

## Summary - AD9361 DAC

Package Prefix	ocpi.assets.devices
Name	ad9361_dac
OpenCPI Release	v1.5 (released 4/2019)
Workers	ad9361_dac.hdl
Tested Platforms	<ul style="list-style-type: none"> <li>• Agilent Zedboard/Analog Devices FMCOMMS2 (Vivado only)</li> <li>• Agilent Zedboard/Analog Devices FMCOMMS3 (Vivado only)</li> <li>• x86/Xilinx ML605/Analog Devices FMCOMMS2 (FMC-LPC slot only)</li> <li>• x86/Xilinx ML605/Analog Devices FMCOMMS3 (FMC-LPC slot only)</li> <li>• Ettus E310 (Vivado only, application for testing exists in e310 project)</li> </ul>

### *Revision History*

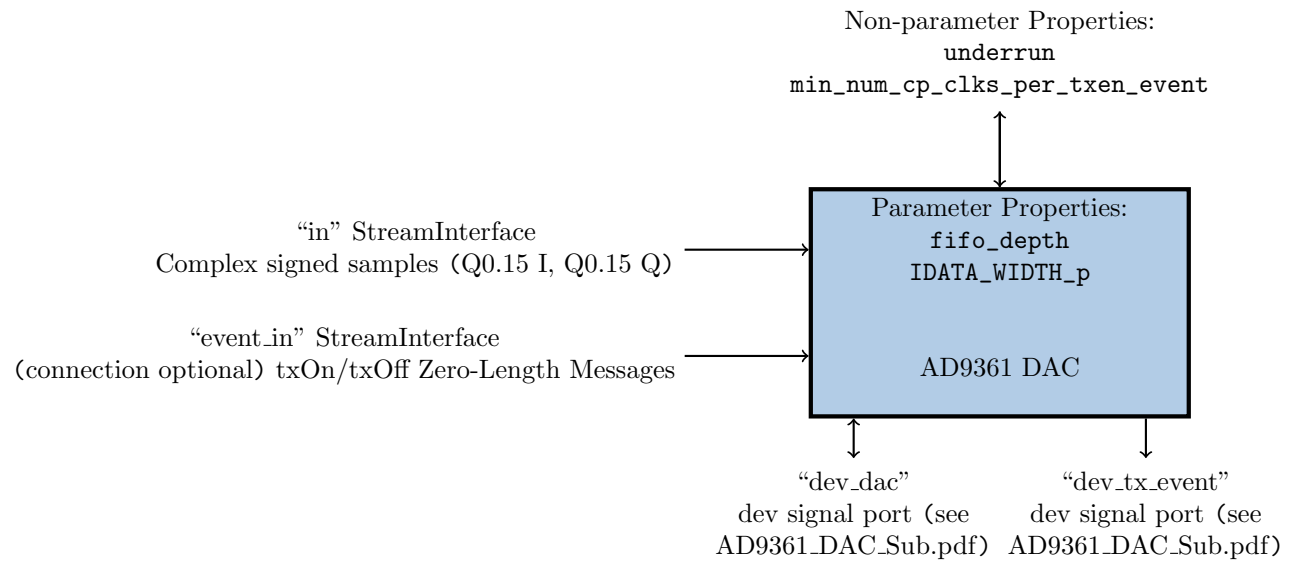
Revision	Description of Change	Date
v1.3	Initial release	3/2018
v1.3.1	Version bump only	4/2018
v1.4	<ul style="list-style-type: none"> <li>• TX powerdown functionality added (added optional <code>event_in</code> port, <code>dev_tx_event</code> devsignal port, and <code>min_num_cp_clks_per_txen_event</code> property)</li> <li>• <code>IDATA_WIDTH_p</code> parameter property added</li> <li>• source dependency list updated (new sources added, paths now specified by project)</li> </ul>	10/2018
v1.5	Version bump only	4/2019

## Functionality

The AD9361 DAC device worker ingests a single TX channel's data to be sent to the AD9361 IC [1]. Up to two instances of this worker can be used to send multichannel TX data to an AD9361 in an independent, non-phase-coherent fashion. This worker also has a port which controls AD9361 transmitter power on/off.

## Block Diagrams

### Top level



## Worker Implementation Details

### ad9361\_dac.hdl

#### DAC Data Flow (in port)

The `ad9361_dac.hdl` worker ingests signed Q0.15 I/Q samples from its `in` port, rounds them to signed Q0.11 I/Q samples, then passes them through an asynchronous First-In-First-Out (FIFO) buffer on to the `dev_dac` devsignal bus. The rounding is done in order to map to the AD9361's 12-bit I/Q DAC bus[2]. For more information on how `ad9361_dac_sub.hdl` handles the data from this worker's `dev_dac` port, see [4]. The asynchronous FIFO is necessary in order to cross clock domains from control clock to `dev_dac`'s `dac_clk` clock. Note that the control clock rate is considered static but platform-specific and that the clock rate of `dac_clk` is potentially runtime variable. The FIFO's depth in number of samples is determined at build-time by the `fifo_depth` parameter property. The `in` port's data width is also configurable via the `IDATA_WIDTH_p` parameter property, whose default value of 32 allows for a signed Q0.15 I/ Q0.15 Q input sample to be processed in a single control clock cycle. An `underrun` property indicates when invalid samples have been clocked in by the DAC due to the FIFO being empty.

#### AD9361 Transmitter Power Control (event\_in port)

The `event_in` port provides a port message-based mechanism (in the control plane clock domain) for turning on/off the AD9361 transmitter. Connection of this port is optional. If the port is disconnected, the transmitter will be on for the duration of an application.

The transmitter is powered *on* when:

- the AD9361 is being initialized (typical duration is 200 ms), or
- one or more `ad9361_dac.hdl` workers exist in the bitstream and
  - all `ad9361_dac.hdl` workers have their `event_in` ports disconnected and any are in the operating state, or
  - any `ad9361_dac.hdl` workers have their `event_in` ports connected and receive a `txOn` message (while in the operating state)

The transmitter is powered *off* when:

- the AD9361 is not being initialized and
  - no `ad9361_dac.hdl` workers exist in the bitstream, or
  - one `ad9361_dac.hdl` worker exists in the bitstream and
    - \* is not in the operating state, or
    - \* is in operating state and has its `event_in` port connected but has not yet received a message on its `event_in` port,
    - \* is in operating state and has its `event_in` port connected and receives a `txOff` message on its `event_in` port
  - two `ad9361_dac.hdl` workers exist in the bitstream and
    - \* both workers are not in the operating state, or
    - \* no workers which are in the operating state and have their `event_in` port connected have yet received a message on their `event_in` port
    - \* both workers have received a `txOff` message on their `event_in` ports in succession (with no `txOn` messages in-between) while in the operating state

Note that the normal use case for utilizing both DAC channels (and thus using two `ad9361_dac.hdl` workers) is for MIMO applications. As such, the normal use case is to either have no `event_in` ports connected, causing the transmitter to default to on for the duration of an application, or to have all connected and to send the same `txOn`/`txOff` message to all `event_in` ports simultaneously.

Note that `event_in` messages exist in the control plane clock domain but the AD9361 registers them in the AD9361 FB\_CLK domain, which may be slower than the control plane clock . The `min_num_cp_clks_per_txen_event` property enforces that tx events are properly synchronized to the AD9361 FB\_CLK without losing any events by ensuring `event_in` messages are properly spaced out. This is done by applying backpressure to the `event_in` port after each message is received. Backpressure is applied for `min_num_cp_clks_per_txen_event` - 1 number of control plane clocks after each `event_in` message is received. See property description for more info on calculation of the `min_num_cp_clks_per_txen_event` value. This property's default value of 180 was calculated using the worst-case (lowest) AD9361 sampling rate, worst-case (highest) AD9361 TX FIR interpolation factor, and highest known control plane clock rate of 125 MHz. The AD9361 LVDS or CMOS single port full duplex DDR mode was used for the default calculation. This property's value can be lowered for specific sampling rates / control plane clock rates if desired. Note that if using with a control plane clock rate of greater than 125 MHz, the default value should be overridden with a higher value.

## Source Dependencies

### ad9361\_dac.hdl

- core/hdl/primitives/util/util\_pkg.vhd
- core/hdl/primitives/util/zlm\_detector.vhd
- assets/hdl/devices/ad9361\_dac.hdl/ad9361\_dac.vhd
- assets/hdl/devices/ad9361\_dac.hdl/trunc\_round\_16\_to\_12\_signed.vhd
- assets/hdl/primitives/misc\_prims/event\_in\_to\_txen/src/event\_in\_to\_txen.vhd
- assets/hdl/primitives/misc\_prims/misc\_prims\_pkg.vhd
- assets/hdl/primitives/util/dac\_fifo.vhd
- assets/hdl/primitives/util/util\_pkg.vhd
- assets/hdl/primitives/util/sync\_status.vhd
- assets/hdl/primitives/bsv/imports/SyncFIFO.v
- assets/hdl/primitives/bsv/imports/SyncResetA.v
- assets/hdl/primitives/bsv/imports/SyncHandshake.v
- assets/hdl/primitives/bsv/bsv\_pkg.vhd

## Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
underrun	Bool	-	-	Volatile, Writable	Standard	-	Flag set when DAC tries to send a sample and the DAC FIFO is empty. Once high, this flag is not cleared (i.e. set low) until the property is written to again (the flag clears regardless of write value, i.e. writing true or false both result in a value of false).

## Worker Properties

### ad9361\_dac.hdl

Scope	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	fifo_depth	ULong	-	-	Parameter	Standard	64	Depth in number of samples of the control-to-DAC clock domain crossing FIFO.
Property	IDATA_WIDTH_p	UShort	-	-	Parameter	Standard	32	
Property	min_num_cp_clks_per_txen_events	UShort	-	-	Initial, Readable	Standard	180	After every ZLM received on the <b>event_in</b> port, backpressure will be held on that port for one less than the number of control plane clock cycles specified by this property. This is done in order to ensure tx events are properly synchronized to the AD9361 FB.CLK without losing any events. Minimum required value is $\text{ceil}(1.5 * \text{control plane clock rate} / \text{AD9361 FB.CLK rate})$ [use lowest expected FB.CLK rate for your scenario]).

## Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
in	False	iqstream_protocol	False	-	Complex signed samples (Q0.15 I, Q0.15 Q).
event_in	False	tx_event-prot	False	-	TX on/off events.

## Worker Interfaces

### ad9361\_dac.hdl

Type	Name	DataWidth	Optional	Advanced	Usage
StreamInterface	in	32	False	-	Complex signed samples (Q0.15 I, Q0.15 Q). This port ingests data and forces backpressure. Because both a “pulling” pressure from the DAC clock and potentially limited “pushing pressure” from this port exists, it is possible for a value to be clocked to the DAC while no new value was yet seen at the in port. This event is monitored via the <b>underrun</b> property.
StreamInterface	event_in	-	True	-	TX on/off events.

Type	Name	Count	Optional	Master	Signal	Direction	Width	Description
DevSignal	dev_dac	1	False	True	present	Output	1	Value is hardcoded to logic 1 inside this worker.
					dac_clk	Input	1	Clock for dac_ready, dac_take, dac_data_I, and dac_data_Q.
					dac_ready	Output	1	Indicates that the dac_data_I and dac_data_Q are valid/ready to be latched on the next rising edge of dac_clk.
					dac_take	Input	1	Indicates that dac_data_I and dac_data_Q were latched on the previous rising edge of dac_clk. If in the previous clock cycle dac_ready was 1, the values of dac_data_I and dac_data_Q should not be allowed to update with a new sample until dac_take is 1.
					dac_data_I	Output	12	Signed Q0.11 I value of DAC sample corresponding to RX channel 1.
					dac_data_Q	Output	12	Signed Q0.11 Q value of DAC sample corresponding to RX channel 1.

# Control Timing and Signals

## Clock Domains

The AD9361 DAC device worker contains two clock domains: the clock from the control plane, and the `dac_clk` clock from the `devsignal`. It is expected that the control plane clock is faster than the `dac_clk` clock in order to prevent a FIFO underrun (monitored via the `underrun` property).

## Latency

The latency from the input port to the `devsignal` data bus is both non-deterministic and dynamic. Non-determinism exists as a result of the data flowing through an asynchronous FIFO with each side in a different clock domain. Runtime dynamism exists as a result of the AD9361 `DATA_CLK_P` clock, and therefore the `dac_clk` clock rates, being runtime dynamic. The use of any FIFO, synchronous or asynchronous, between the input port and the `devsignal` also creates runtime dynamism in latency.

## Backpressure

Backpressure is transferred from the `devsignal`'s `dac_clk` clock to the input port. The input port is expected to frequently experience backpressure in order to prevent a FIFO underrun. Backpressure is applied to the `event_in` port according to the `min_num_cp_clks_per_txen_event` property value.

# Worker Configuration Parameters

## ad9361\_dac.hdl

Table 3: Table of Worker Configurations for worker: `ad9361_adc`

Configuration	ocpi_debug	ocpi_endian	fifo_depth
0	false	little	64

# Performance and Resource Utilization

## ad9361\_dac.hdl

Fmax refers to the maximum allowable clock rate for any registered signal paths within a given clock domain for an FPGA design. Fmax in the table below is specific only to this worker and represents the maximum possible Fmax for any OpenCPI bitstream built with this worker included. Note that the Fmax value for a given clock domain for the final bitstream is often worse than the Fmax specific to this worker, even if this worker is the only one included in the bitstream.

Table 4: Resource Utilization Table for worker: `ad9361_dac`

Configuration	OCPI Target	Tool	Version	Device	Registers (Typ)	LUTs (Typ)	Fmax (MHz) (Typ)		Memory/Special Functions
							control plane	dev_dac.dac_clk	
0	zynq	Vivado	2017.1	xc7z020clg484-1	235	225	186	232	
0	stratix4	Quartus	17.1.0	EP4SGX230KF40C2	232	315	N/A		N/A
0	virtex6	ISE	14.7	6vlx240tff1156-1	254	371	322.373	383.877	RAM64M: 8

<sup>1</sup>These measurements were the result of a Vivado timing analysis which was different from the Vivado analysis performed by default for OpenCPI worker builds. For more info see Appendix 1

<sup>2</sup>Quartus does not perform timing analysis at the OpenCPI worker build (i.e. synthesis) stage.



## References

- [1] AD9361 Datasheet and Product Info  
<https://www.analog.com/en/products/ad9361.html>
- [2] AD9361 Reference Manual UG-570  
AD9361\_Reference\_Manual\_UG-570.pdf
- [3] FPGA Vendor Tools Installation Guide  
[https://opencpi.github.io/FPGA\\_Vendor\\_Tools\\_Installation\\_Guide.pdf](https://opencpi.github.io/FPGA_Vendor_Tools_Installation_Guide.pdf)
- [4] AD9361 DAC Sub Component Data Sheet  
[https://opencpi.github.io/assets/AD9361\\_DAC\\_Sub.pdf](https://opencpi.github.io/assets/AD9361_DAC_Sub.pdf)

# 1 Appendix - Vivado Timing Analysis

The Vivado timing report that OpenCPI runs for device workers may erroneously report a max delay for a clocking path which should have been ignored. Custom Vivado tcl commands had to be run for this device worker to extract pertinent information from Vivado timing analysis. After building the worker, the following commands were run from the assets project directory (after the Vivado settings64.sh was sourced):

```
cd hdl/devices/
vivado -mode tcl
```

Then the following commands were run inside the Vivado tcl terminal:

```
open_project ad9361_dac.hdl/target-zynq/ad9361_dac_rv.xpr
synth_design -part xc7z020clg484-1 -top ad9361_dac_rv -mode out_of_context
create_clock -name clk1 -period 0.001 [get_nets {ctl_in[Clk]}]
create_clock -name clk2 -period 0.001 [get_nets {dev_dac_in[dac_clk]}]
set_clock_groups -asynchronous -group [get_clocks clk1] -group [get_clocks clk2]
```

The Fmax for the control plane clock for this worker is computed as the maximum magnitude slack with a control plane clock of 1 ps plus 2 times the assumed 1 ps control plane clock period ( $5.372 \text{ ns} + 0.002 \text{ ns} = 5.374 \text{ ns}$ ,  $1/5.374 \text{ ns} = 186.08 \text{ MHz}$ ). The Fmax for the dac.clk clock from the devsignal is computed as the maximum magnitude slack with dac.clk of 1 ps plus 2 times the assumed 1 ps dac.clk period ( $4.306 \text{ ns} + 0.002 \text{ ns} = 4.308 \text{ ns}$ ,  $1/4.287 \text{ ns} = 232.12 \text{ MHz}$ ).

The following command is run to get control plane clock timing:

```
report_timing -delay_type min_max -sort_by slack -input_pins -group clk1
```

The expected output of the command is as follows:

```
INFO: [Timing 38-35] Done setting XDC timing constraints.
INFO: [Timing 38-91] UpdateTimingParams: Speed grade: -1, Delay Type: min_max.
INFO: [Timing 38-191] Multithreading enabled for timing update using a maximum of 8 CPUs
WARNING: [Timing 38-242] The property HD.CLK_SRC of clock port "ctl_in[Clk]" is not set. In out-of-context mode, this prevents timing estimation for clock \
    delay/skew
Resolution: Set the HD.CLK_SRC property of the out-of-context port to the location of the clock buffer instance in the top-level design
WARNING: [Timing 38-242] The property HD.CLK_SRC of clock port "dev_dac_in[dac_clk]" is not set. In out-of-context mode, this prevents timing estimation for \
    clock delay/skew
Resolution: Set the HD.CLK_SRC property of the out-of-context port to the location of the clock buffer instance in the top-level design
INFO: [Timing 38-78] ReportTimingParams: -max_paths 1 -nworst 1 -delay_type min_max -sort_by slack.
Copyright 1986-2017 Xilinx, Inc. All Rights Reserved.

-----
| Tool Version : Vivado v.2017.1 (lin64) Build 1846317 Fri Apr 14 18:54:47 MDT 2017
| Date       : Wed Oct 3 16:41:06 2018
| Host      : <removed> running 64-bit CentOS Linux release 7.5.1804 (Core)
| Command   : report_timing -delay_type min_max -sort_by slack -input_pins -group clk1
| Design    : ad9361_dac_rv
| Device    : 7z020-clg484
| Speed File : -1 PRODUCTION 1.11 2014-09-11
|-----

Timing Report

Slack (VIOLATED) :      -5.372ns (required time - arrival time)
Source:            IN_port/fifo/data0_reg_reg[13]/C
                   (rising edge-triggered cell FDRE clocked by clk1 {rise@0.000ns fall@0.001ns period=0.001ns})
Destination:       worker/fifo/fifo/fifoMem_reg/DIADI[9]
```

```

(rising edge-triggered cell RAMB18E1 clocked by clk1 {rise@0.000ns fall@0.001ns period=0.001ns})
Path Group:      clk1
Path Type:       Setup (Max at Slow Process Corner)
Requirement:     0.002ns (clk1 rise@0.002ns - clk1 rise@0.000ns)
Data Path Delay: 4.374ns (logic 1.904ns (43.533%) route 2.470ns (56.467%))
Logic Levels:    5 (CARRY4=3 LUT4=1 LUT5=1)
Clock Path Skew: -0.049ns (DCD - SCD + CPR)

Destination Clock Delay (DCD): 0.924ns = ( 0.926 - 0.002 )
Source Clock Delay (SCD): 0.973ns
Clock Pessimism Removal (CPR): 0.000ns
Clock Uncertainty: 0.035ns ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
Total System Jitter (TSJ): 0.071ns
Total Input Jitter (TIJ): 0.000ns
Discrete Jitter (DJ): 0.000ns
Phase Error (PE): 0.000ns

```

Location	Delay type	Incr(ns)	Path(ns)	Netlist Resource(s)
	(clock clk1 rise edge)	0.000	0.000	r
		0.000	0.000	r ctl_in[Clk] (IN)
net (fo=198, unset)		0.973	0.973	IN_port/fifo/ctl_in[Clk]
FDRE				r IN_port/fifo/data0_reg_reg[13]/C
FDRE (Prop_fdre_C_Q)		0.518	1.491	r IN_port/fifo/data0_reg_reg[13]/Q
net (fo=3, unplaced)		0.759	2.250	IN_port/fifo/IN_data[5]
				r IN_port/fifo/fifoMem_reg_i_36/I1
LUT4 (Prop_lut4_I1_0)		0.295	2.545	r IN_port/fifo/fifoMem_reg_i_36/0
net (fo=1, unplaced)		0.902	3.447	IN_port/fifo/fifoMem_reg_i_36_n_0
				r IN_port/fifo/fifoMem_reg_i_24/I1
LUT5 (Prop_lut5_I1_0)		0.124	3.571	r IN_port/fifo/fifoMem_reg_i_24/0
net (fo=1, unplaced)		0.000	3.571	IN_port/fifo/fifoMem_reg_i_24_n_0
				r IN_port/fifo/fifoMem_reg_i_5/S[0]
CARRY4 (Prop_carry4_S[0]_C0[3])		0.513	4.084	r IN_port/fifo/fifoMem_reg_i_5/C0[3]
net (fo=1, unplaced)		0.009	4.093	IN_port/fifo/fifoMem_reg_i_5_n_0
				r IN_port/fifo/fifoMem_reg_i_4/CI
CARRY4 (Prop_carry4_CI_C0[3])		0.117	4.210	r IN_port/fifo/fifoMem_reg_i_4/C0[3]
net (fo=1, unplaced)		0.000	4.210	IN_port/fifo/fifoMem_reg_i_4_n_0
				r IN_port/fifo/fifoMem_reg_i_3/CI
CARRY4 (Prop_carry4_CI_0[1])		0.337	4.547	r IN_port/fifo/fifoMem_reg_i_3/0[1]
net (fo=1, unplaced)		0.800	5.347	worker/fifo/fifo/sD_IN[9]
RAMB18E1				r worker/fifo/fifo/fifoMem_reg/DIADI[9]
	(clock clk1 rise edge)	0.002	0.002	r
		0.000	0.002	r ctl_in[Clk] (IN)
net (fo=198, unset)		0.924	0.926	worker/fifo/fifo/ctl_in[Clk]
RAMB18E1				r worker/fifo/fifo/fifoMem_reg/CLKBWRCLK
clock pessimism		0.000	0.926	
clock uncertainty		-0.035	0.891	
RAMB18E1 (Setup_ramb18e1_CLKBWRCLK_DIADI[9])		-0.916	-0.025	worker/fifo/fifo/fifoMem_reg
required time			-0.025	
arrival time			-5.347	

```
-----
slack                                -5.372
-----
```

```
report_timing: Time (s): cpu = 00:00:08 ; elapsed = 00:00:09 . Memory (MB): peak = 2095.184 ; gain = 497.547 ; free physical = 7704 ; free virtual = 54670
```

The following command is run to get dev\_dac.dac\_clk timing:

```
report_timing -delay_type min_max -sort_by slack -input_pins -group clk2
```

The expected output of the command is as follows:

```
INFO: [Timing 38-91] UpdateTimingParams: Speed grade: -1, Delay Type: min_max.
INFO: [Timing 38-191] Multithreading enabled for timing update using a maximum of 8 CPUs
INFO: [Timing 38-78] ReportTimingParams: -max_paths 1 -nworst 1 -delay_type min_max -sort_by slack.
Copyright 1986-2017 Xilinx, Inc. All Rights Reserved.
-----
| Tool Version : Vivado v.2017.1 (lin64) Build 1846317 Fri Apr 14 18:54:47 MDT 2017
| Date       : Thu Oct 4 10:56:37 2018
| Host       : <removed> running 64-bit CentOS Linux release 7.5.1804 (Core)
| Command    : report_timing -delay_type min_max -sort_by slack -input_pins -group clk2
| Design     : ad9361_dac_rv
| Device     : 7z020-clg484
| Speed File  : -1 PRODUCTION 1.11 2014-09-11
-----

Timing Report

Slack (VIOLATED) :      -4.306ns (required time - arrival time)
Source:            worker/fifo/fifo/dEnqPtr_reg[0]/C
                   (rising edge-triggered cell FDCE clocked by clk2 {rise@0.000ns fall@0.001ns period=0.001ns})
Destination:       worker/fifo/fifo/fifoMem_reg/ENARDEN
                   (rising edge-triggered cell RAMB18E1 clocked by clk2 {rise@0.000ns fall@0.001ns period=0.001ns})
Path Group:        clk2
Path Type:         Setup (Max at Slow Process Corner)
Requirement:       0.002ns (clk2 rise@0.002ns - clk2 rise@0.000ns)
Data Path Delay:   3.781ns (logic 1.061ns (28.063%) route 2.720ns (71.937%))
Logic Levels:      3 (LUT2=1 LUT6=2)
Clock Path Skew:   -0.049ns (DCD - SCD + CPR)
  Destination Clock Delay (DCD): 0.924ns = ( 0.926 - 0.002 )
  Source Clock Delay (SCD):      0.973ns
  Clock Pessimism Removal (CPR): 0.000ns
Clock Uncertainty: 0.035ns ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
  Total System Jitter (TSJ):     0.071ns
  Total Input Jitter (TIJ):      0.000ns
  Discrete Jitter (DJ):          0.000ns
  Phase Error (PE):              0.000ns

Location      Delay type      Incr(ns) Path(ns)  Netlist Resource(s)
-----
              (clock clk2 rise edge)  0.000  0.000 r
              0.000  0.000 r dev_dac_in[dac_clk] (IN)
net (fo=35, unset)  0.973  0.973 worker/fifo/fifo/dev_dac_in[dac_clk]
FDCE              r worker/fifo/fifo/dEnqPtr_reg[0]/C
-----
```

FDCE (Prop_fdce_C_Q)	0.518	1.491	r	worker/fifo/fifo/dEnqPtr_reg[0]/Q
net (fo=1, unplaced)	0.965	2.456		worker/fifo/fifo/dEnqPtr[0]
			r	worker/fifo/fifo/dGDeqPtr_rep[0]_i_3/I0
LUT6 (Prop_lut6_I0_0)	0.295	2.751	r	worker/fifo/fifo/dGDeqPtr_rep[0]_i_3/0
net (fo=1, unplaced)	0.449	3.200		worker/fifo/fifo/dGDeqPtr_rep[0]_i_3_n_0
			r	worker/fifo/fifo/dGDeqPtr_rep[0]_i_1/I1
LUT6 (Prop_lut6_I1_0)	0.124	3.324	r	worker/fifo/fifo/dGDeqPtr_rep[0]_i_1/0
net (fo=18, unplaced)	0.506	3.830		worker/fifo/fifo/dGDeqPtr0
			r	worker/fifo/fifo/fifoMem_reg_i_1/I0
LUT2 (Prop_lut2_I0_0)	0.124	3.954	r	worker/fifo/fifo/fifoMem_reg_i_1/0
net (fo=1, unplaced)	0.800	4.754		worker/fifo/fifo/fifoMem_reg_i_1_n_0
RAMB18E1			r	worker/fifo/fifo/fifoMem_reg/ENARDEN
-----				
(clock clk2 rise edge)	0.002	0.002	r	
	0.000	0.002	r	dev_dac_in[dac_clk] (IN)
net (fo=35, unset)	0.924	0.926		worker/fifo/fifo/dev_dac_in[dac_clk]
RAMB18E1			r	worker/fifo/fifo/fifoMem_reg/CLKARDCLK
clock pessimism	0.000	0.926		
clock uncertainty	-0.035	0.891		
RAMB18E1 (Setup_ramb18e1_CLKARDCLK_ENARDEN)				
	-0.443	0.448		worker/fifo/fifo/fifoMem_reg
-----				
required time		0.448		
arrival time		-4.754		
-----				
slack		-4.306		