High precision Time to Digital Convert Measurement Application Guide

(AN04016, V1.0)

(26.03.2022)

Shenzhen Pango Microsystems Co., Ltd.

Revisions History

Document Revisions

Version	Date of Release	Revisions
V1.0	26.03.2022	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
CLM	Configurable Logic Module
LUT	LookUp Table
GTP	Generic Technology Primitive
TDC	Time to Digital Convert

Table of Contents

Tables

Figures

Chapter 1 Overview

1.1 Introduction

This document mainly introduces an application scheme for implementing TDC based on the carry chain of Logos2 family devices. Its main content includes

the function list, design architecture, interface definitions, interface timing, supported devices and reference design.

1.2 Main Functions

The main functions supported include:

- FPGA transmits an pulse start signal, receives a stop signal, and measures the time interval between the rising edges of these two pulses. The measuring accuracy at room temperature is approximately 70ps (mainly depending on the precision of the carry chain).
- A column in one region supports 50 CLMs cascaded.

1.3 Design Information

Table 1-1 Design Information

TDC Application Reference Design	
Supported Devices	PG2L100H
Supported User Interface	Custom
Provided Design Files	
Design File	Verilog files
Reference Designs	Verilog files
Simulation File	Verilog files
Constraint File	fdc file
Development Tools	
Design Tools	PDS development suite Pango Design Suite 2021.4
Simulation Tool	Third-party simulation tools

1.4 Resource Usage

Table 1-2 Resource Usage Rate

Device	DRM	FF	LUT	PLL
PG2L100H	0	349	408	1

Chapter 2 Function Description

2.1 TDC Design Architecture

The overall design block diagram of TDC is shown in Figure 2-1. The main functional modules include clock module (clk_gen), start signal control module (start_sig_ctl), carry chain module (carrychain), and total time calculation module (time_compute).

The clock module generates the system clock sys_clk through PLL frequency multiplication. The start_sig_ctl module generates a start impulse signal upon receiving the trigger signal and transmits it out from FPGA to the remote device and time_compute module. At this time, the time_compute module starts counting. Upon receiving the start signal, the remote device immediately transmits a stop signal. After receiving the stop signal, the FPGA waits for the carry chain module to calculate the time interval ΔT between the rising edge of the stop signal and the first rising edge of the next sys_clk clock. The time_compute module stops counting upon receiving the stop signal, and calculates the interval between the rising edge of the start signal and that of the stop signal through $N * T_0 - \Delta T$, as shown in Figure 2-7.

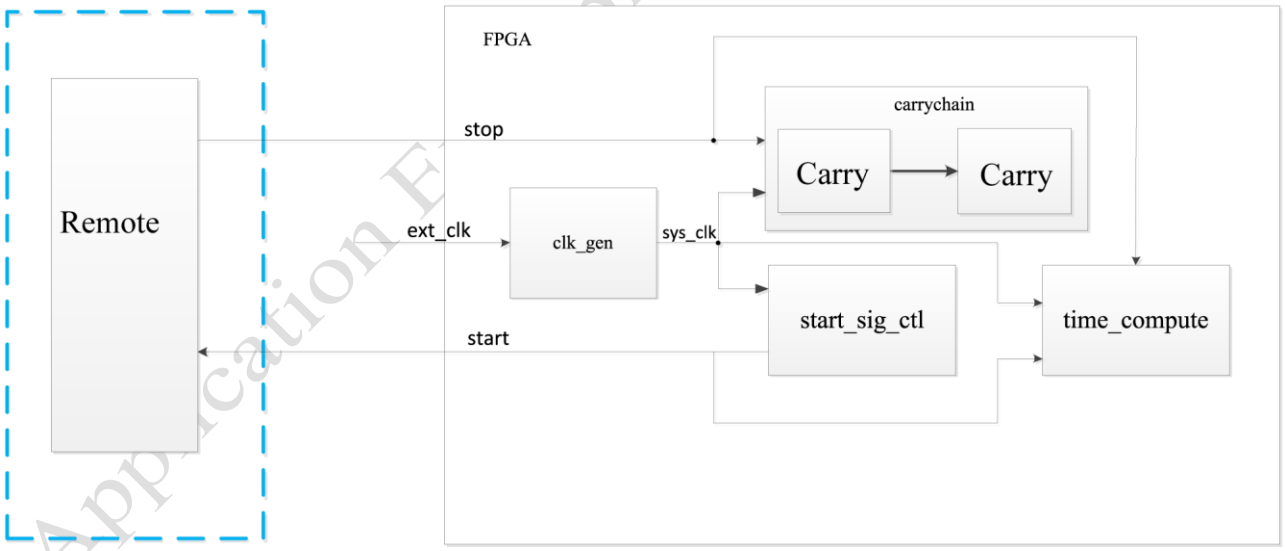

Figure 2-1 Overall Design Block Diagram of TDC

2.2 Module Function Introduction

2.2.1 clk_gen Module

The external clock generates a clock sys_clk with the same frequency and different phases through PLL frequency multiplication, with a clock frequency of 250MHz.

2.2.2 start_sig_ctl Module

This module generates impulse signals. When the trigger signal i_start_trig is at a high impulse, this module generates an start pulse signal.

2.2.3 carrychain Module

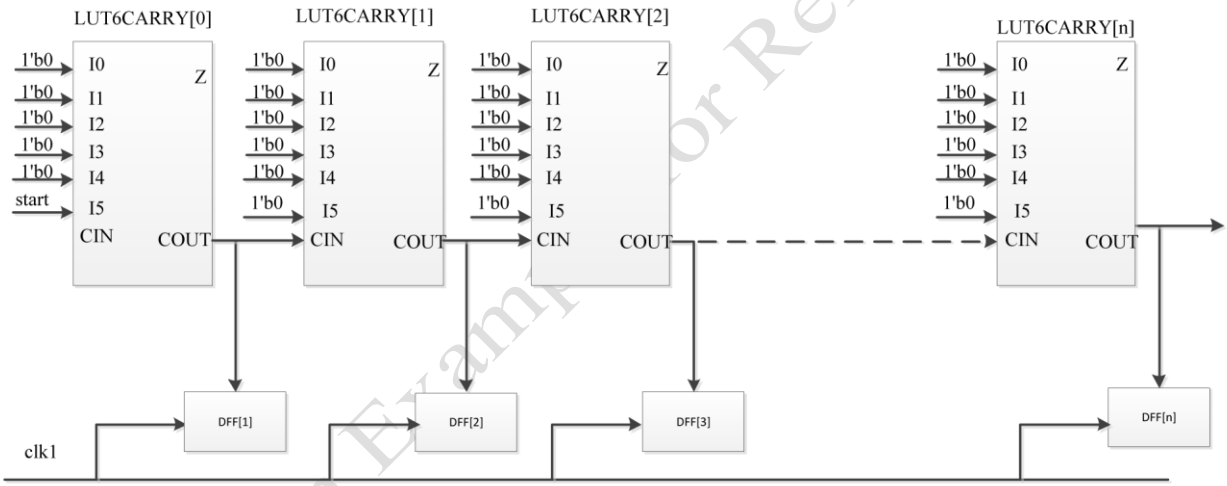

Figure 2-2 GTP_LUT6CARRY Cascade Diagram

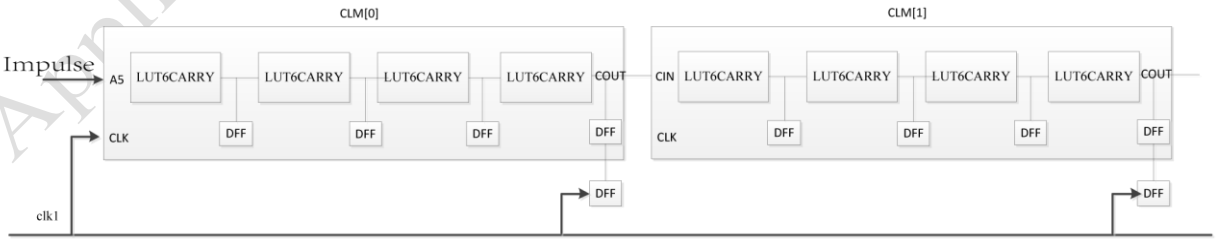

Figure 2-3 CLM Cascade

This module calculates the time interval between the rising edge of the impulse signal and the first rising edge of the next sys_clk through carry chain. Taking one CLM as one stage, there are 50 stages when cascading one column in a region. Except for the first stage, each stage has the same configuration, ensuring that each stage has the same time delay.

One CLM includes 4 LUT6CARRYs, which are cascaded in series as shown in Figure 2-3. Through a register,LUT6CARRYs in each stage are output and grouped in sets of 4. The output of the fourth LUT6CARRY of each group is taken as the output of each stage of the CLM. There are 200 LUT6CARRYs in total. The output of each stage of CLM is stored in another register. Calculate the total number of such registers with the value of 1, and multiply the total number by the delay time of each stage to get ΔT.

Because CIN can only be linked with COUT, the start signal enters from the first stage's I5, with I0-I4 configured as 0 and INIT parameter configured as 0, as shown in Figure 2-4. For each subsequent stage, the configuration is the same, with I0-I5 configured as 0 and INIT parameter configured as 1, as shown in Figure 2-5.

```
GTP_LUT6CARRY #(
    .INIT         (64'h0000_0000_0000_0000_0000_0000_0000_0000),
    .I5_TO_CARRY ("TRUE"),
    .I5_TO_LUT    ("TRUE")
) LUT6CARRY_inst0(
    .COUT         (c0_cout_w[i] ),// OUTPUT
    .Z            (c0_z_w[i]    ),// OUTPUT
    .CIN          (             ),// INPUT
    .I0           (1'b0         ),// INPUT
    .I1           (1'b0         ),// INPUT
    .I2           (1'b0         ),// INPUT
    .I3           (1'b0         ),// INPUT
    .I4           (1'b0         ),// INPUT
    .I5           (i_start      ) // INPUT
);
```

Figure 2-4 Instantiation Diagram of the First Stage of Carry Chain

```
GTP_LUT6CARRY #(
    .INIT         (64'h0000_0000_0000_0000_0000_0000_0000_0001),
    .I5_TO_CARRY ("TRUE"),
    .I5_TO_LUT    ("TRUE")
) LUT6CARRY_inst1(
    .COUT         (c0_cout_w[i]     ),// OUTPUT
    .Z            (c0_z_w[i]        ),// OUTPUT
    .CIN          (c0_cout_w[i-1]   ),// INPUT
    .I0           (1'b0             ),// INPUT
    .I1           (1'b0             ),// INPUT
    .I2           (1'b0             ),// INPUT
    .I3           (1'b0             ),// INPUT
    .I4           (1'b0             ),// INPUT
    .I5           (1'b0             ) // INPUT
);
```

Figure 2-5 Instantiation Diagram of Multiple Subsequent Stages After the First Stage of Carry Chain

Constrain the location of the carry chain, as indicated by the arrow in Figure 2-6. Constrain 50 CLMs to the same region. Locate the first CLM in a certain column within the same region, with the order from bottom to top, and constrain the first CARRY_CHAIN to the CLM's FYA. The software will automatically constrain the entire carry chain by column from bottom to top as well as the registers that store the carry chain to the same CLM, without manual operation. It should be noted that the register of the carry chain cannot be used in decision logic. Such as if(c0_cout_r0!=0), the results compiled by the tool may occupy LUT resources, making it impossible to constrain the registers with the carry chain to the same CLM, causing inconsistencies in the delay of all carry chain outputs to the registers.

The constraint command of carry chain is as follows:

define_attribute {i:carrychain_inst.CARRY_CHAIN[0].genblk1.LUT6CARRY_inst0}{PAP_LOC}{CLMA_207_612:FYA}

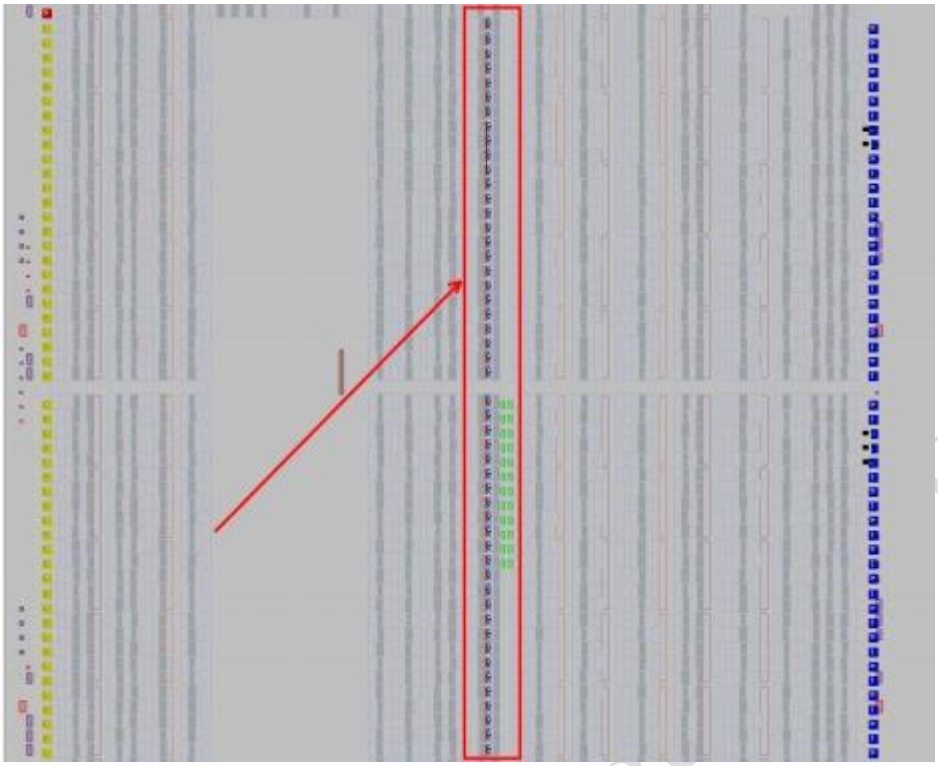

Figure 2-6 Diagram of Carry Chain Location Constraint

2.2.4 time_compute Module

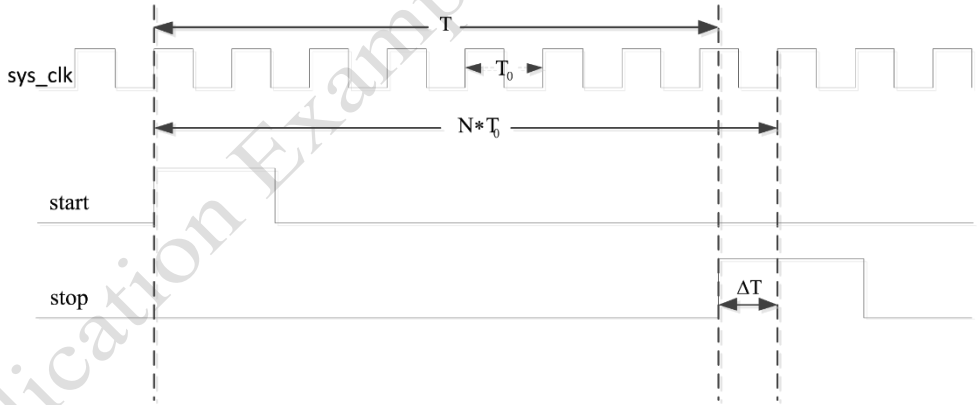

Figure 2-7 Diagram of Impulse Signal Interval Calculation Timing

The calculation timing for the interval "T" between the rising edges of the `start` signal and the `stop` signal is shown in Figure 2-7. Counting starts when a high impulse is detected on the start signal, and stops when the carry chain module figures out the fine measurement time ΔT. Calculate N * T0. T = N * T0 - ΔT.

2.3 Interface List

Table 2-1 tdc_top Module Interface List

Signal Name	Input/Output	Bit width	Description
ext_clk	Input	1	External clock signal, 20MHz
rst_n	Input	1	System reset signal, active-low
i_stop	Input	1	Stop impulse signal
i_start_trig	Input	1	Trigger signal for transmitting start signal, active at a high impulse
o_start	Output	1	Start impulse, with three sys_clk cycles at a high level
o_carrychain_full	Output	1	Indicator for holding allCLM outputs as 1 until the nextstop signal is received
o_carrychain_empty	Output	1	Indicator for all LUT6CARRY outputs as 0. Pull down when stop is at a high level, and pull high when stop is at a low level
o_total_time	Output	32	Time interval between the rising edge of start signal and that of stop signal, in ps

Table 2-2 clk_gen Module Interface List

Signal Name	Input/Output	Bit width	Description
clk	Input	1	External clock signal, 20MHz
rst_n	Input	1	System reset signal, active-low
sys_clk	Output	1	System clock output, 250MHz. Users can modify the PLL output clock frequency as needed.

Table 2-3 start_sig_ctl Module Interface List

Signal Name	Input/Output	Bit width	Description
clk	Input	1	System clock signal, 250MHz
rst_n	Input	1	System reset signal, active-low
i_start_trig	Input	1	Trigger signal for transmitting start signal, active at a high impulse
o_start	Output	1	Impulse signal output, with three sys_clk impulses

Table 2-4 carrychain Module Interface List

Signal Name	Input/Output	Bit width	Description
clk	Input	1	System clock signal, 250MHz
rst_n	Input	1	System reset signal, active-low
i_stop	Input	1	Stop impulse signal
i_total_comp_done	Input	1	Interval time T, calculation completion signal, active-high
o_carrychain_full	Output	1	Indicator for holding allCLM outputs as 1 until the nextstop signal is received
o_carrychain_empty	Output	1	Indicator for all LUT6CARRY outputs as 0. Pull down when stop is at a high level, and pull high when stop is at a low level
o_fine_time	Output	14	Fine measurement time ΔT, in ps
o_fine_comp_done	Output	1	Fine measurement time completion signal, active-high

Table 2-5 time_compute Module Interface List

Signal Name	Input/Output	Bit width	Description
clk	Input	1	System clock signal, 250MHz
rst_n	Input	1	System reset signal, active-low
i_start	Input	1	Start impulse, with three sys_clk cycles at a high level
i_stop	Input	1	Stop impulse signal
i_fine_time	Input	14	Fine measurement time ΔT, in ps
i_fine_comp_done	Input	1	Fine measurement time completion signal, active-high
o_total_time	Output	32	Time interval between the rising edge of start signal and that of stop signal, in ps
o_total_comp_done	Output	1	Interval time T, calculation completion signal, active-high

2.4 Parameter Definitions

Table 2-6 Parameter Definitions

Parameter	Description
CRY_NUM	Number of LUT6CARRYs cascaded, 200by default, covering a column of CLMs in a region
CRY_SDELAY	One-stage delay parameter of the carry chain, 71ps by default
CARYYCHAIN_FULL	Parameter of setting all CLM outputs of a carry chain as 1, 50 by default
start_sig_ctl module INTERVAL_TIME	Counter for the interval from receiving the trigger signal to transmitting the start signal, 100 by default

2.5 File Directory

```
|—docs                          //Design document
|—pnr                           //Project directory
|—|—ipcore                      //IP file
|—|—sim                         //Simulation file
|—|—tdc_demo.fdc                //Project constraint file
|—|—tdc_demo.pds                //PDS project file
|—src                           //Project directory
|—|—tdc_top                     //Project top-level RTL file
|—|—clk_gen                     //Clock generation module RTL file
|—|—start_sig_ctl               //Impulse generation module RTL file
|—|—carrychain                  //Carry chain module RTL file
|—|—time_compute                //Time interval calculation RTL file

|—example_design                //Test case
|—|—bench                       //Simulation top-level
|—|—pnr                         //Project directory
|—|—|—ipcore                    //IP file
|—|—|—sim                       //Simulation file
|—|—|—example_design.fdc        //Project constraint file
|—|—|—example_design.pds        //PDS project file
|—|—src                         //Project directory
|—|—|—tdc_demo_top              //Project top-level RTL file
|—|—|—tdc_top                   //TDC top-level RTL file
|—|—|—clk_gen                   //Clock generation module RTL file
|—|—|—start_sig_ctl             //Impulse generation module RTL file
|—|—|—carrychain                //Carry chain module RTL file
|—|—|—time_compute              //Time interval calculation RTL file
|—|—|—test_rev_stop             //Stop delay RTL file
|—|—synthesize                  //Synthesis directory
|—|—|—tdc_top_syn.fic           //Debug file
```

Figure 2-8 File Directory

Chapter 3 Reference Design

3.1 Reference Function Design

The reference design block diagram is shown in Figure 3-1. Compared with Figure 2-1 overall design block diagram of TDC, the main difference is that, for the reference design,a test_rev_stop module is designed in the FPGA to replace the remote device. This module receives the start signal, and then forwards the stop signal after a period of time. clk_gen generates two 250MHz clocks with the same frequency but different phases. The carrychain module uses the sys_clk clock to keep a certain phase relationship with the stop signal, and calculates the time difference ΔT. test_clk changes through the dynamic phase adjustment method.In the reference design, the VCO frequency is 1000Mhz, the dynamic phase adjustment precision is Tvco/64, i.e. 15.625ps.

This scheme can test the time difference between two impulse signals, T.

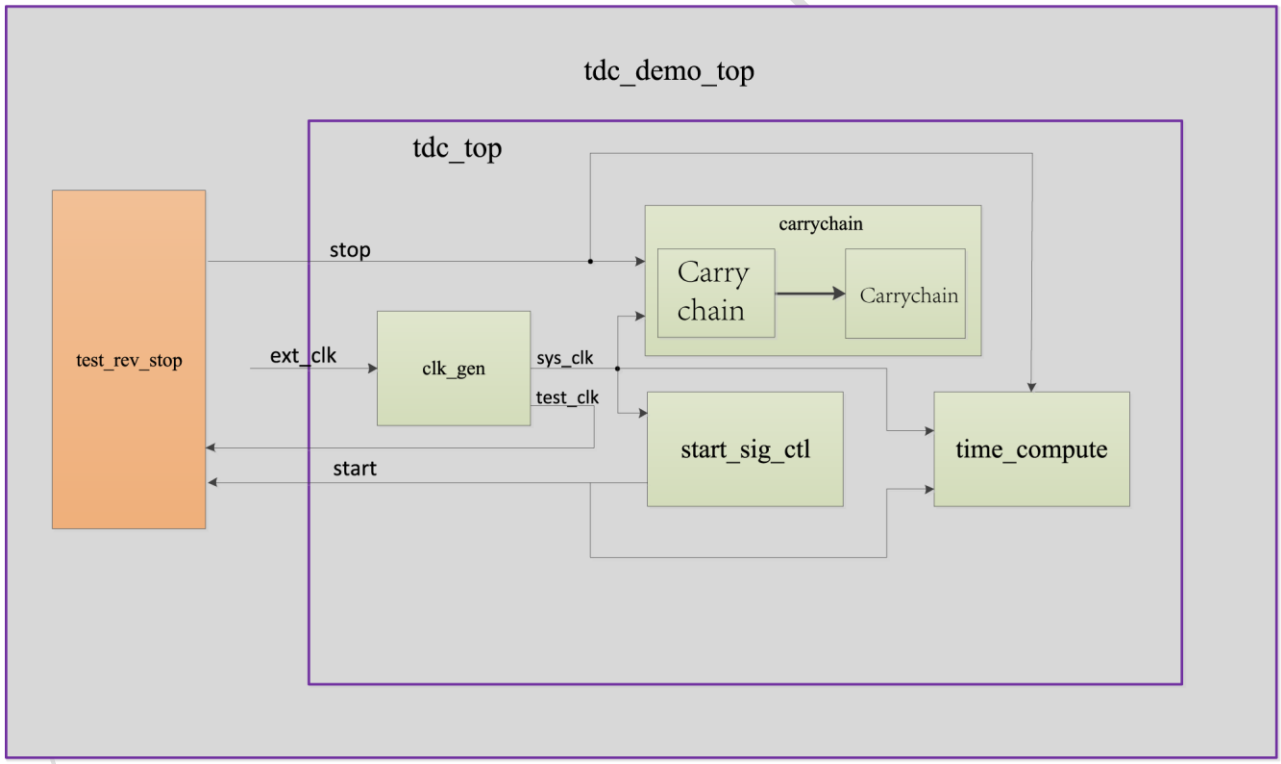

Figure 3-1 Reference Design Block Diagram

3.2 Reference Design Interface List

This table only lists the ports excluded in 2.3.

Table 3-1 Reference Design Interface List

Signal Name	Input/Output	Bit width	Description
test_rev_stop module			
o_stop_delay	Output	1	stop signal output. This signal is forwarded after a period of time after the start signal is received.
clk_gen module			
sw	Input	1	DIP switch, which controls the phase of the clock module test_clk, active-high
dsp_ctrl	Input	1	DIP switch, which controls the phase adjustment direction 1'b1: decrease; 1'b1: increase;
o_test_clk	Output	1	First PLL output clock, 250MHz, used by the impulse generation module, with a phase difference of 180 degrees with sys_clk
o_sys_clk	Output	1	Second PLL output clock, 250MHz, used by the impulse generation module
o_cfg_done	Output	1	PLL phase configuration completion signal
o_delay_step	Output	8	Indicator for the number of PLL phase adjustments, counting 1 for each adjustment

3.3 Reference Design Simulation

Conduct simulation using modelsim, with the simulation file of tdc_demo_top_tb. The simulation waveform is shown in the figure below:

1. tdc_top module simulation

After the DIP switch i_sw is active, wait for a period of time to finally get the time T as 45500ps.

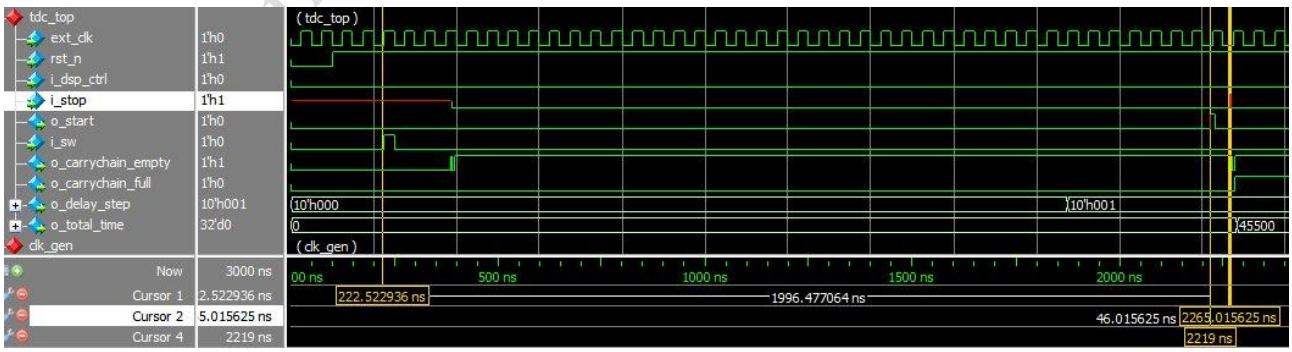

Figure 3-2 tdc_top Simulation Waveform

2. clk_gen module simulation

During simulation, the clk_gen module's CFG_WAIT_TIME parameter should be set to 'd14; when compiling the bit file, change this parameter to the default value of 'd10_00014. As shown below, set the DIP switch sw to 1. After the dynamic phase configuration completes, add delay_step by 1.

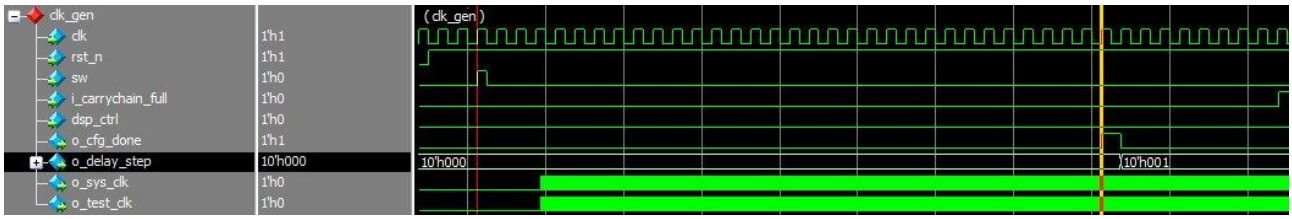

Figure 3-3 clk_gen Module Simulation Waveform

3. start_sig_ctl module simulation

As shown in the figure, after the trigger signal high impulse occurs, the module transmits the start signal after 392ns.

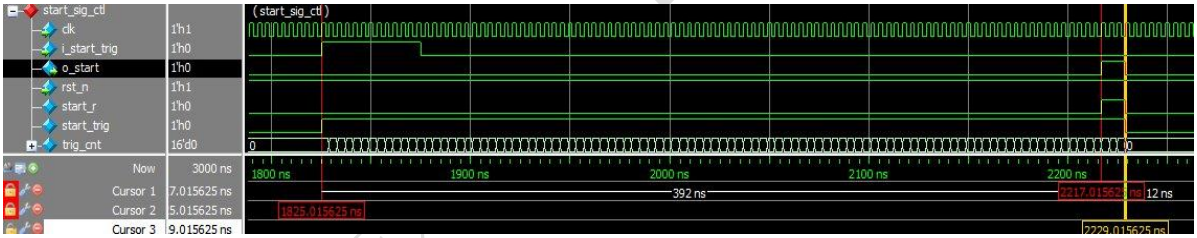

Figure 3-4 sig_ctl Module Simulation Waveform

4. carrychain module simulation

The carrychain module simulation waveform is shown in the figure below. After receiving the stop signal, calculate the fine_time (i.e., ΔT).

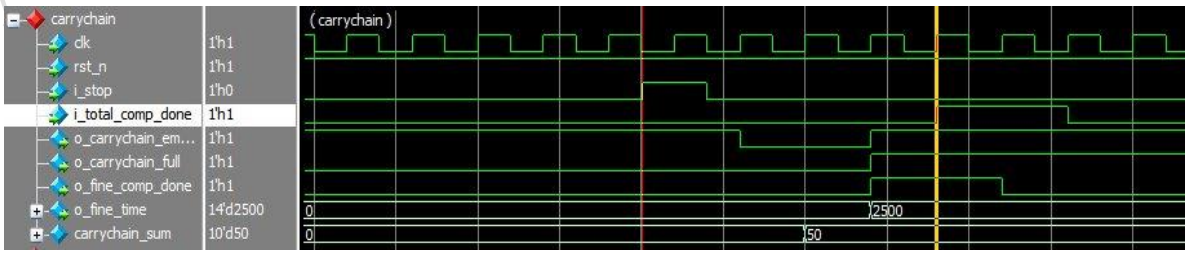

Figure 3-5 carrychain Module Simulation Waveform

5. time_compute module simulation

As shown in the figure below, 12 sys_clk cycles passed from the time when a start signal is transmitted to the time when a stop signal at a high level is detected, that is 48ns.

T=48000-2500ps=45500ps.

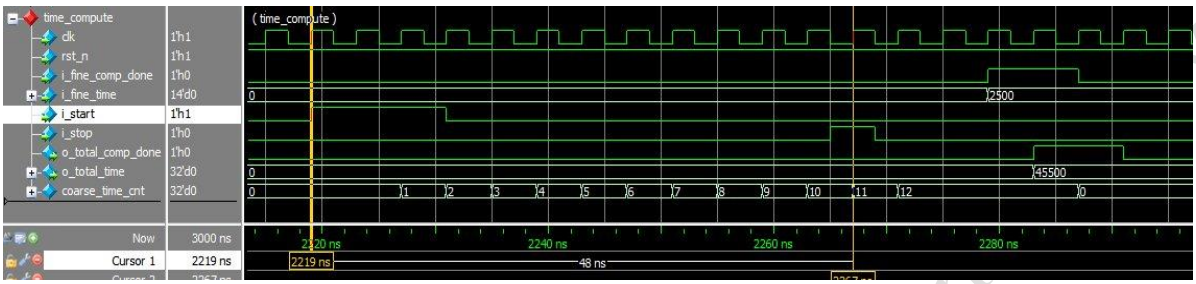

Figure 3-6 time_compute Module Simulation Waveform Diagram

3.4 Reference Design On-board Verification

3.4.1 Verification Method

Set the phase difference between sys_clk and test_clk through PLL static phase adjustment, to generate a certain phase difference between the stop signal and sys_clk.Perform testing through capturing signal by Fabric Debugger, as shown in Figure 3-7.Record the value of carry_chain_sum when delay_step equals 1. If the value of carry_chain_sum is 0, adjust the phase difference between

sys_clk and test_clk to regenerate a bit, ensuring that the value of carry_chain_sum is not 0. After adjustment, record the data of each test according to Table 3-2.Delay for each carry chain stage is (fine_time2 - fine_time2)/((sum2 - sum1).

Values at specific moments can be captured by setting the trigger condition. For example, when setting carry_chain_sum to 30, the software will capture the value at this moment.

Trigger Unit	Function	Value	Radix
TU0(TriggerPort0)	==	XX_XXXX_XXXX_XXXX_XXXX_XX...	Bin
nt_ext_clk		X	
nt_o_carrychain_empty		X	
nt_o_carrychain_full		X	
nt_o_start		X	
stop_delay		X	
nt_o_delay_step		XX_XXXX_XXXX	
nt_o_total_time		XXXX_XXXX_XXXX_XXXX_XXXX_...	
tdc_top_inst/time_compute_inst/coarse_time_cnt		XXX_XXXX_XXXX_XXXX_XXXX_X...	
tdc_top_inst/carrychain_inst/carrychain_sum		XXX_XXXX	
tdc_top_inst/fine_time		XX_XXXX_XXXX_XXXX	
stop_delay_n1		X	
tdc_top_inst/carrychain_inst/c0_cout_r1		XX_XXXX_XXXX_XXXX_XXXX_XX...	

Figure 3-7 Main debug Signals

3.4.2 Verification Process

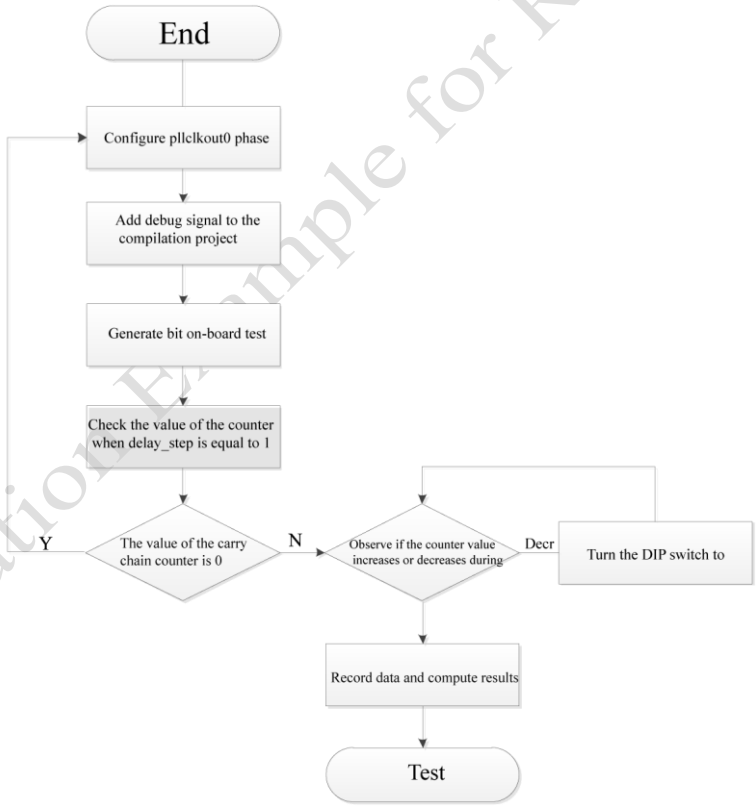

Figure 3-8 Verification Flowchart

Note: Users can perform verification according to their own designs; this flowchart just provides a verification method for reference.

The verification process is shown in Figure 3-8.

Configure PLL clkout0 phase: Use Advance mode for configuration, as shown in Figure 3-9. Configure the division ratio to make the VCO frequency equal to 1000MHz, then the dynamic phase adjustment precision is Tvco/8=15.625ps. Enable the dynamic phase adjustment port, and configure clkout0 static phase adjustment division ratio, causing a phase difference between clkout0 and clkout1.

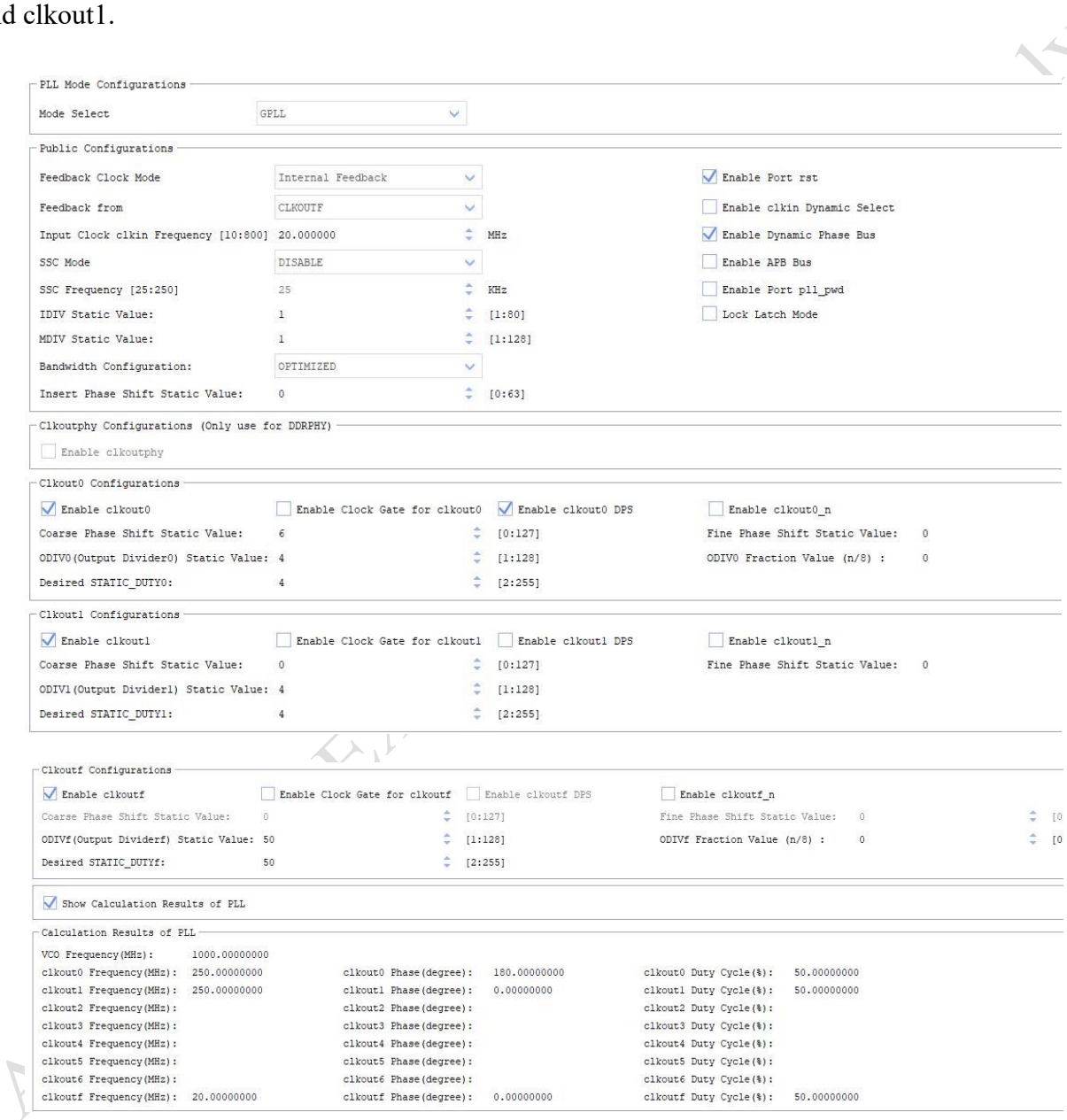

Figure 3-9 PLL Configuration

Add Debugsignals to the compilation project. The tdc_top_syn.fic files of test case can be directly added to the project:

1. o_delay_step: Used to observe the number of PLL phase adjustments.
2. carry_chain_sum: Used to observe at which stage the carry chain is set to 1.
3. fine_time: Carry chain test time.
4. o_total_time: Total test time.
5. coarse_time_cnt: Coarse measurement time counter. For the reference design system clock frequency of 250MHz, the coarse measurement time is coarse_time_cnt*1000*4, in ps.
6. c0_cout_r1: Used to observe the CLMs where the final-stage LUT6CARRY output is set to 1.
7. o_start: Start impulse, set to trigger on the rising edge. Toggle the switch from 0 to 1, and record the step value and the corresponding counter value at this moment.

When the value of the carry chain counter is 0: Adjust carrychain_sum to non-zero.

Observe if the counter value increases or decreases during adjustment: Due to the different traces between the clock and impulse signal, the actual phase difference between the impulse signal and sys_clk is not the same as that between sys_clk and test_clk. Therefore, when dynamically adjusting the clkout0 phase, it is necessary to observe whether the counter value increases or decreases. If it decreases, adjust the DIP switch controlling i_dsp_ctrl signal to adjust the direction of clk0 phase adjustment, ensuring that the counter accumulates during phase adjustment.

Record data: Record the data of each test according to Table 3-2. The delay for each stage of the carry chain is (fine_time2 - fine_time1) / (sum2 - sum1).

3.4.3 Verification Results

1. On-board test results

Set the carry chain delay parameter CRY_SDELAY to 71, and the test results are shown in Table 3-2. Use fine time to calculate the average delay of the carry chain, (1420 - 1207) / (20 - 17) = 71. For a given bit, a total of 5 sets of data are tested, and the average values are all 71ps, consistent with the set value, indicating a normal logical function. The signals captured in the 1st and 3rd groups are shown in Figure 3-10 and Figure 3-11.

Table 3-2 Carry Chain Delay Test Data

No.	delay step	carrychain sum	coarse time cnt	fine time	total time	Average Delay
1	1	17	12	1207	46193	
2	16	20	12	1420	46580	71ps

No.	delay step	carrychain sum	coarse time cnt	fine time	total time	Average Delay
3	65	30	13	2130	49870	71ps
4	109	40	13	2840	49160	71ps
5	154	50	13	3550	48450	71ps

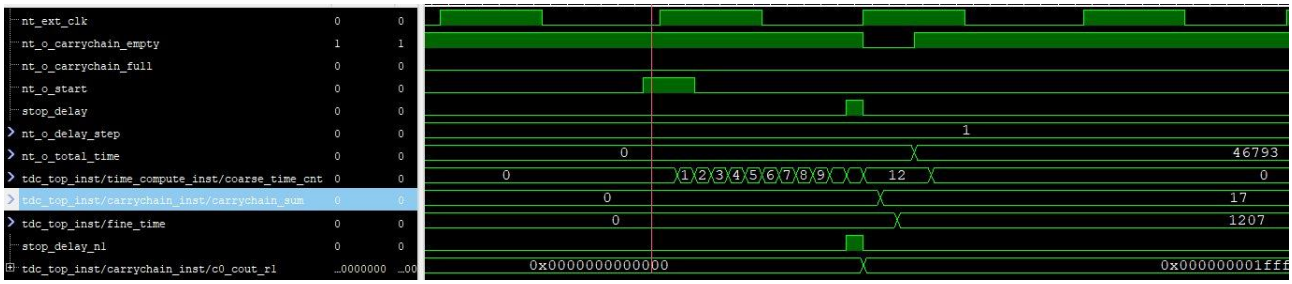

Figure 3-10 Debug Signals for Carry Chain Counter Value 17

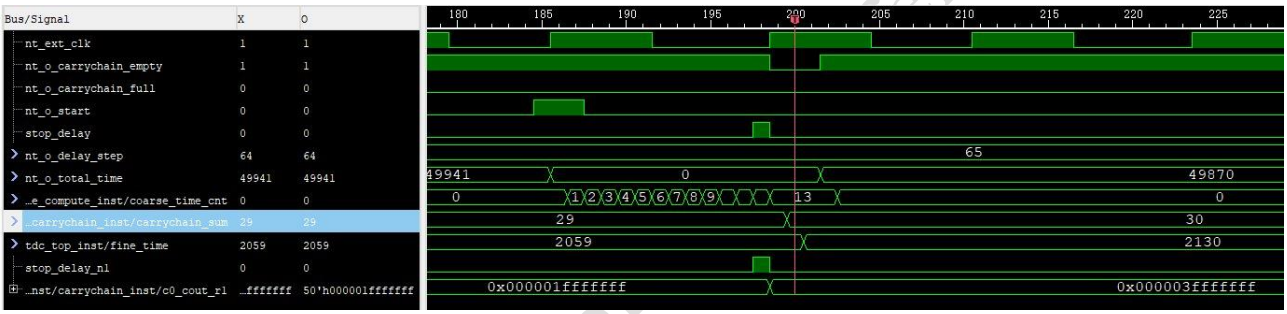

Figure 3-11 Debug Signals for Carry Chain Counter Value 30

2. Timing report

Under the report_timing directory, find the carrychain_top.rtr file to view the carry chain timing report. Figure 3-12 shows the carry chain delay by Slow Corner timing analysis, with a delay of 85ps between two stages. Figure 3-13 shows the carry chain delay by Fast Corner timing analysis, with a delay of 56ps between two stages.

	net (fanout=8)	0.247	5.944	nt_o_start
CLMA_207_612/COUT	td	0.184	6.128 f	carrychain_inst/c0_cout_r0[3]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.128	carrychain_inst/c0_cout_w [3]
CLMA_207_618/COUT	td	0.085	6.213 f	carrychain_inst/c0_cout_r0[7]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.213	carrychain_inst/c0_cout_w [7]
CLMA_207_624/COUT	td	0.085	6.298 f	carrychain_inst/c0_cout_r0[11]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.298	carrychain_inst/c0_cout_w [11]
CLMA_207_630/COUT	td	0.085	6.383 f	carrychain_inst/c0_cout_r0[15]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.383	carrychain_inst/c0_cout_w [15]
CLMA_207_636/COUT	td	0.085	6.468 f	carrychain_inst/c0_cout_r0[19]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.468	carrychain_inst/c0_cout_w [19]
CLMA_207_642/COUT	td	0.085	6.553 f	carrychain_inst/c0_cout_r0[23]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.553	carrychain_inst/c0_cout_w [23]
CLMA_207_648/COUT	td	0.085	6.638 f	carrychain_inst/c0_cout_r0[27]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.638	carrychain_inst/c0_cout_w [27]
CLMA_207_654/COUT	td	0.085	6.723 f	carrychain_inst/c0_cout_r0[31]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.723	carrychain_inst/c0_cout_w [31]
CLMA_207_660/COUT	td	0.085	6.808 f	carrychain_inst/c0_cout_r0[35]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.808	carrychain_inst/c0_cout_w [35]
CLMA_207_666/COUT	td	0.085	6.893 f	carrychain_inst/c0_cout_r0[39]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.893	carrychain_inst/c0_cout_w [39]
CLMA_207_672/COUT	td	0.085	6.978 f	carrychain_inst/c0_cout_r0[43]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	6.978	carrychain_inst/c0_cout_w [43]
CLMA_207_678/COUT	td	0.085	7.063 f	carrychain_inst/c0_cout_r0[47]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	7.063	carrychain_inst/c0_cout_w [47]
CLMA_207_684/COUT	td	0.085	7.148 f	carrychain_inst/c0_cout_r0[51]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	7.148	carrychain_inst/c0_cout_w [51]
CLMA_207_690/COUT	td	0.085	7.233 f	carrychain_inst/c0_cout_r0[55]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	7.233	carrychain_inst/c0_cout_w [55]
CLMA_207_696/COUT	td	0.085	7.318 f	carrychain_inst/c0_cout_r0[59]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	7.318	carrychain_inst/c0_cout_w [59]
CLMA_207_702/COUT	td	0.085	7.403 f	carrychain_inst/c0_cout_r0[63]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	7.403	carrychain_inst/c0_cout_w [63]
CLMA_207_708/COUT	td	0.085	7.488 f	carrychain_inst/c0_cout_r0[67]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	7.488	carrychain_inst/c0_cout_w [67]

Figure 3-12 Carry Chain Delay by Slow Corner Timing Analysis

	net (fanout=8)	0.147	4.288	nt_o_start
CLMA_207_612/COUT	td	0.112	4.400 f	carrychain_inst/c0_cout_r0[3]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.400	carrychain_inst/c0_cout_w [3]
CLMA_207_618/COUT	td	0.056	4.456 f	carrychain_inst/c0_cout_r0[7]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.456	carrychain_inst/c0_cout_w [7]
CLMA_207_624/COUT	td	0.056	4.512 f	carrychain_inst/c0_cout_r0[11]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.512	carrychain_inst/c0_cout_w [11]
CLMA_207_630/COUT	td	0.056	4.568 f	carrychain_inst/c0_cout_r0[15]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.568	carrychain_inst/c0_cout_w [15]
CLMA_207_636/COUT	td	0.056	4.624 f	carrychain_inst/c0_cout_r0[19]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.624	carrychain_inst/c0_cout_w [19]
CLMA_207_642/COUT	td	0.056	4.680 f	carrychain_inst/c0_cout_r0[23]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.680	carrychain_inst/c0_cout_w [23]
CLMA_207_648/COUT	td	0.056	4.736 f	carrychain_inst/c0_cout_r0[27]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.736	carrychain_inst/c0_cout_w [27]
CLMA_207_654/COUT	td	0.056	4.792 f	carrychain_inst/c0_cout_r0[31]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.792	carrychain_inst/c0_cout_w [31]
CLMA_207_660/COUT	td	0.056	4.848 f	carrychain_inst/c0_cout_r0[35]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.848	carrychain_inst/c0_cout_w [35]
CLMA_207_666/COUT	td	0.056	4.904 f	carrychain_inst/c0_cout_r0[39]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.904	carrychain_inst/c0_cout_w [39]
CLMA_207_672/COUT	td	0.056	4.960 f	carrychain_inst/c0_cout_r0[43]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	4.960	carrychain_inst/c0_cout_w [43]
CLMA_207_678/COUT	td	0.056	5.016 f	carrychain_inst/c0_cout_r0[47]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	5.016	carrychain_inst/c0_cout_w [47]
CLMA_207_684/COUT	td	0.056	5.072 f	carrychain_inst/c0_cout_r0[51]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	5.072	carrychain_inst/c0_cout_w [51]
CLMA_207_690/COUT	td	0.056	5.128 f	carrychain_inst/c0_cout_r0[55]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	5.128	carrychain_inst/c0_cout_w [55]
CLMA_207_696/COUT	td	0.056	5.184 f	carrychain_inst/c0_cout_r0[59]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	5.184	carrychain_inst/c0_cout_w [59]
CLMA_207_702/COUT	td	0.056	5.240 f	carrychain_inst/c0_cout_r0[63]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	5.240	carrychain_inst/c0_cout_w [63]
CLMA_207_708/COUT	td	0.056	5.296 f	carrychain_inst/c0_cout_r0[67]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	5.296	carrychain_inst/c0_cout_w [67]
CLMA_207_714/COUT	td	0.056	5.352 f	carrychain_inst/c0_cout_r0[71]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	5.352	carrychain_inst/c0_cout_w [71]
CLMA_207_720/COUT	td	0.056	5.408 f	carrychain_inst/c0_cout_r0[75]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	5.408	carrychain_inst/c0_cout_w [75]
CLMA_207_726/COUT	td	0.056	5.464 f	carrychain_inst/c0_cout_r0[79]/opit_0_inv_AL5Q/COUT
	net (fanout=1)	0.000	5.464	carrychain_inst/c0_cout_w [79]

Figure 3-13 Carry Chain Delay by Fast Corner Timing Analysis

3.5 Precautions

1. Reference design PLL input clock is 20MHz. If the PLL reference clock changes, the PLL should be reconfigured. The system clock is 250MHz. To change the clock frequency, modify the time_compute module code. The red box in the figure below indicates the system clock cycle. If the system clock changes to 500MHz, the value is 2.

```
//------------------------------------------------------------------------------
//Calculate the interval between the rising edge of the start signal and that of the stop signal
always @ (posedge clk or negedge rst_n)
    if(!rst_n)begin
        total_time<=0;
    end
    else if(i_fine_comp_done)begin
        total_time<=coarse_time_cnt*1000*4-i_fine_time;
    end
    else if(i_start)begin
        total_time<=0;
    end
```

2. When changing the parameter CRY_NUMto alter the number of carry chain stages, ensure that the parameter is a multiple of 4, and at the same time,CARYYCHAIN_FULL must also be changed to the value of CRY_NUM/4.

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage license, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its series products at any time without prior notice.
4. The information contained in this document is intended to assist users in resolving application-related issues. While we strive for accuracy, we cannot guarantee that the document is entirely free from flaws. Should any functional abnormalities and performance degradation arise due to deviation from the prescribed procedures outlined herein, our company will neither be held liable nor concede that such issues stem from product deficiencies. The solutions presented in this document are just one of the feasible options and cannot cover all application scenarios. Consequently, if users encounter functional abnormalities or performance degradation despite adhering to the prescribed procedures outlined herein, we cannot assure that such issues are indicative of product deficiencies.