Summary - AD9361 DAC Sub

Name	ad9361_dac_sub
Worker Type	Device
Version	v1.3
Release Date	Aug 2017
Component Library	ocpi.devices
Workers	ad9361_dac_sub.hdl
Tested Platforms	Zedboard (ISE and Vivado), ML605 (FMC LPC slot)

Functionality

The AD9361 DAC Sub is a subdevice worker whose primary purpose is to time-interleave data streams for two TX channels in preparation for sending to the AD9361 IC pins (independent of which of the IC's P0/P1 buses the TX data streams are sent to). Time-interleaving occurs according to the timing diagrams specified in figures 66, 69, 72, 75, and 80 in [2]. This worker ingests data from at most two instances of the ad9361_dac.hdl device worker, each which handles a single TX channel data stream, and time-interleaves all channels onto a single data bus that is sent out eventually (via devsignals to ad9361_data_sub.hdl[4]) to the appropriate TX data stream pins of the AD9361 IC[1].

Worker Implementation Details

ad9361_dac_sub.hdl

The ad9361_dac_sub.hdl subdevice worker handles registering and interleaving of two independent TX data streams which are sent to this worker via the `dev_data_ch0_in` and `dev_data_ch1_in` devsignal ports. Data is sent out via the `dev_data_to_pins` devsignal port which ad9361_data_sub.hdl routes to the AD9361[4]. This worker's `LVDS_p`, `HALF_DUPLEX_p`, `SINGLE_PORT_p`, and `DATA_RATE_CONFIG_p` parameter properties enforce build-time configuration[1] for all of the possible AD9361 TX data time-interleaved modes:

- CMOS Single Port Half Duplex SDR,
- CMOS Single Port Half Duplex DDR,
- CMOS Single Port Full Duplex SDR,
- CMOS Single Port Full Duplex DDR,
- CMOS Dual Port Half Duplex SDR,
- CMOS Dual Port Half Duplex DDR,
- CMOS Dual Port Full Duplex SDR,
- CMOS Dual Port Full Duplex DDR, and
- LVDS (Dual Port Full Duplex DDR).

[1] Although parameter property infrastructure is in place for all data interface configurations, LVDS is the only currently supported configuration.

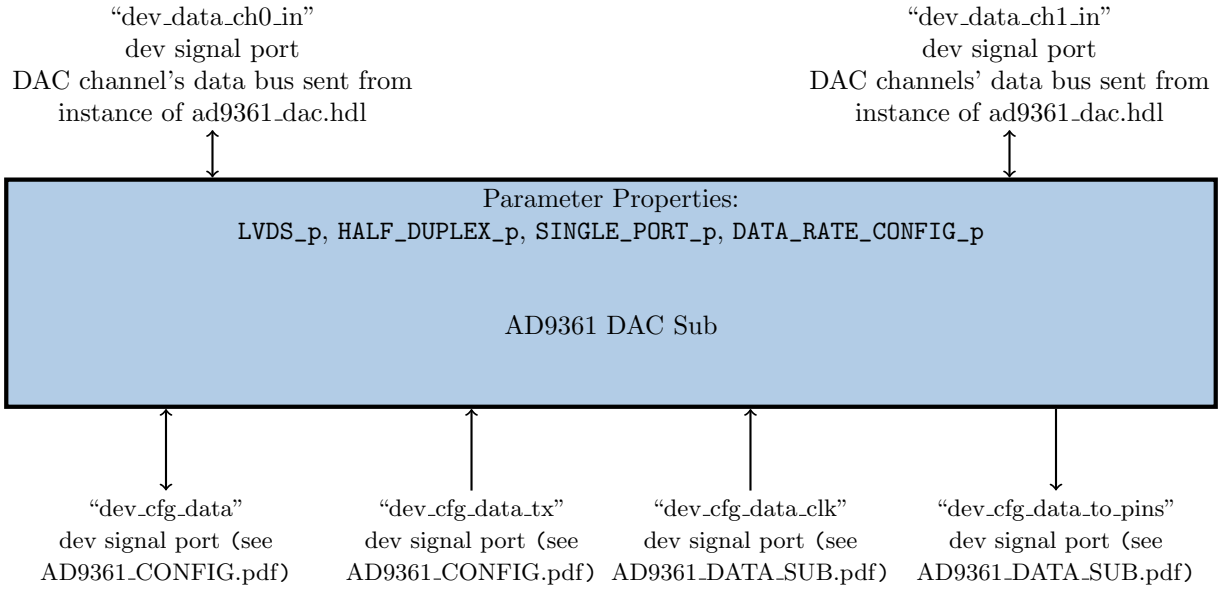
Note that "channel 0" within the context of this worker corresponds to the AD9361 T1 channel and "channel 1" corresponds to the AD9361 T2 channel in the AD9361 timing diagrams[2]. The mapping between the AD9361's T1/T2 channels and the AD9361 physical TX connector outputs is variable depending on the AD9361 register configuration. This relationship is shown in the following table. Due to the AD9361's T2 behavior[3], channel 1 should only ever be used when 2 TX channels are desired.

Table 1: Channel Connectivity (D.C. means Don't Care.)

ad9361_dac_sub.hdl devsignal channel	AD9361 timing diagram channel	AD9361 TX RF Port	AD9361 Register 0x010 Bit D5[4] (channel swap)	AD9361 Register 0x002 Bits [D7 D6][5] (channel enable)	AD9361 Register 0x004 Bit D6 (port select)
0	T1	TX1A	0	[D.C. 1]	0
0	T1	TX1B	0	[D.C. 1]	1
0	T1	TX2A	1	[1 D.C.]	0
0	T1	TX2B	1	[1 D.C.]	1
1 [3]	T2 [3]	TX2A	0	[1 1]	0
1 [3]	T2 [3]	TX2B	0	[1 1]	1
1 [3]	T2 [3]	TX1A	1	[1 1]	0
1 [3]	T2 [3]	TX1B	1	[1 1]	1

Block Diagrams

Top level



[2]For more info, see e.g. Figure 80 in [2].

[3]Data sent via T2 is only ever transmitted when the AD9361 register 0x002 Bits D7 and D6 are 1 (which corresponds to one of 1R2T or 2R2T modes).

[4]Note that AD9361 register 0x010 Bit D5 is controlled by no-OS's AD9361_InitParam struct's tx_channel_swap_enable member[2] and that the ad9361_config_proxy.rcc worker's `ad9361_init` property sets that member value[6].

[5]Note that AD9361 register 0x002 Bits [D7 D6] are controlled by no-OS's AD9361_InitParam struct's one_rx_one_tx_use_tx_num member and two_rx_two_tx_mode_enable member[2] and that the ad9361_config_proxy.rcc worker's `ad9361_init` property sets these member values[6].

Source Dependencies

ad9361_dac_sub.hdl

- opencpi/hdl/devices/ad9361_dac_sub.hdl/ad9361_dac_sub.vhd
- opencpi/hdl/primitives/bsv/imports/SyncBit.v
- opencpi/hdl/primitives/bsv/bsv_pkg.vhd

Component Spec Attributes

Attribute	Value
NoControl	True

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
-	-	-	-	-	-	-	-

Worker Properties

ad9361_dac_sub.hdl

Scope	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	`LVDS_p`	Bool	-	-	Parameter	Standard	False	Use LVDS TX data bus interleaving scheme, otherwise use CMOS interleaving scheme. Default is CMOS.
Property	`HALF_DUPLEX_p`	Bool	-	-	Parameter	Standard	False	Use half duplex mode, otherwise use full duplex mode. Must be false when using LVDS mode.
Property	`SINGLE_PORT_p`	Bool	-	-	Parameter	Standard	False	Use single port, otherwise use both (dual) ports. Default is to use both ports. Must be false when using LVDS mode.
Property	`DATA_RATE_CONFIG_p`	Enum	-	-	Parameter	SDR, DDR	DDR	This should have a value of DDR when `LVDS_p` has a value of true. Either value is acceptable when `LVDS_p` has a value of false (i.e. CMOS mode is used).

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
-	-	-	-	-	-

Worker Interfaces

ad9361_dac_sub.hdl

Type	Name	Count	Optional	Master	Signal	Direction	Width	Description
DevSignal	dev_cfg_data	1	False	True	config_is_two_r	Input	1	Some data port configurations (such as LVDS) require the TX bus to use 2R2T timing if either 2 TX or 2 RX channels are used. For example, if using LVDS and this has a value of 1, 2R2T timing will be forced.
					ch0_handler_is_present	Output	1	Value is 1 if the dev_data_ch0 dev signal is connected to a worker (that "handles" the data) and 0 otherwise. This is expected to be hardcoded at buildtime.
					ch1_handler_is_present	Output	1	Value is 1 if the dev_data_ch1 dev signal is connected to a worker (that "handles" the data) and 0 otherwise. This is expected to be hardcoded at buildtime.
					data_bus_index_direction	Output	1	Value is 1 if the bus indexing of the P0_D/P1_D signals from dev_data_from_pins was reversed before processing. This is expected to be hardcoded at buildtime.
					data_clk_is_inverted	Output	1	Value is 1 if the clock in via dev_data_clk was inverted inside this worker before used as an active-edge rising clock. This is expected to be hardcoded at buildtime.
					islvds	Output	1	Value is 1 if `LVDS_p` has a value of true and 0 if `LVDS_p_p` has a value of false. Because `LVDS_p` is a parameter property, this is hardcoded at buildtime. The purpose of this devsignal is to feed this worker's buildtime-specified LVDS/CMOS mode through ad9361_config.hdl to ad9361_config_proxy.rcc so No-OS knows which LVDS/CMOS mode to use when initializing the AD9361 IC.
					isdualport	Output	1	Value is 1 if `SINGLE_PORT_p` has a value of false and 0 if `SINGLE_PORT_p` has a value of true. Because `SINGLE_PORT_p` is a parameter property, this is hardcoded at buildtime. The purpose of this devsignal is to feed this worker's buildtime-specified single/dual port mode through ad9361_config.hdl to ad9361_config_proxy.rcc so No-OS knows which single/dual port mode to use when initializing the AD9361 IC.
					isfullduplex	Output	1	Value is 1 if `HALF_DUPLEX_p` has a value of false and 0 if `HALF_DUPLEX_p` has a value of true. Because `HALF_DUPLEX_p` is a parameter property, this is hardcoded at buildtime. The purpose of this devsignal is to feed this worker's buildtime-specified half/full duplex mode through ad9361_config.hdl to ad9361_config_proxy.rcc so No-OS knows which half/full duplex mode to use when initializing the AD9361 IC.
					isDDR	Output	1	Value is 1 if `DATA_RATE_CONFIG_p` has a value of DDR and 0 if `DATA_RATE_CONFIG_p` has a value of SDR. Because `DATA_RATE_CONFIG_p` is a parameter property, this is hardcoded at buildtime. The purpose of this devsignal is to feed this worker's buildtime-specified SDR/DDR mode through ad9361_config.hdl to ad9361_config_proxy.rcc so No-OS knows which half/full duplex mode to use when initializing the AD9361 IC.
					present	Output	1	Used to communicate to ad9361_config.hdl that it should validate the islvds, isdualport, isfullduplex, and isddr signals against similar signals in the ad9361_adc_sub.hdl and ad9361_data_sub.hdl workers if they are present in the bitstream. This is expected to be hardcoded at buildtime.

DevSignal	dev_cfg_data_tx	1	False	True	config_is_two_t	Input	1	Some data port configurations (such as LVDS) require the TX bus to use 2R2T timing if either 2 TX or 2 RX channels are used. For example, if using LVDS and this has a value of 1, 2R2T timing will be forced.
					force_two_r_two_t_timing	Input	1	Expected to match value of AD9361 register 0x010 bit D2[3].
DevSignal	dev_data_clk	1	False	True	DATA_CLK_P	Input	1	Buffered version of AD9361 DATA_CLK_P pin.
DevSignal	dev_data_to_pins	1	False	True	data	Output	24	Data bus containing configuration-specific AD9361 pins corresponding to the TX data path: * CMOS single port half duplex: [12'b0 P0_D[11:0]], * CMOS single port full duplex: [18'b0 P0_D[11:6]], * CMOS dual port half duplex: [P0_D[11:0] P1_D[11:0]], * CMOS dual port full duplex: [12'b0 P1_D[11:0]], * LVDS: [18'b0 TX_D[5:0]], or, if ports are swapped: * CMOS single port half duplex: [12'b0 P1_D[11:0]], * CMOS single port full duplex: [18'b0 P1_D[11:6]], * CMOS dual port half duplex: [P1_D[11:0] P0_D[11:0]], * CMOS dual port full duplex: [12'b0 P0_D[11:0]], * LVDS: (unsupported with port swap). For more info see [4].
					tx_frame	Output	1	Signal which will drive the output buffer which drives the AD9361 TX_FRAME_P pin.
					fb_clk	Output	1	Signal which will drive the output buffer which will drive the AD9361 FB_CLK_P pin.
DevSignal	dev_data_ch0_in	1	False	False	present	Output	1	Value is 1 if a worker is connected to this devsignal port.
					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 adc_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 0.
					dac_data_Q	Output	12	Signed Q0.11 Q value of DAC sample corresponding to RX channel 0.
DevSignal	dev_data_ch1_in	1	True	False	present	Output	1	Value is 1 if a worker is connected to this devsignal port.
					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 adc_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.

Subdevice Connections

Supports Worker	Supports Worker Port	ad9361_dac_sub.hdl Port	ad9361_dac_sub.hdl Port Index
ad9361_dac	dev_dac	dev_data_ch0_in	0
ad9361_dac	dev_dac	dev_data_ch1_in	0

Control Timing and Signals

The AD9361 DAC Sub subdevice worker contains four clock domains: control plane, FB_CLK_P, "dac_clk" and "dad2_clk".

Control Plane Clock Domain

The `data_cfg_tx` signals enter this worker in the control plane clock domain. Inside this worker, they are combined combinatorially into a single signal which is subsequently synchronized to the dacd2_clk domain. The dacd2_clk domain signal is then used to handle time-interleaving of multiple channel's data.

DATA_CLK_P/FB_CLK_P Clock Domain

The AD9361 DATA_CLK_P clock enters this worker via the `dev_data_clk` devsignal. DATA_CLK_P clock is forwarded to the `FB_CLK_P` device signal as well as used to generate dac_clk.

"dac_clk" Clock Domain

The "dac_clk" clock is an inverted version of DATA_CLK_P. Note that data transitions for the TX data sent out via the `dev_data_to_pins` devsignal port are falling-edge aligned with `FB_CLK_P` since the falling edge is what the AD9361 datasheets specifies its setup/hold requirements against[2]. The dac_clk domain allows for logic within this worker to be falling edge aligned with `FB_CLK_P`.

"dacd2_clk" Clock Domain

The "dacd2_clk" clock is a divided by 2 version of "dac_clk". TX data for channel 0 and channel 1 enter this worker in the dacd2_clk domain from the `dev_data_ch0_in` and `dev_data_ch1_in` devsignal ports, respectively. The dacd2_clk clock was a necessary replacement for some of the dac_clk logic in order to alleviate timing violations in the dac_clk domain for the zed_ise platform. Note that because dacd2_clk is a divided version of dac_clk, synchronization logic between the two is not necessary and not included.

Data latency

- **LVDS**
  Latency for LVDS is given as the number of clock cycles from a given channel's data becoming ready on the dac_data_I and dac_data_Q devsignals to the starting edge of the high 6-bit I word on the AD9361 data bus output. Note that, for multichannel modes, latency can be two possible values depending on the current state of the 2-state channel serialization state machine exists when dac_data_I/dac_data_Q becomes ready. The latency for the various LVDS configurations are as follows:
  - 1R1T
    channel 0 data latency = 3 FB_CLK_P cycles (2 are pipeline delay which are arguably unnecessary and 1 is a 12-bit word to 6-bit word serialization register),
  - 2R1T/1R2T/2R2T
    channel 0 data latency = 3 or 5 FB_CLK_P cycles (2 are pipeline delay which are arguably unnecessary, 2 are possible channel serialization register delay, and 1 is a 12-bit word to 6-bit word serialization register), channel 1 data latency = 3 or 5 FB_CLK_P cycles (2 are pipeline delay which are arguably unnecessary, 2 are possible channel serialization register delay, and 1 is a 12-bit word to 6-bit word serialization register).
- **CMOS**
  Not yet implemented or supported.

Multichannel Phase Coherency

Note that the two channel data made available via dev_data_ch0_in and dev_data_ch1_in are only ever considered to be phase coherent if coherency is guaranteed by the worker(s) that dev_data_ch0_in and dev_data_ch1_in are connected to. For example, multiple instances of ad9361_dac.hdl would not guarantee phase coherence because their datastreams would be independent. However, if a single device worker was created which interfaced with both dev_data_ch0_in and dev_data_ch1_in, phase coherency could be guaranteed by updating the values for the two channels in an every-other-clock fashion.

Performance and Resource Utilization

ad9361_dac_sub.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. Note also that the DATA_CLK_P devsignal's rate will only ever go as high as 245.76 MHz[2].

Table entries are a result of building the worker with the following parameter property sets:

- `LVDS_p`=True
- `HALF_DUPLEX_p`=False
- `SINGLE_PORT_p`=False
- `DATA_RATE_CONFIG_p`=DDR

Device	Registers (typical)	LUTs (typical)	Fmax (typical) (dev_data_clk)	Memory/Special Functions	Design Suite
Zynq XC7Z020-1-CLG484	91	88	286 MHz [1]	2 BUFR, 8 ODDR	Vivado 2017.1
	99	105	704 MHz	2 BUFR, 8 ODDR	ISE 14.7
Virtex-6 XC6VLX240T-1-FF1156	99	105	632 MHz	2 BUFR, 8 ODDR	ISE 14.7
Stratix IV EP4SGX230K-C2-F40	(unsupported)	(unsupported)	2	(unsupported)	Quartus Prime 15.1

Test and Verification

The test outlined in [5] includes validation of this worker's functionality (for LVDS only). Note that this worker does not successfully operate with the FMCOMMS2/3 card on the ML605 FMC-HPC slot due to an unknown data fidelity issue.

[1] 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

[2] Quartus does not perform timing analysis at the OpenCPI worker build (i.e. synthesis) stage.

References

[1] AD9361 Datasheet and Product Info
`http://www.analog.com/en/products/rf-microwave/integrated-transceivers-transmitters-receivers/wideband-transceivers-ic/ad9361.html`

[2] AD9361 Reference Manual UG-570
AD9361_Reference_Manual_UG-570.pdf

[3] AD9361 Register Map Reference Manual UG-671
AD9361_Register_Map_Reference_Manual_UG-671.pdf

[4] AD361 Data Sub Component Data Sheet
AD9361_Data_Sub.pdf

[5] AD361 DAC Component Data Sheet
AD9361_DAC.pdf

[6] AD361 Configy Proxy Component Data Sheet
AD9361_Config_Proxy.pdf

1 Appendix - Vivado Timing Analysis

The Vivado timing report that OpenCPI runs for device workers erroneously reports net delays which are not indicative of the maximum allowable clock rate for a given clock pin. Custom Vivado tcl commands have to be run device workers to calculate this information.

The desired maximum clock rate for the DATA_CLK_P clock domain for this worker is computed as the inverse of the DATA_CLK_P clock period which causes the worst case slack to be 0 ns. In preparation for running the timing analys, build the worker, source the Vivado settings64.sh script, the run the following commands from the base project directory):

```
cd hdl/devices/
vivado -mode tcl
open_project ad9361_dac_sub.hdl/target-zynq/ad9361_dac_sub_rv.xpr
synth_design -part xc7z020clg484-1 -top ad9361_dac_sub_rv -mode out_of_context
```

To get the desired period, we must define DATA_CLK_P as a clock in Vivado (create_clock command) with some arbitrarily chosen period (we chose 0.001 ns). Note that get_nets dev_data_clk* returns dev_data_clk_in[DATA_CLK_P], which is the desired DATA_CLK_P net which is passed into this device worker via the dev_data_clk_in devsignal.

```
create_clock -name clk2 -period 0.001 [get_nets dev_data_clk*]
```

The timing report is then run to to report the worst case slack for our defined clock period.

```
report_timing -delay_type min_max
```

The following is the output of this report:

```
Timing Report

Slack (VIOLATED) :     -3.485ns (required time - arrival time)
  Source:              worker/data_mode_lvds.reg_duplication_loop[6].dacd2_ch0_i_r_reg[6]/C
                         (rising edge-triggered cell FDRE clocked by dev_data_ch1_in_out[dac_clk] {rise@0.001ns fall@0.002ns period=0.002ns})
  Destination:         worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]/D
                         (rising edge-triggered cell FDRE 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 fall@0.002ns - dev_data_ch1_in_out[dac_clk] rise@0.001ns)
  Data Path Delay:     1.772ns (logic 1.020ns (57.562%) route 0.752ns (42.438%))
  Logic Levels:        2  (LUT3=1 MUXF7=1)
  Clock Path Skew:     -1.760ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD): 2.709ns = ( 2.712 - 0.002 )
    Source Clock Delay    (SCD):   4.632ns = ( 4.633 - 0.001 )
    Clock Pessimism Removal (CPR): 0.163ns
  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 dev_data_ch1_in_out[dac_clk] rise edge)
                                              0.001    0.001 r
                                              0.000    0.001 f dev_data_clk_in[DATA_CLK_P] (IN)
                     net (fo=0)               0.973    0.974   worker/dev_data_clk_in[DATA_CLK_P]
```

```
                    LUT1 (Prop_lut1_I0_O)      0.124    1.097 r worker/dac_clk_bufr_i_1/O
                    net (fo=1, unplaced)       0.800    1.897   worker/dac_dev_data_clk_p
                    BUFR (Prop_bufr_I_O)       0.537    2.434 r worker/dac_clk_bufr/O
                    net (fo=20, unplaced)      0.584    3.018   worker/dac_clk
                    BUFR (Prop_bufr_I_O)       1.031    4.049 r worker/BUFR_inst/O
                    net (fo=91, unplaced)      0.584    4.633   worker/dev_data_ch1_in_out[dac_clk]
                    FDRE                                    r  worker/data_mode_lvds.reg_duplication_loop[6].dacd2_ch0_i_r_reg[6]/C
  ------------------------------------------------------------------- -------------------
                    FDRE (Prop_fdre_C_Q)       0.478    5.111 r worker/data_mode_lvds.reg_duplication_loop[6].dacd2_ch0_i_r_reg[6]/Q
                    net (fo=2, unplaced)       0.752    5.863   worker/data_mode_lvds.reg_duplication_loop[6].dacd2_ch0_i_r_reg_n_0_[6]
                    LUT3 (Prop_lut3_I0_O)      0.295    6.158 r worker/data_mode_lvds.dac_data_ddr_first_r[0]_i_3/O
                    net (fo=1, unplaced)       0.000    6.158   worker/dacd2_i_r__0[6]
                    MUXF7 (Prop_muxf7_I1_O)    0.247    6.405 r worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]_i_1/O
                    net (fo=1, unplaced)       0.000    6.405   worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]_i_1_n_0
                    FDRE                                    r  worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]/D
  ------------------------------------------------------------------- -------------------

                    (clock clk2 fall edge)     0.002    0.002 f
                                               0.000    0.002 f dev_data_clk_in[DATA_CLK_P] (IN)
                    net (fo=0)                 0.924    0.927   worker/dev_data_clk_in[DATA_CLK_P]
                    LUT1 (Prop_lut1_I0_O)      0.100    1.027 r worker/dac_clk_bufr_i_1/O
                    net (fo=1, unplaced)       0.760    1.787   worker/dac_dev_data_clk_p
                    BUFR (Prop_bufr_I_O)       0.486    2.273 r worker/dac_clk_bufr/O
                    net (fo=20, unplaced)      0.439    2.712   worker/dac_clk
                    FDRE                                    r  worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]/C
                    clock pessimism            0.163    2.875
                    clock uncertainty         -0.035    2.839
                    FDRE (Setup_fdre_C_D)      0.080    2.919   worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]
  -------------------------------------------------------------------
                    required time                       2.919
                    arrival time                       -6.405
  -------------------------------------------------------------------
                    slack                              -3.485
```

In order to calculate the clock period which would give a 0.000 ns slack, we add 2*0.001 ns to the slack violation which corresponded to a 0.001 ns period. For this example, the result is -(-3.485) + 2*(0.001) = 3.487 ns. The inverse of this value is what corresponds to the 286 MHz measurement in the table above[??]. This measurement can be verified by re-defining the clock period to have a period of 3.487 ns and re-running the timing report to see a worst case slack of 0 ns.

```
create_clock -name clk2 -period 3.487 [get_nets dev_data_clk*]
report_timing -delay_type min_max
```

```
Timing Report

Slack (MET) :           0.000ns (required time - arrival time)
  Source:               worker/data_mode_lvds.reg_duplication_loop[6].dacd2_ch0_i_r_reg[6]/C
                          (rising edge-triggered cell FDRE clocked by dev_data_ch1_in_out[dac_clk] {rise@1.743ns fall@5.230ns period=6.974ns})
  Destination:          worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]/D
                          (rising edge-triggered cell FDRE clocked by clk2' {rise@0.000ns fall@1.743ns period=3.487ns})
  Path Group:           clk2
  Path Type:            Setup (Max at Slow Process Corner)
  Requirement:          3.487ns (clk2 fall@5.230ns - dev_data_ch1_in_out[dac_clk] rise@1.743ns)
  Data Path Delay:      1.772ns (logic 1.020ns (57.562%) route 0.752ns (42.438%))
  Logic Levels:         2  (LUT3=1 MUXF7=1)
```

```
Clock Path Skew:      -1.760ns (DCD - SCD + CPR)
  Destination Clock Delay (DCD): 2.709ns = ( 7.940 - 5.230 )
  Source Clock Delay    (SCD):   4.632ns = ( 6.376 - 1.743 )
  Clock Pessimism Removal (CPR): 0.163ns
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 dev_data_ch1_in_out[dac_clk] rise edge)
                                              1.743    1.743 r
                                              0.000    1.743 f dev_data_clk_in[DATA_CLK_P] (IN)
                     net (fo=0)               0.973    2.716   worker/dev_data_clk_in[DATA_CLK_P]
                     LUT1 (Prop_lut1_I0_O)    0.124    2.840 r worker/dac_clk_bufr_i_1/O
                     net (fo=1, unplaced)     0.800    3.640   worker/dac_dev_data_clk_p
                     BUFR (Prop_bufr_I_O)     0.537    4.177 r worker/dac_clk_bufr/O
                     net (fo=20, unplaced)    0.584    4.761   worker/dac_clk
                     BUFR (Prop_bufr_I_O)     1.031    5.792 r worker/BUFR_inst/O
                     net (fo=91, unplaced)    0.584    6.376   worker/dev_data_ch1_in_out[dac_clk]
                     FDRE                                  r  worker/data_mode_lvds.reg_duplication_loop[6].dacd2_ch0_i_r_reg[6]/C
------------------------------------------------------------------- -------------------
                     FDRE (Prop_fdre_C_Q)     0.478    6.854 r worker/data_mode_lvds.reg_duplication_loop[6].dacd2_ch0_i_r_reg[6]/Q
                     net (fo=2, unplaced)     0.752    7.606   worker/data_mode_lvds.reg_duplication_loop[6].dacd2_ch0_i_r_reg_n_0_[6]
                     LUT3 (Prop_lut3_I0_O)    0.295    7.901 r worker/data_mode_lvds.dac_data_ddr_first_r[0]_i_3/O
                     net (fo=1, unplaced)     0.000    7.901   worker/dacd2_i_r__0[6]
                     MUXF7 (Prop_muxf7_I1_O)  0.247    8.148 r worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]_i_1/O
                     net (fo=1, unplaced)     0.000    8.148   worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]_i_1_n_0
                     FDRE                                  r  worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]/D
------------------------------------------------------------------- -------------------

                     (clock clk2 fall edge)   5.230    5.230 f
                                              0.000    5.230 f dev_data_clk_in[DATA_CLK_P] (IN)
                     net (fo=0)               0.924    6.155   worker/dev_data_clk_in[DATA_CLK_P]
                     LUT1 (Prop_lut1_I0_O)    0.100    6.255 r worker/dac_clk_bufr_i_1/O
                     net (fo=1, unplaced)     0.760    7.015   worker/dac_dev_data_clk_p
                     BUFR (Prop_bufr_I_O)     0.486    7.501 r worker/dac_clk_bufr/O
                     net (fo=20, unplaced)    0.439    7.940   worker/dac_clk
                     FDRE                                  r  worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]/C
                     clock pessimism          0.163    8.103
                     clock uncertainty       -0.035    8.067
                     FDRE (Setup_fdre_C_D)    0.080    8.147   worker/data_mode_lvds.dac_data_ddr_first_r_reg[0]
-------------------------------------------------------------------
                     required time                     8.147
                     arrival time                     -8.148
-------------------------------------------------------------------
                     slack                             0.000
```