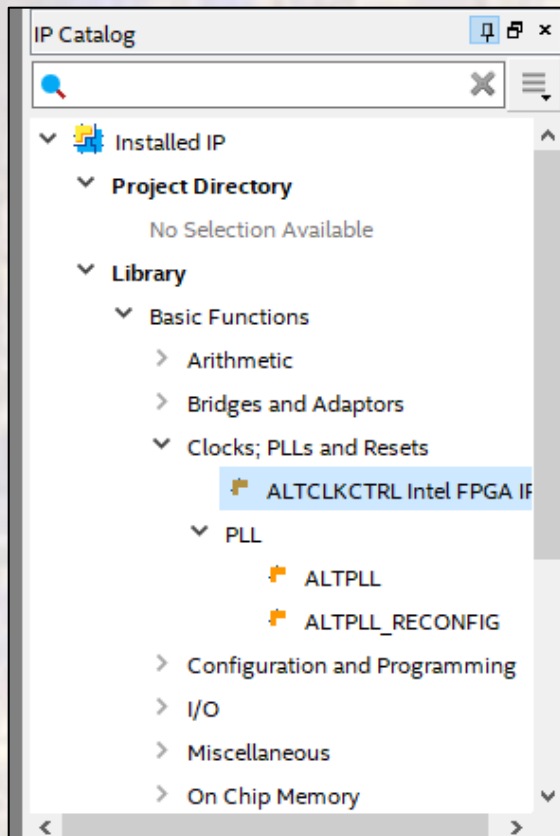
```
component pll1
PORT
(
    areset    : IN STD_LOGIC := '0';
    inclk0    : IN STD_LOGIC := '0';
    pfdena    : IN STD_LOGIC := '1';
    c0        : OUT STD_LOGIC ;
    c1        : OUT STD_LOGIC ;
    locked    : OUT STD_LOGIC
);
end component;
```

```
pll1_inst : pll1 PORT MAP (
    areset => areset_sig,
    inclk0 => inclk0_sig,
    pfdena  => pfdena_sig,
    c0      => c0_sig,
    c1      => c1_sig,
    locked  => locked_sig
);
```

# MAX10 PLL MegaWizard

- MegaFunction AltClkCtrl – stand alone example
- Required to get CLOCK\_50 to the PLL input

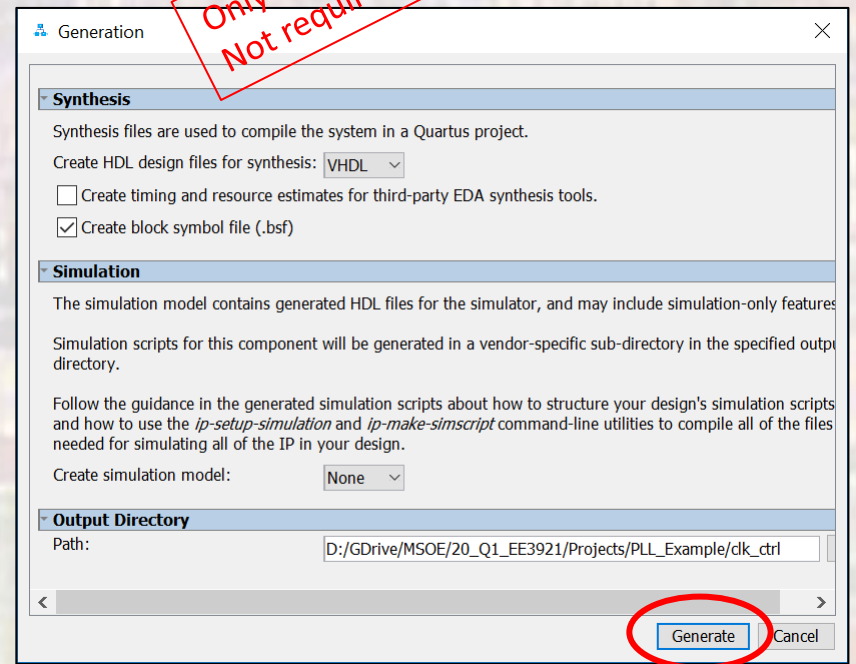
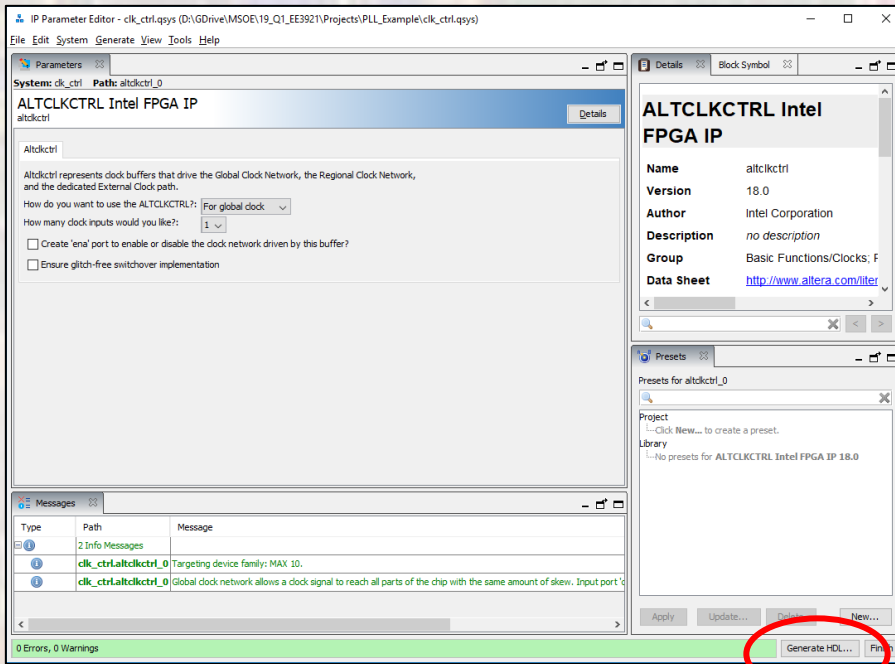
Only required when the PLL is the only block  
Not required for any of our designs



# MAX10 PLL MegaWizard

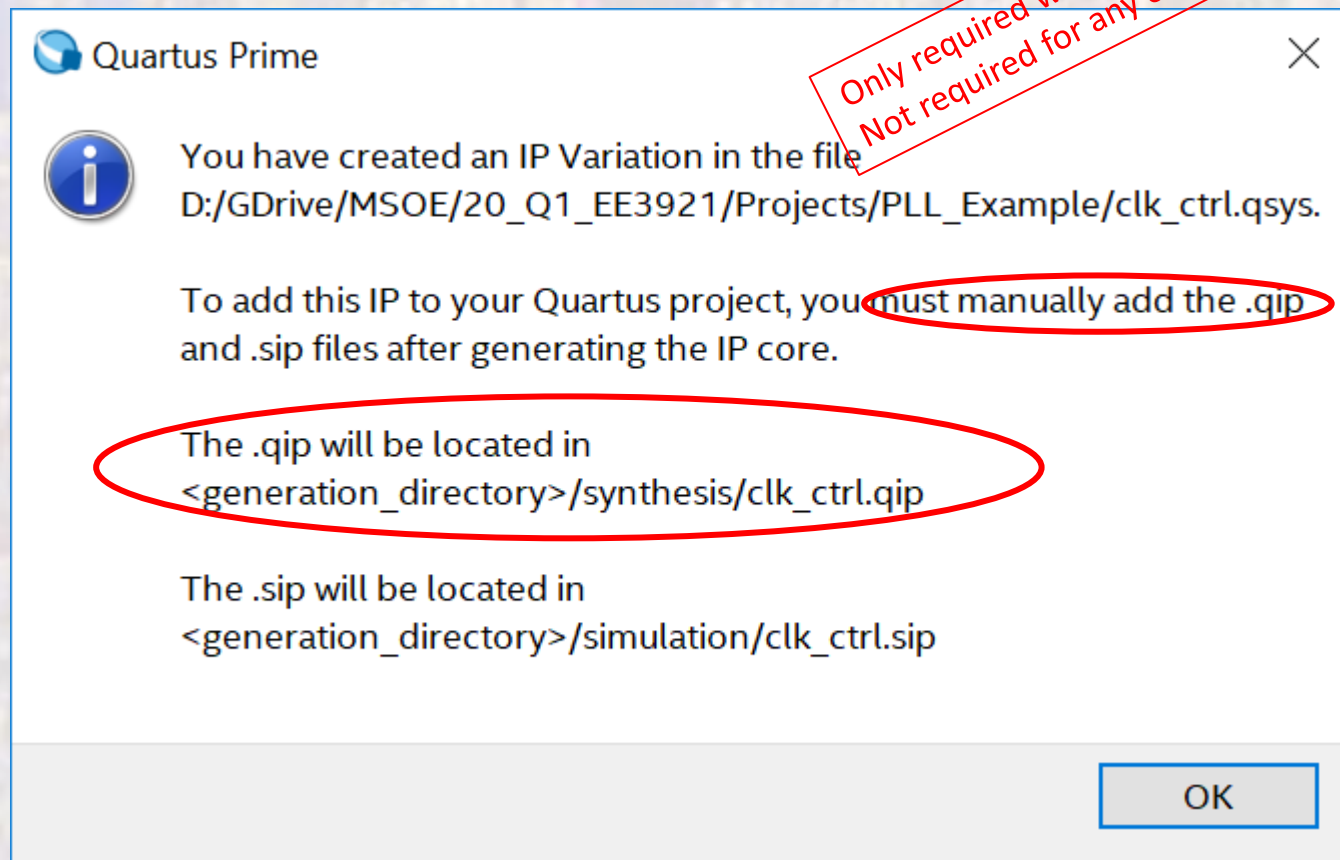
- MegaFunction AltClkCtrl – stand alone example
- Required to get CLOCK\_50 to the PLL input

Only required when the PLL is the only block  
Not required for any of our designs



# MAX10 PLL MegaWizard

- MegaFunction AltClkCtrl – stand alone example
- Required to get CLOCK\_50 to the PLL input





# MAX10 PLL MegaWizard

- MegaFunction AltPll – stand alone example

```
--
-- pll_example_de10.vhd1
--
-- by: johnsontimoj
--
-- created: 8/12/2018
--
-- version: 0.0
--
-----
--
-- PLL example - de10 implementation
--
-- inputs: CLK, enable, reset
--
-- outputs: 3 clocks - different frequencies and phases, locked
--
-- Use System Console - ADC Toolkit for validation
--
-----
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity pll_example_de10 is
  port (
    CLOCK_50: in std_logic;
    SW: in std_logic_vector(9 downto 0);
    ARDUINO_IO: inout std_logic_vector(4 downto 0)
  );
end entity;
```

```
architecture hardware of pll_example_de10 is
  signal clk_50_intermediate: std_logic;

  component pll1
    PORT
    (
      areset      : IN STD_LOGIC := '0';
      inclk0      : IN STD_LOGIC := '0';
      pfdena      : IN STD_LOGIC := '1';
      c0          : OUT STD_LOGIC ;
      c1          : OUT STD_LOGIC ;
      locked      : OUT STD_LOGIC
    );
  end component;

  component clk_ctr1 is
    port (
      inclk : in  std_logic := '1';
      outclk : out std_logic := '0'
    );
  end component clk_ctr1;

begin
  pll1_inst : pll1 PORT MAP (
    areset      => SW(0),
    inclk0      => clk_50_intermediate,
    pfdena      => ARDUINO_IO(0),
    c0          => ARDUINO_IO(1),
    c1          => ARDUINO_IO(2),
    locked      => ARDUINO_IO(3)
  );

  clk1 : clk_ctr1
  port map (
    inclk => CLOCK_50,
    outclk => clk_50_intermediate
  );
end architecture;
```

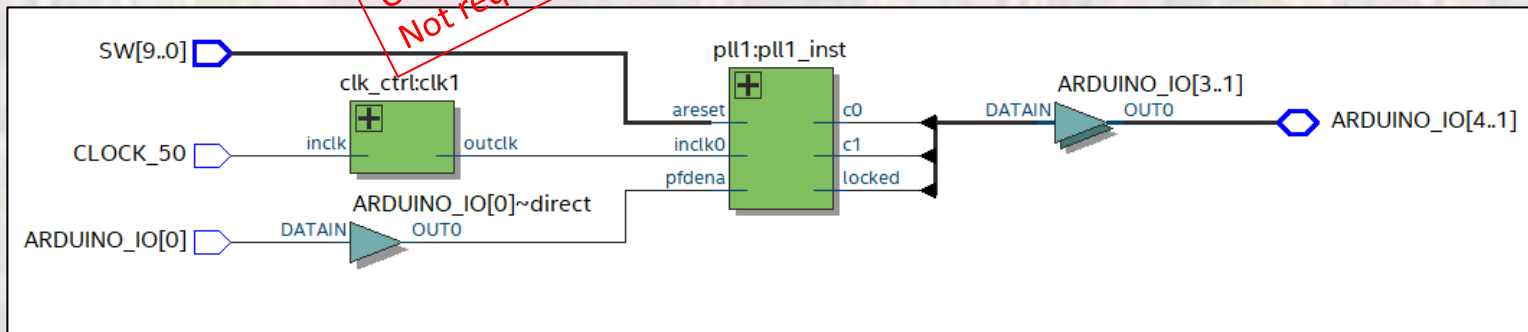
Only required when the PLL is the only block  
Not required for any of our designs



# MAX10 PLL MegaWizard

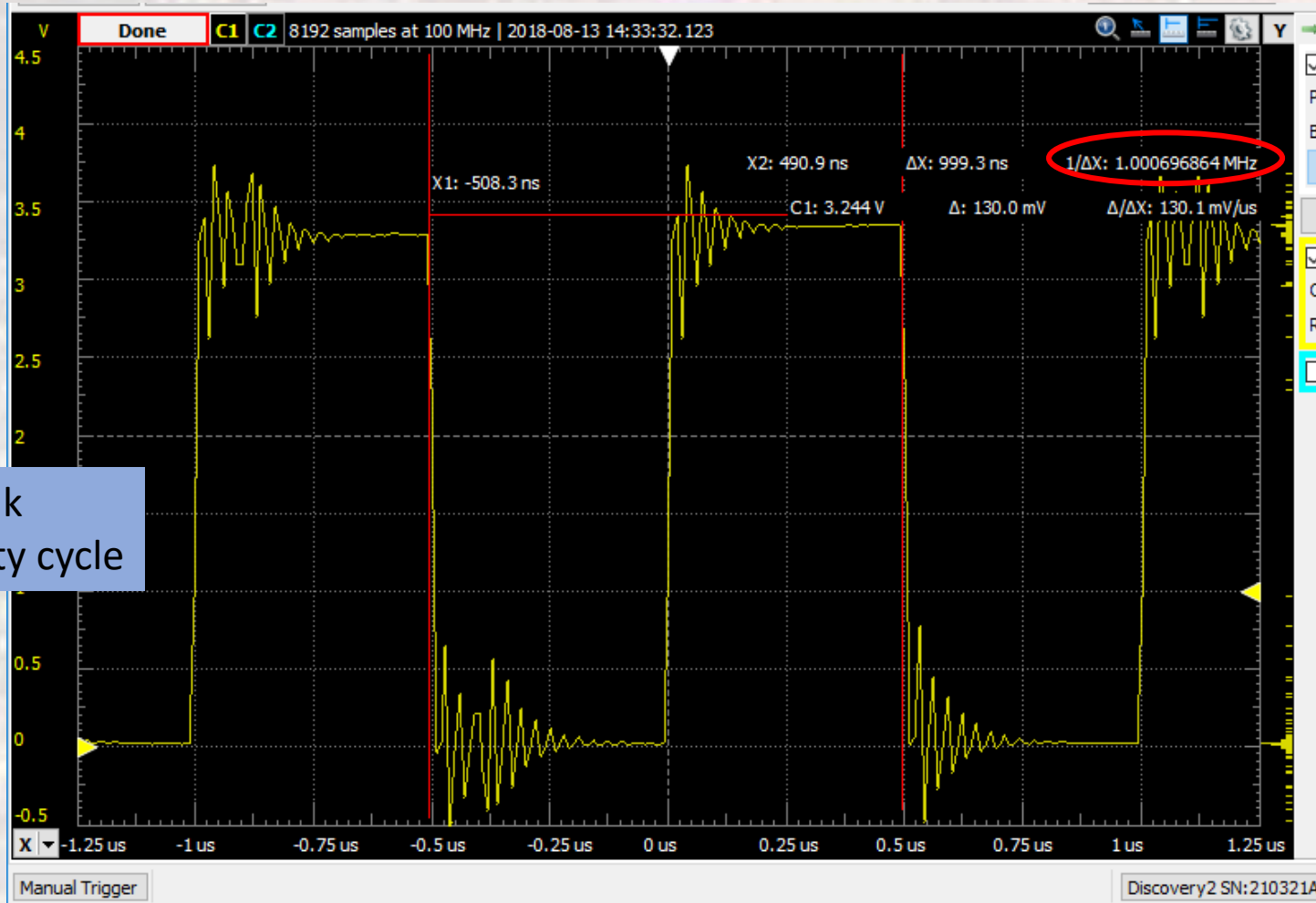
- MegaFunction AltPll – stand alone example

Only required when the PLL is the only block  
Not required for any of our designs



# MAX10 PLL MegaWizard

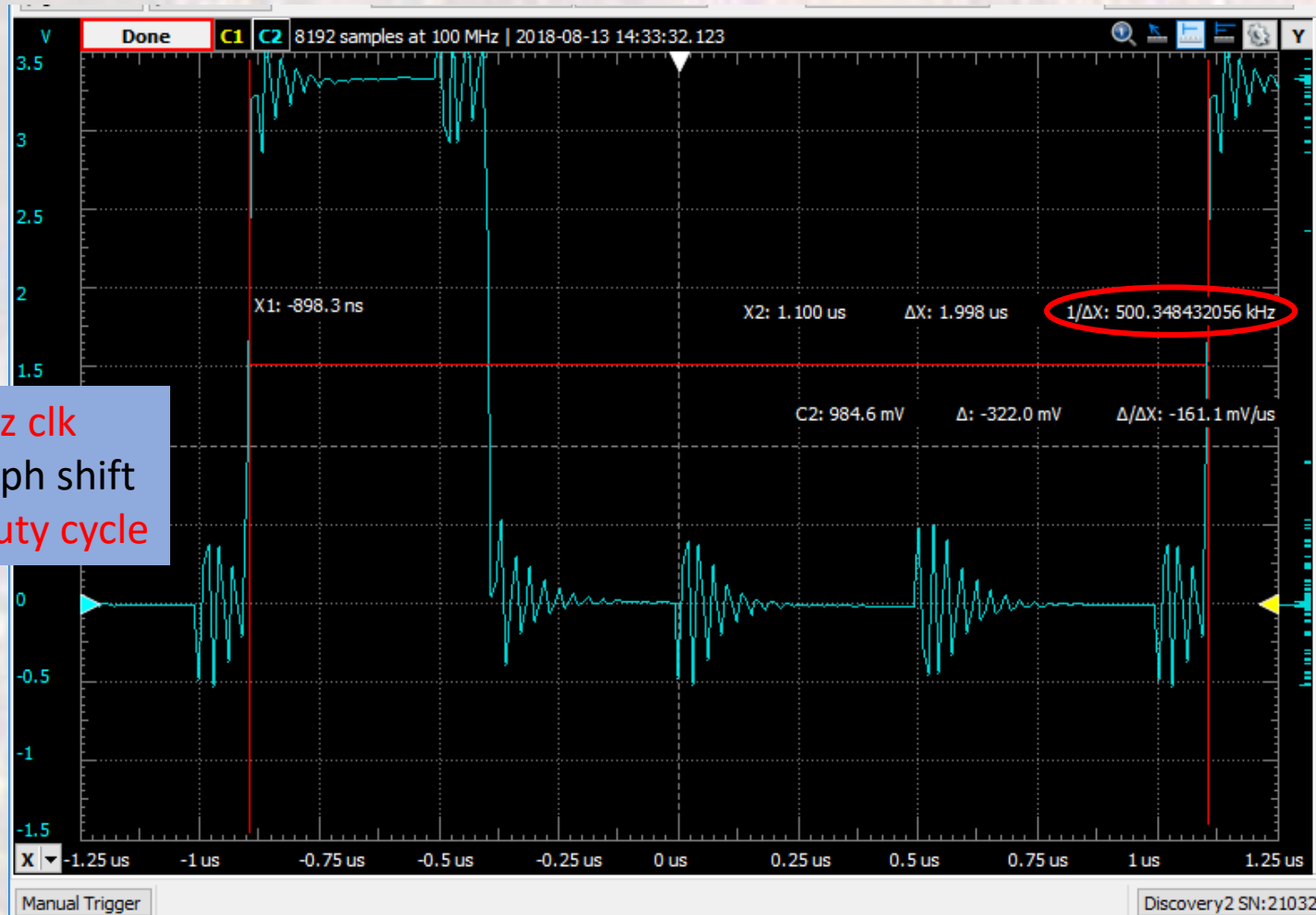
- MegaFunction AltPll – stand alone example 1



1MHz clk  
50% duty cycle

# MAX10 PLL MegaWizard

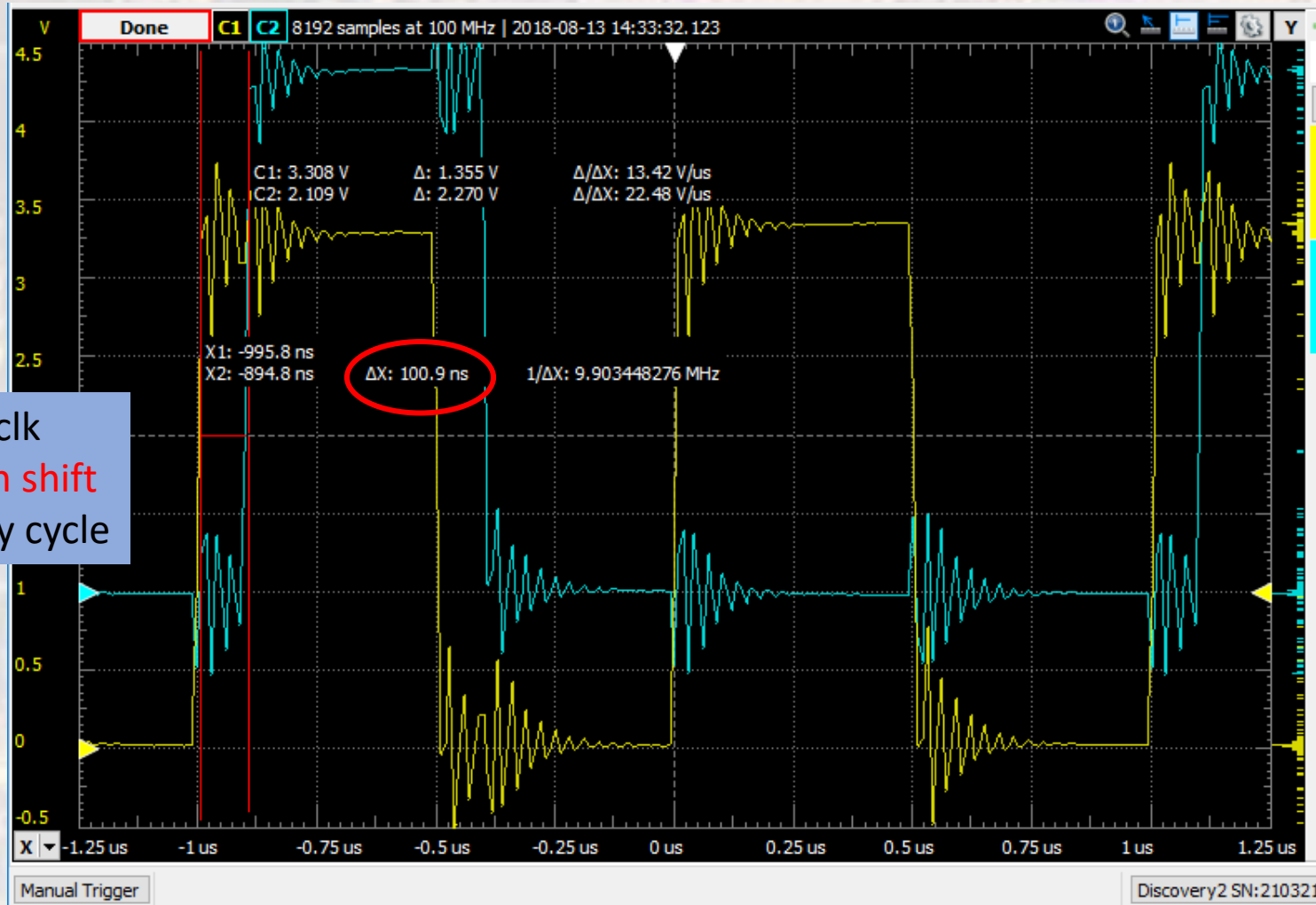
- MegaFunction AltPll – stand alone example 2



500kHz clk  
100ns ph shift  
25% duty cycle

# MAX10 PLL MegaWizard

- MegaFunction AltPll – stand alone example 3



500KHz clk  
100ns ph shift  
25% duty cycle