Generated Clocks Demo Script

Introduction

This demonstration provides high-level instructions on creating constraints for generated clocks and analyzing the timing reports of the generated clock.

Preparation:

- Required files: `$TRAINING_PATH/Generated_Clock/demo/KCU105/verilog`
- Required hardware: None

Generated Clocks

Action with Description	Point of Emphasis and Key Takeaway
• Launch Vivado® Design Suite 2021.2.	
• Open the **wave_gen.xpr** project from the following directory: `$TRAINING_PATH/Generated_Clock/demo/KCU105/verilog`	

• Open the synthesized design.	• You can open the synthesized design by using either:
	  • Flow Navigator
	  • Tcl Console
	  • Horizontal toolbar

• Open and view **wave_gen_timing.xdc**.	• How many clocks are created in the XDC? • wave_gen_timing has one created clock constraint on clk_pin_p.

Action with Description	Point of Emphasis and Key Takeaway
• Enter `report_clocks` in the Tcl Console.	• `report_clocks` returns a table showing all the clocks in the design. • How many clocks are returned from the `report_clocks` command? • Four total clocks are returned from the command. You can observe that there are three generated clocks that are propagated from one primary clock.

```
Clock Report


Attributes
  P: Propagated
  G: Generated
  A: Auto-derived
  R: Renamed
  V: Virtual
  I: Inverted

Clock              Period(ns)  Waveform(ns)   Attributes  Sources
clk_pin_p          3.333       {0.000 1.666}  P           {clk_pin_p}
clkfbout_clk_core  9.999       {0.000 5.000}  P,G,A       {clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKFBOUT}
clk_out1_clk_core  5.000       {0.000 2.500}  P,G,A       {clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKOUT0}
clk_out2_clk_core  5.161       {0.000 2.580}  P,G,A       {clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKOUT1}


==================================================
Generated Clocks
==================================================

Generated Clock   : clkfbout_clk_core
Master Source     : clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKIN1
Master Clock      : clk_pin_p
Edges             : {1 2 3}
Edge Shifts(ns)   : {0.000 3.333 6.666}
Generated Sources : {clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKFBOUT}

Generated Clock   : clk_out1_clk_core
Master Source     : clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKIN1
Master Clock      : clk_pin_p
Edges             : {1 2 3}
Edge Shifts(ns)   : {0.000 0.833 1.667}
Generated Sources : {clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKOUT0}

Generated Clock   : clk_out2_clk_core
Master Source     : clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKIN1
Master Clock      : clk_pin_p
Edges             : {1 2 3}
Edge Shifts(ns)   : {0.000 0.914 1.828}
Generated Sources : {clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKOUT1}
```

- What are generated clocks?
  - Clocks are generated **automatically** when a primary clock propagates to a cell that generates new clocks.
  - All these clocks can be described in XDC.

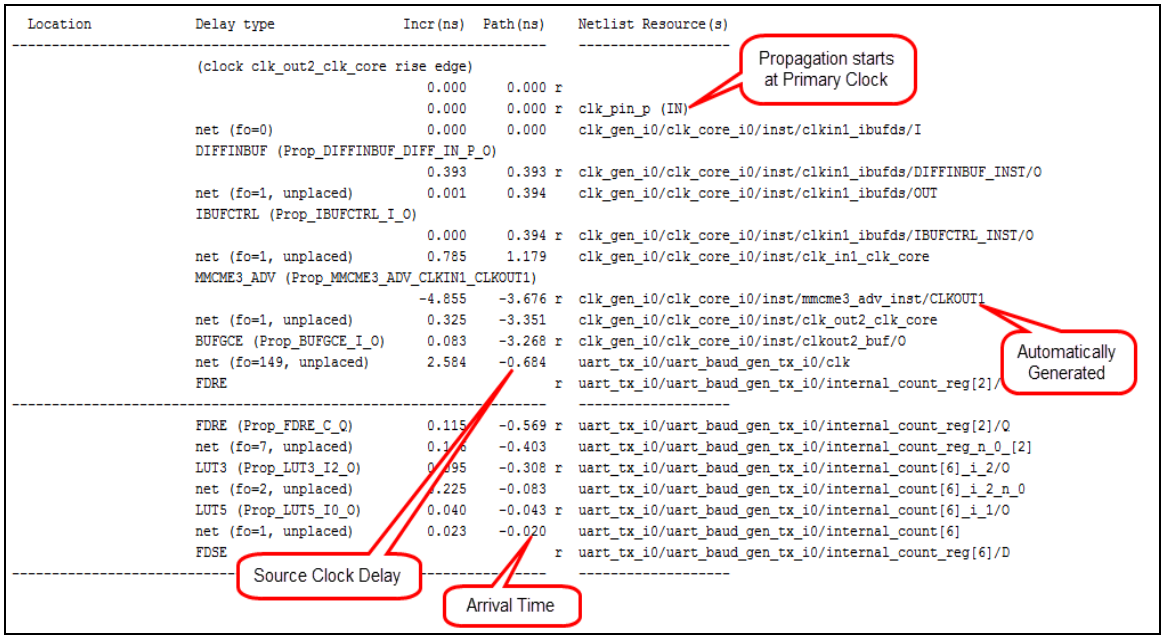
Action with Description	Point of Emphasis and Key Takeaway
• View the details of the generated clocks in the `report_clocks` command output.	• In addition to the summary of clocks, the `report_clocks` command shows how each generated clock is generated. • **Master Source** is the pin of the clock management cell that receives the input clock. • **Master Clock** is the clock that propagated to the Master Source. • The relationship between the master and generated clock is shown by: • Multiply By, Divided By, or Edges, and Edge Shift.
• Let's find out one of the generated clocks by using the master source as a pin. • Enter `get_clocks -of_objects [get_pins clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKOUT1]` in the Tcl console. • This command returns the generated clock clk_out2_clk_core.	• Clocks generated automatically by the tool are objects. • Like all objects, they should be queried by using the available commands. • The names of the clocks are not guaranteed to follow any naming convention and may vary between tool versions. • The clock should be obtained through an object to which it is attached.

Action with Description	Point of Emphasis and Key Takeaway
• Selecting the path with clk_out2_clk_core as the path group. • Here you can select any path, one such path is selected here as an example. • Enter `report_timing -from [get_pins uart_tx_i0/uart_baud_gen_tx_i0/internal_count_reg[2]/C] -to [get_pins uart_tx_i0/uart_baud_gen_tx_i0/internal_count_reg[6]/D]` in the Tcl console. • View the contents of the report.	• The requirement used for a path running on a generated clock is determined by the attributes of the generated clock. • The clock used for both the source and destination flip-flop is running on the clk_out2_clk_core, which is the output of the MMCM running at 193.75 MHz.

```
Slack (MET) :             4.340ns  (required time - arrival time)
  Source:                 uart_tx_i0/uart_baud_gen_tx_i0/internal_count_reg[2]/C
                            (rising edge-triggered cell FDRE clocked by clk_out2_clk_core {rise@0.000ns fall@2.580ns period=5.161ns})
  Destination:            uart_tx_i0/uart_baud_gen_tx_i0/internal_count_reg[6]/D
                            (rising edge-triggered cell FDSE clocked by clk_out2_clk_core {rise@0.000ns fall@2.580ns period=5.161ns})
  Path Group:             clk_out2_clk_core
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            5.161ns  (clk_out2_clk_core rise@5.161ns - clk_out2_clk_core rise@0.000ns)
  Data Path Delay:        0.664ns  (logic 0.250ns (37.651%)  route 0.414ns (62.349%))
  Logic Levels:           2  (LUT3=1 LUT5=1)
  Clock Path Skew:        -0.145ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    -1.061ns = ( 4.100 - 5.161 )
    Source Clock Delay      (SCD):    -0.684ns
    Clock Pessimism Removal (CPR):    0.232ns
  Clock Uncertainty:      0.071ns  ((TSJ^2 + DJ^2)^1/2) / 2 + PE
    Total System Jitter     (TSJ):    0.071ns
    Discrete Jitter          (DJ):    0.122ns
    Phase Error              (PE):    0.000ns
```

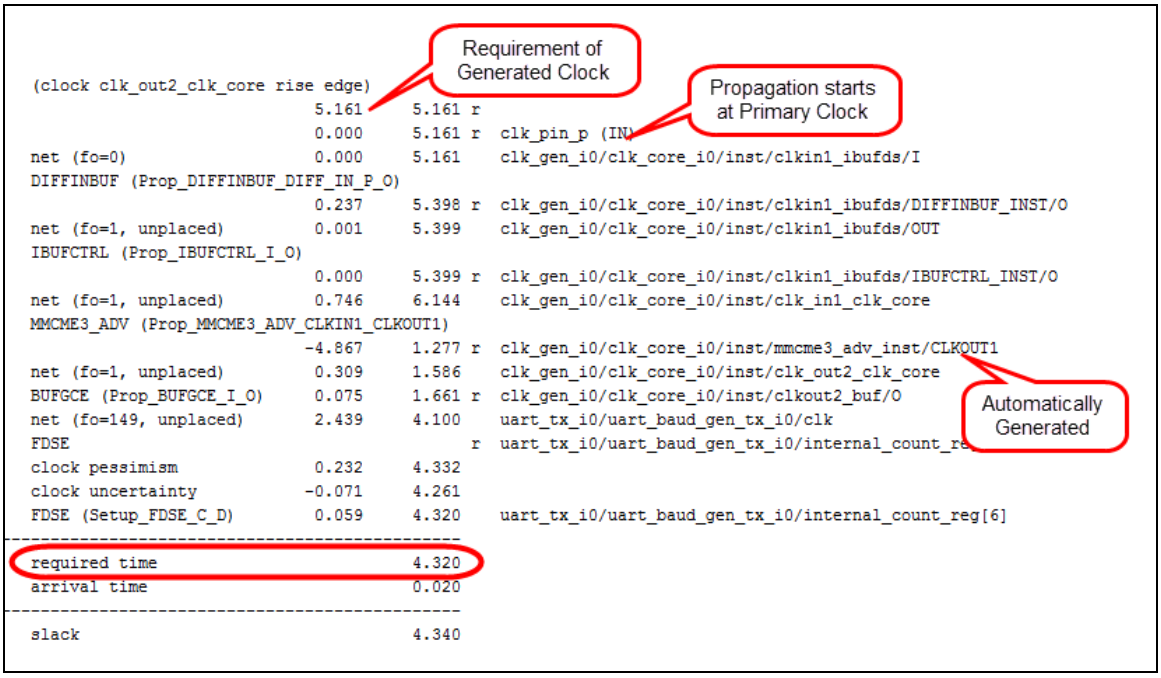
- **Note:** The timing numbers may vary depending on the version of the Vivado Design Suite and the OS.

Action with Description	Point of Emphasis and Key Takeaway
• View the source clock path and datapath delay from the report.	• Timing reports always start at primary clocks. • Propagates forward to generated clocks, and then on to the clocked elements. • The source clock path starts from clk_pin_p and propagates on to mmcm output CLKOUT1; i.e., the generated clock.



- **Note:** The timing numbers may vary depending on the version of the Vivado® Design Suite and the OS.

Action with Description	Point of Emphasis and Key Takeaway
• View the destination clock path timing.	• Like other setup checks, the destination clock delay starts at the next clock edge of the primary clock. • Propagates to the generated clock and on to the destination flip-flop. • The slack is required time – arrival time. • Note that a minus sign is added by the tool which may cancel the minus of a negative number.




```
(clock clk_out2_clk_core rise edge)
                            5.161     5.161 r
                            0.000     5.161 r  clk_pin_p (IN)
net (fo=0)                  0.000     5.161    clk_gen_i0/clk_core_i0/inst/clkin1_ibufds/I
DIFFINBUF (Prop_DIFFINBUF_DIFF_IN_P_O)
                            0.237     5.398 r  clk_gen_i0/clk_core_i0/inst/clkin1_ibufds/DIFFINBUF_INST/O
net (fo=1, unplaced)        0.001     5.399    clk_gen_i0/clk_core_i0/inst/clkin1_ibufds/OUT
IBUFCTRL (Prop_IBUFCTRL_I_O)
                            0.000     5.399 r  clk_gen_i0/clk_core_i0/inst/clkin1_ibufds/IBUFCTRL_INST/O
net (fo=1, unplaced)        0.746     6.144    clk_gen_i0/clk_core_i0/inst/clk_in1_clk_core
MMCME3_ADV (Prop_MMCME3_ADV_CLKIN1_CLKOUT1)
                           -4.867     1.277 r  clk_gen_i0/clk_core_i0/inst/mmcme3_adv_inst/CLKOUT1
net (fo=1, unplaced)        0.309     1.586    clk_gen_i0/clk_core_i0/inst/clk_out2_clk_core
BUFGCE (Prop_BUFGCE_I_O)    0.075     1.661 r  clk_gen_i0/clk_core_i0/inst/clkout2_buf/O
net (fo=149, unplaced)      2.439     4.100    uart_tx_i0/uart_baud_gen_tx_i0/clk
FDSE                                       r  uart_tx_i0/uart_baud_gen_tx_i0/internal_count_re
clock pessimism             0.232     4.332
clock uncertainty          -0.071     4.261
FDSE (Setup_FDSE_C_D)       0.059     4.320    uart_tx_i0/uart_baud_gen_tx_i0/internal_count_reg[6]
-------------------------------------------------------------------
required time                         4.320
arrival time                          0.020
-------------------------------------------------------------------
slack                                 4.340
```

- **Note:** The timing numbers may vary depending on the version of the Vivado Design Suite and the OS.

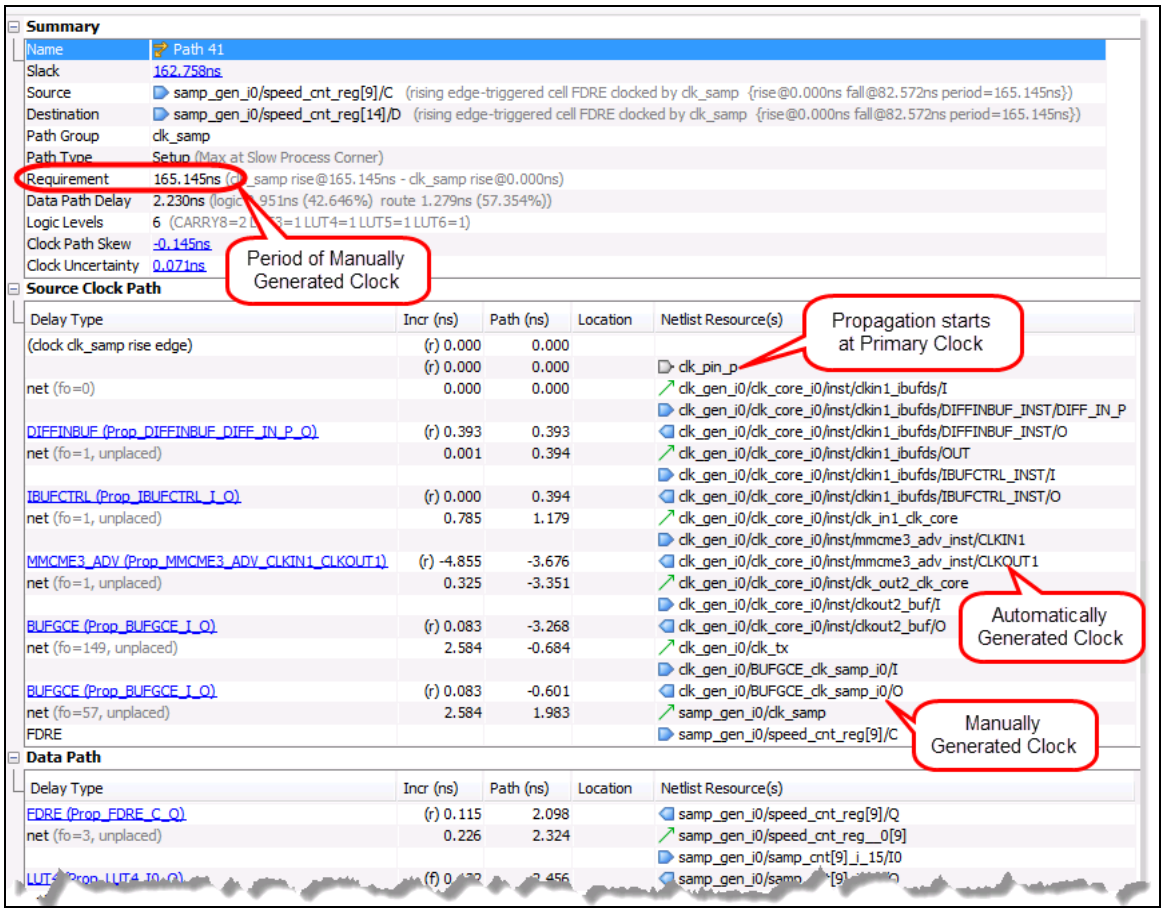
Action with Description	Point of Emphasis and Key Takeaway
• From the Netlist window, select **wave_gen** > **clk_gen_i0** and press <**F4**> to create a schematic. • Click **+** in clk_gen_i0 to expand. • Click **+** in clk_core_i0 to expand. • Click **+** in the inst module to expand it. • View the design schematic to analyze the logic for clk_samp (CLK). • Why should clk_samp (CLK) be constrained?	• Examining these above source and destination paths using the schematic. • A clock gate (BUFGCE/ BUFHCE) that is enabled periodically generates a decimated clock. • The period of the generated clock is N times the period of the input clock if the gate is activated one out of N clocks. • The timing engine cannot analyze the structure of the logic generating the CE and hence cannot automatically generate this clock. • clk_samp (CLK) is an example for this. You have to manually create a constraint for this generated clock.

Action with Description	Point of Emphasis and Key Takeaway

• Enter the following command in the Tcl Console to manually create a constraint:
 	
`create_generated_clock -name clk_samp -source [get_pins {clk_gen_i0/BUFGCE_clk_samp_i0/I}] -divide_by 32 [get_pins {clk_gen_i0/BUFGCE_clk_samp_i0/O}]`
	• The generated clock can be manually generated with the `create_generated_clock` command.
	• `create_generated_clock -name <name> -source <source> <relationship> <objects>`
	  • `<name>` is the user-assigned name for the new clock.
	  • `<source>` is a port or pin that is associated with the clock to use as the reference clock.
	  • `<relationship>` is one of a number of options for specifying the relationship between the source clock and the generated clock.
	  • `<objects>` is the list of pins/ports/nets to attach the new clock to.
	• Save the constraints. You can see the new constraint in wave_gen_timing.xdc to create generated clock *clk_samp*.

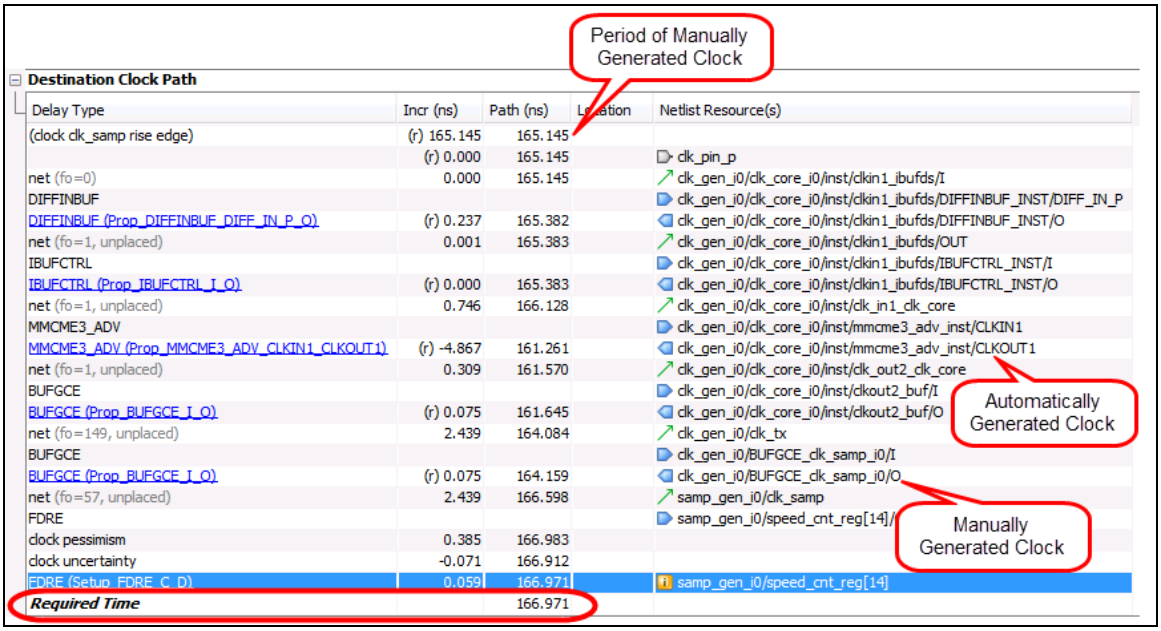
• Enter `report_clocks` in the Tcl Console.	• Observe that clk_samp has been added under the generated clocks.
• Run the Timing Summary report.	

Action with Description	Point of Emphasis and Key Takeaway
• Select **Intra clock paths** > **clk_samp** > **setup** in the Timing Summary report. • Double-click any path under this group (the most critical path, for example) to view its properties.	• The source clock delay always starts at the primary clock. • Propagates through all generated clocks. • In this case, propagates through both an automatically and manually generated clock. • The requirement is the period of the manually generated clock.



- **Note:** The timing numbers may vary depending on the version of the Vivado Design Suite and the OS.

Action with Description	Point of Emphasis and Key Takeaway
• View the Destination Clock Path section in the Timing Summary report.	• Like source clock delay, destination clock delay also starts at the primary clock and propagates through all generated clocks.



- **Note:** The timing numbers may vary depending on the version of the Vivado Design Suite and the OS.

• Close the project and exit the Vivado Design Suite.	

Summary

This demonstration illustrated how to create generated clock constraints and analyze the timing reports of the generated clocks.

References:

- Supporting materials
  - *Vivado Design Suite User Guide: Using Constraints* (UG903)
  - *Vivado Design Suite User Guide: Design Analysis and Closure Techniques* (UG906)