RI_ADPLL_2GMP Datasheet

IP ri_adpll_22fdx

Datasheet

Date: 2021/12/06

Status: initial

Table of Contents

1 Document Structure

1.1 List of Figures

1.2 List of Tables

2 Document Structure

2.1 Revision History

Rev	Date/Editor	Description
0.0.1	2017-06-15, FP	Initial version
0.0.2	2017-09-27, FP	Frequency calculation, book level schematic, input and output discription
0.0.3	2017-12-20, FP	Select Config for divider chain of clk_core
0.0.4	2018-01-16, FP	BISC Mode A removed, Look-In Modes, Frequency Range
0.0.5	2018-04-04, FP	BISC config and status in adpll_config/status included
1.0.0	2019-10-16, FP	Physical Specification
1.0.1	2020-03-12, FP	DCO tuning section
	2021-03-31, LI	added LDO specification
1.1.0	2021-05-18, FP	removed BISC feature, N4 changes, splitted config vector
1.1.1	2021-05-27, LI, FP	added LDO startup time, BIST, status capture
1.1.2	2021-06-21, FP	Coarsetune compensation, BIST splitted in fine and coarse tune
	2021-07-13, LI	updated LDO specification
1.1.3	2021-07-28, FP	added scan_clk_i

3 Introduction

This documentation describes the structure, features and functions of the clock generator for a nominal frequency of 2GHz (0.8V).

Feature summary:

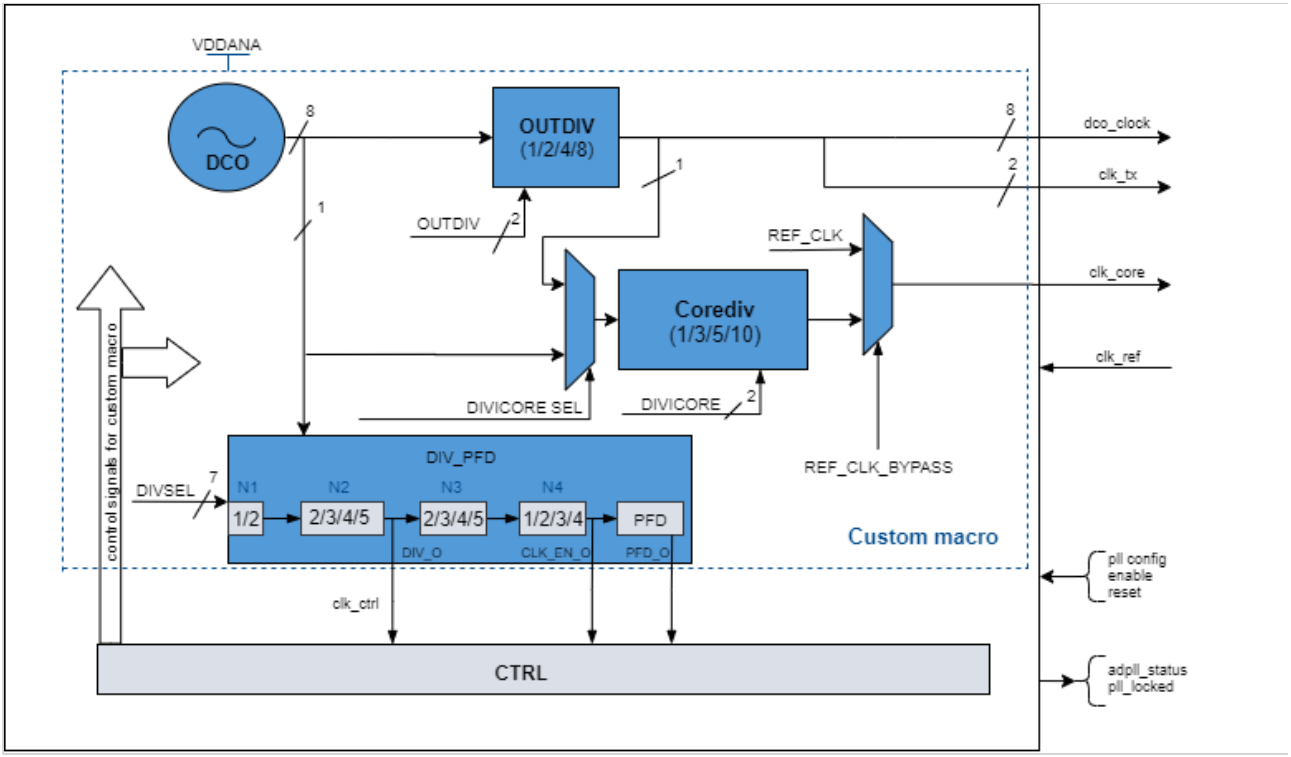
- All-digital phase locked loop (ADPLL) clock generator
- Clock generation based on a digitally controlled oscillator DCO
- Globalfoundries 22FDX technology
- Programmable clock frequency dividers for ADPLL loop and core clock outputs
- Fast lock-in achieved by binary search

4 Clock Generator Circuit Description

4.1 Clock Generator Structure

Figure 1 shows a simplified block level schematic of the ADPLL clock generator.

The ADPLL acts as clock multiplier of a reference clock signal.



1 Block level schematic of clock generator

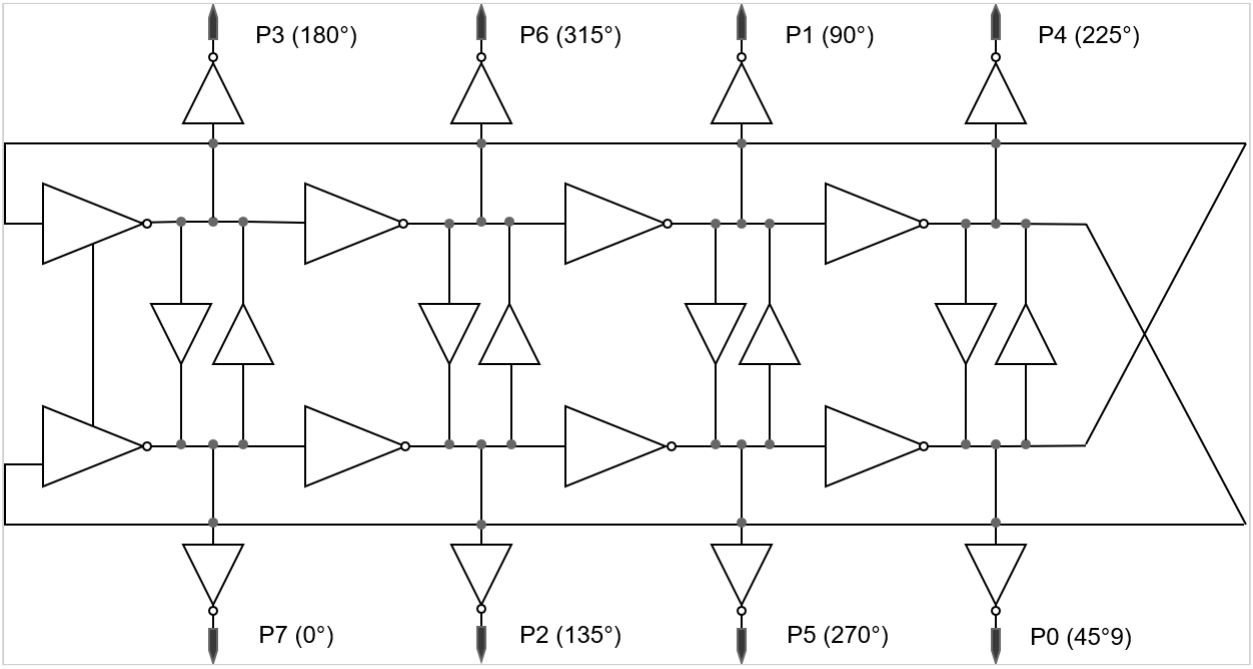
4.2 Digital Controlled oscillator (DCO)

4.2.1 DCO Overview

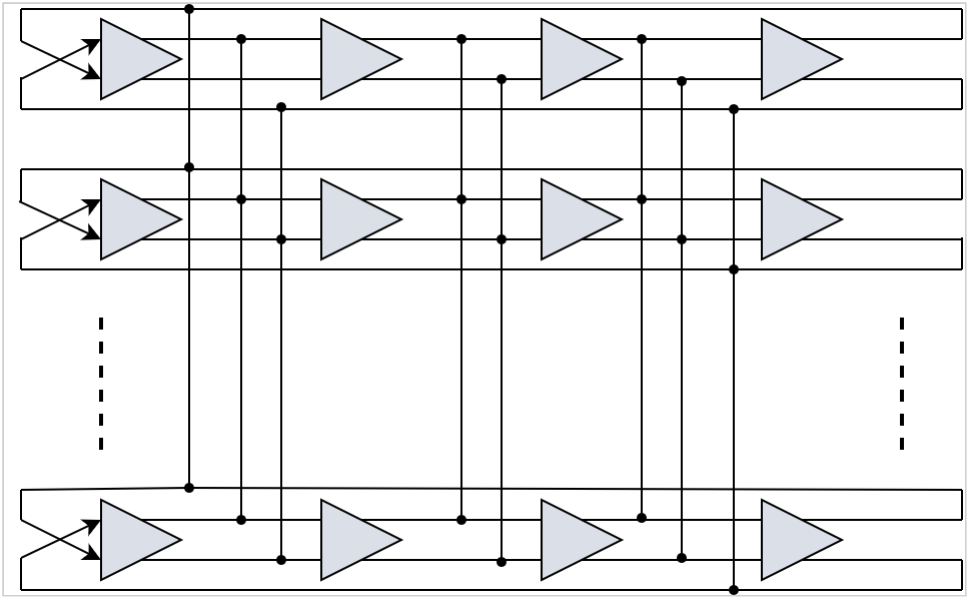
The DCO is based on a ring oscillator architecture (**Figure 2**). The ring oscillator consists of differential inverter elements which has the advantage to have an even number of inverter stages for clock oscillation.
Therefore, the DCO can provide eight clock phase which is important for applications like SERDES or Ethernet IPs.
The cross-coupled inverters ensures a phase shift of 180° for each stage.
For frequency tuning the DCO consists of two tuning mechanism:

- Coarse Tuning
- Fine Tuning

For a coarse frequency tuning several ring oscillators are connected in parallel (**Figure 3**).
A Coarse Tuning Decoder is used to control either which single inverter are switched on and off or which complete inverter stage are enabled and disabled.
If inverters or complete stages are enabled the driving strength increase.
At the same time, capacitive load is nearly constant, therefore the frequency of each phase increase.

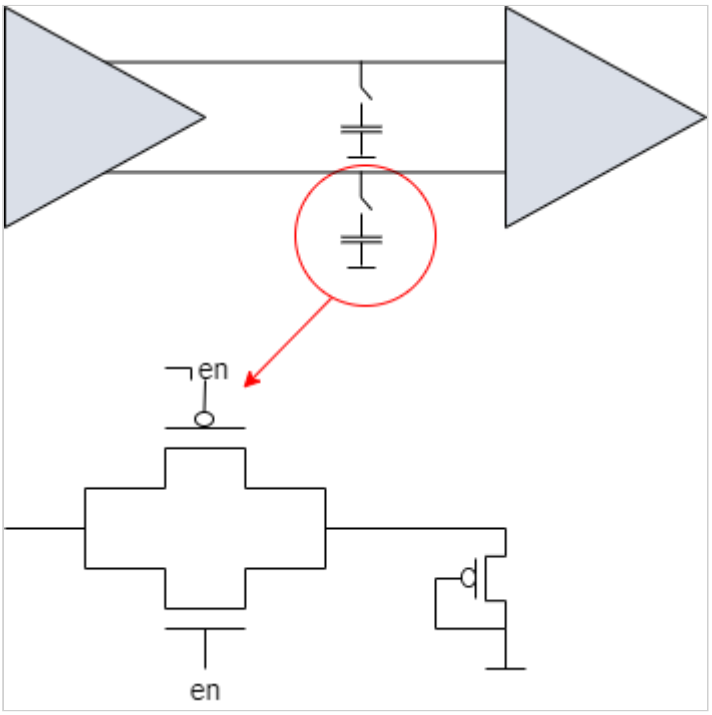


2 DCO ring oscillator structure



3 Ring Oscillator Array for Coarse Tuning

The ring oscillator array consists of eight ring oscillator stages.
The frequency steps of the coarse tuning are not sufficient for several devices such as a SERDES. For this reason, the DCO has an additional fine tune mechanism.
Each inverter stage has fine tune elements (**Figure 4**). These elements are switchable load capacities which can influence the delays of each inverter stage.



4 Fine Tune Capacitance

4.2.2 DCO Tuning

Table **1** shows the inputs on the DCO which controls the frequency tuning.

Tuning Input	Width	Description
CT COLSEL_I	8	Selects the columns from coarsetune array
CT ROWSEL_I	6	Selects the rows from coarsetune array
CT_ROWON_I	6	CT_ROWON_I[5:0] = {1'b0, CT_ROWSEL_I[5:1]}
FT_I	256	Finetune
FBT_I	3	Selects cross coupled inverters

1 DCO Tuning Inputs

In Table **2** the decoding of the coarse tuning is shown. The minimum coarsetune value to fulfil the oscilation condition is four.

Coarsetune value	CT_COLSEL_I	CT_ROWSEL_I	DCO oscillation condition fulfilled?
1	1	1	no
2	2	1	no
3	3	1	no
4	4	1	yes
5	5	1	yes
6	6	1	yes
7	7	1	yes
8	8	1	yes
9	1	2	yes
10	2	2	yes
. . .	. . .	. . .	. . .
47	7	6	yes
48	8	6	yes

2 Coarsetune Decoding

4.3 Available divider configuration

4.3.1 Main divider

Divider	Values	Config Bits
N1	1 2	c_main_div_n1_i=0 c_main_div_n1_i=1
N2	2 3 4 5	c_main_div_n2_i[1:0]=00 c_main_div_n2_i[1:0]=01 c_main_div_n2_i[1:0]=10 c_main_div_n2_i[1:0]=11
N3	2 3 4 5	c_main_div_n3_i[1:0]=00 c_main_div_n3_i[1:0]=01 c_main_div_n3_i[1:0]=10 c_main_div_n3_i[1:0]=11
N4	1 2 3 4	c_main_div_n4_i[1:0]=00 c_main_div_n4_i[1:0]=01 c_main_div_n4_i[1:0]=10 c_main_div_n4_i[1:0]=11

3 Main divider

fDCO = fref • N1 • N2 • N3 • N4

⚠ Note:
The following constraints must be fulfilled:

N3xN4>2 and N3xN4xfref<=500MHz

4.3.2 8-Phase Clock Output frequencies (dco_clock)

Divider	Values	Config Bits
OUTDIV	1 2 4 8	OUTDIV[1:0]=00 → c_out_div_sel_i[1:0]=00 OUTDIV[1:0]=01 → c_out_div_sel_i[1:0]=01 OUTDIV[1:0]=10 → c_out_div_sel_i[1:0]=10 OUTDIV[1:0]=11 → c_out_div_sel_i[1:0]=11

4 OUTDIV

fDCO_OUT = fDCO/OUTDIV

4.3.3 Core frequencies (clk_core)

The core divider ensures a clock duty cycle of 50%

Divider	Values	Config Bits
COREDIV	1 3 5 10	DIVCORE[1:0]=00 → c_divcore_sel[1:0]=00 DIVCORE[1:0]=01 → c_divcore_sel[1:0]=01 DIVCORE[1:0]=10 → c_divcore_sel[1:0]=10 DIVCORE[1:0]=11 → c_divcore_sel[1:0]=11

5 COREDIV

The core frequency depends on the selected divider chain for clk_core. You can either select the divider chain consists of COREDIV and OUTDIV (DIVICORE_SEL=1)
or you can configure to have only the COREDIV as frequency divider (DIVICORE_SEL=0).

The resulting frequency depends on chosen configuration:

1.DIVICOR_SEL=0:

fCORE = fDCO/COREDIV

2.DIVICORE_SEL=1:

fCORE = fDCO_OUT/COREDIV

4.4 Input and Output descriptions

Name	Widh	Direction	Description
testmode_i	1	input	testmode enable signal
scan_enable_i	1	input	scan Mode enable signal
scan_in_i	3	input	data inputs for scan chains
scan_out_o	3	output	data outputs for scan chains
ref_clk_i	1	input	ADPLL reference clock signal
scan_clk_i	1	input	ADPLL scan clock signal
reset_q_i	1	input	asynchronous reset (low-active)
en_adpll_ctrl_i	1	input	enable signal for pll controller

Name	Widh	Direction	Description
clk_core_o	1	output	low speed core clock output
pll_locked_o	1	output	lock signal (frequency + phase)
c_*_i	55	input	Configuration inputs; see Table 5
adpll_status_o	21	output	ADPLL status bits
adpll_status_capture_i	1	input	ADPLL status capture bit, capture at rising edge
dco_clk_o[7:0]	8	output	multi-phase DCO clock output to SerDes AFE
clk_tx_o[1:0]	2	output	TX output clocks (dco_clk_o[7], dco_clk_o[3])
pfd_o	1	output	output signal of bang bang PFD
status_ack_o	1	output	• While status bits are read out status_ack_o is 1'b1 • Status_ack_o is also 1'b1 if en_adpll_ctrl_i==1'b0
bist_busy_o	1	output	1: BIST is running 0: BIST finished
bist_fail_coarse_o	1	output	1: BIST fail for coarsetune (frequency monotony error or BIST was not started correctly) 0: BIST pass
bist_fail_fine_o	1	output	1: BIST fail for finetune (frequency monotony error or BIST was not started correctly); in case bist_fail_coarse_o==1'b1 also the BIST for finetune will fail as finetune BIST will not be started in case of coarsetune BIST error 0: BIST pass

6 ADPLL ports

4.5 BIST

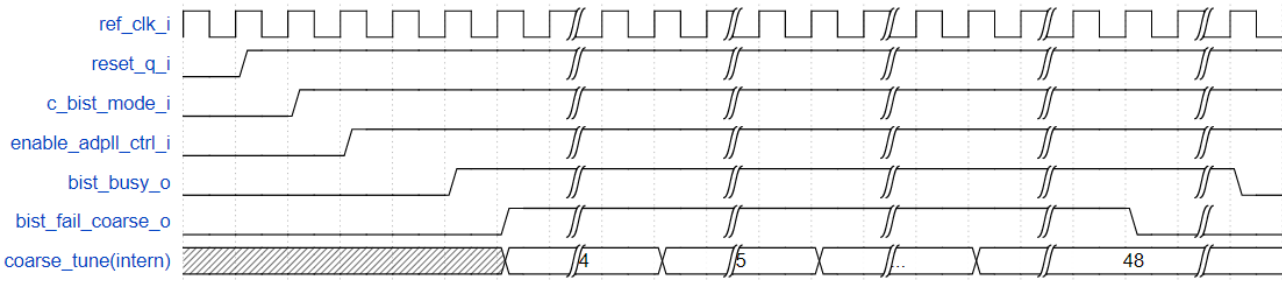
The PLL has an integrated BIST (Build In Self Test). It checks that the coarse and fine tune has an monotonic increasing of frequency.

The output *bist_busy_o* shows if BIST is still running or already finished. It is set synchronous to *ref_clk_i.*

After BIST is finished you can see if BIST passed or failed based on output *bist_fail_coarse_o/bist_fail_fine_o*. You can assume that bist fail bits are statical after BIST is finished.

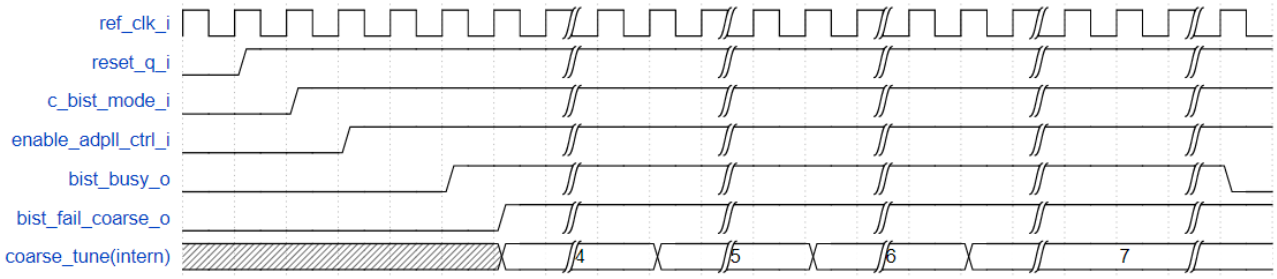
The BIST feature is designed for a reference frequency of 100MHz.

The following figure shows a BIST run (only for coarse tune) where the test passed:



5 BIST timing diagram without tune error

On the other hand you can see a BIST (also only for coarse) where you have an error with the tuning word of 7.



6 BIST timing diagram with coarse tune error with coarse tuning word 7

4.6 Lock-In Modes

The ADPLL provides different modes for lock-in of phase and frequency as summarized in following table:

Coarse Tune Enable	Fast Lock Enable	Description
0	0	Coarse tune value from configuration is used; Fine tune is determined by closed-loop ADPLL operation. No binary frequency search. Long lock-in time.
0	1	Coarse tune value from configuration is used; Initial fine tune value is determined by binary search, and then tuned by closed-loop ADPLL operation. Fast-lock-in. Note that in this mode the DCO is not continously running.
1	0	Automatic binary search of coarse tune value; Fine tune is determined by closed-loop ADPLL operation (start value: internal reset value). No binary fine tune frequency search. Long lock-in time. This is the default setting for using PLL in combination with an LDO.

Coarse Tune Enable	Fast Lock Enable	Description
1	1	Automatic binary search of coarse tune value; Initial fine tune value is determined by binary search, and then tuned by closed-loop ADPLL operation. Fast-look-in. Note that in this mode the DCO is not continously running.

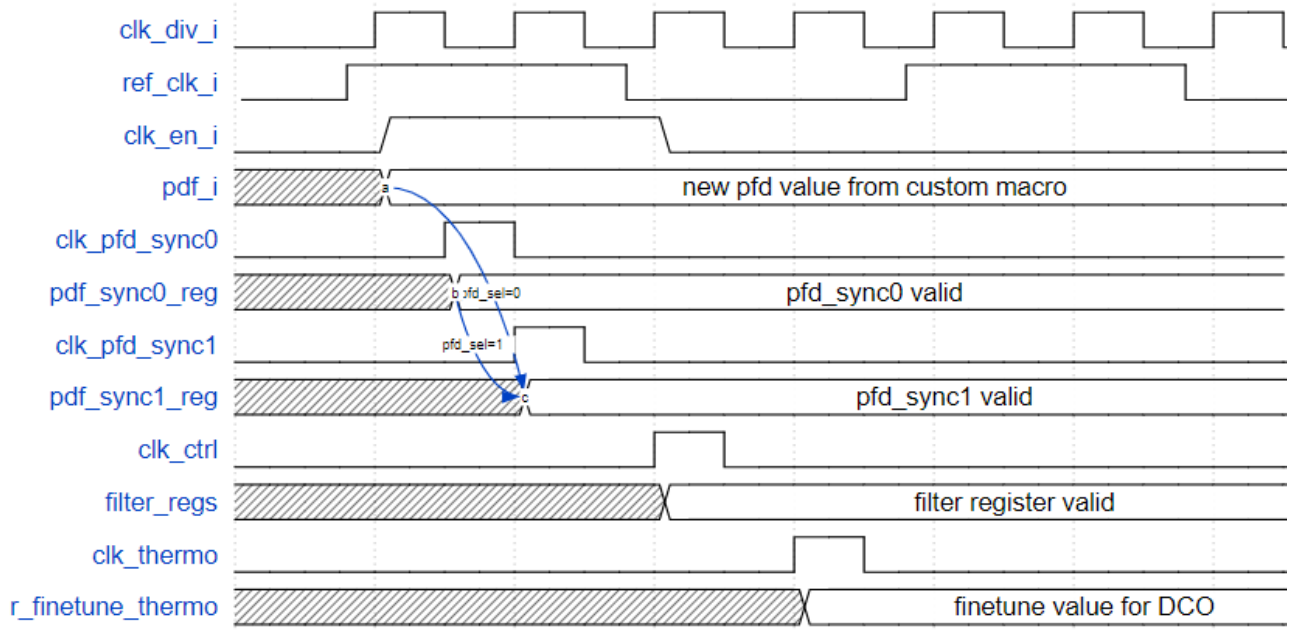
7 Lock-In Modes

4.7 ADPLL Configuration and Status bits

Name	Bit width	default value	ADPLL config bit	Description
CI	5	4	c_ci_i[4:0]	integral coefficent of loop filter
CP	8	48	c_cp_i[7:0]	proportional coefficient of loop filter
MAIN_DIV_N1	1	1	c_main_div_n1_i	ADPLL main loop divider N1 c_main_div_sel_n1_i: 0:N1=1 1:N1=2
MAIN_DIV_N2	2	0	c_main_div_n2_i[1:0]	ADPLL main loop divider N2 c_main_div_sel_n2_i[1:0]: 00:N2=2 01:N2=3 10:N2=4 11:N2=5
MAIN_DIV_N3	2	2	c_main_div_n3_i[1:0]	ADPLL main loop divider N3 c_main_div_sel_n3_i[1:0]: 00:N3=2 01:N3=3 10:N3=4 11:N3=5
MAIN_DIV_N4	2	0	c_main_div_n4_i[1:0]	ADPLL main loop divider N4 c_main_div_sel_n4_i[1:0]: 00:N4=1 01:N4=2 10:N4=3 11:N4=4
OUTDIV	2	0	c_out_div_sel_i[1:0]	ADPLL output divider: 00 Divide by 1 01 Divide by 2 10 Divide by 4 11 Divide by 8
OPEN_LOOP	1	0	c_open_loop_i	ADPLL open-loop mode: 0: disabled; 1: enabled
FT	8	128	c_ft_i[7:0]	ADPLL fine-tune value for open-loop mode and for automatic coarse tune
DIVCORE	2	0	c_divcore_sel_i[1:0]	frequency divider setting for core output clock: 00 Divide by 1 01 Divide by 3

Name	Bit width	default value	ADPLL config bit	Description
				10 Divide by 5 11 Divide by 10
CT	6	8	c_coarse_i[5:0]	Coarse tune value for open loop(CT>=4!) and in the case AUTO_CT=0
AUTO_CT	1	1	c_auto_coarsetune_i	Enable for automatic coarse tune
ENFORCE_LOCK	1	0	c_enforce_lock_i	overwrite locked-bit
PFD_SEL	1	0	c_pfd_select_i	Per default pfd signal from custom macro is synchronized with one synchronizer stage. By enabling c_pfd_select_i (1'b1) you can add an additional stage. See figure "PDF synchronization"
LOCK_WINDOW	1	0	c_lock_window_sel_i	Lock Detection Window: 0: Long; 1: short
DIVCORE_SEL	1	0	c_div_core_mux_sel_i	selects the divider chain for clk_core 0: COREDIV 1: OUTDIV + COREDIV
FILTER_SHIFT	2	2	c_filter_shift_i	shift filter coefficients CI and CP left by FILTER_SHIFT when not locked
FAST_LOCK	1	1	c_en_fast_lock_i	enable internal fast lock-in for fine tune value using binary search
SAR_LIMIT	3	0	c_sar_limit_i	limit for binary search fast lock-in
SET_OP_LOCK	1	0	c_set_op_lock_i	force locked bit to 1 in OP mode. This mode bypasses the lock detection.
DISABLE_LOCK	1	0	c_disable_lock_i	0: normal operation 1: force locked bit to 0
REF_BYPASS	1	0	c_ref_bypass_i	enable/disable reference clock bypass to ADPLL core clockBIST output
BIST_MODE	1	0	c_bist_mode_i	BIST mode 1:BIST; 0: normal PLL usage
COARSETUNE_COMPENSATION	1	0	c_ct_compensation_i	if enabled: in case of finetune underflow/overflow coarsetune value is automatically decreased/increased

8 ADPLL configuration



7 PFD synchronization

Name	Bits	Mapped to ADPLL pin	Discription
PLL_LOCKED	1	adpll_status_o[0]	locked bit
PLL_FT_OF	1	adpll_status_o[1]	overflow flag for fine tune
PLL_FT_UF	1	adpll_status_o[2]	underflow flag for fine tune
PLL_FT	8	adpll_status_o[10:3]	fine tune Value
PLL_STATE	2	adpll_status_o[12:11]	ADPLL FSM state
PLL_COARSE TUNE	6	adpll_status_o[18:13]	coarse tune value
PLL_CT_OF	1	adpll_status_o[19]	overflow flag for coarse tune
PLL_CT_UF	1	adpll_status_o[20]	underflow flag for coarse tune

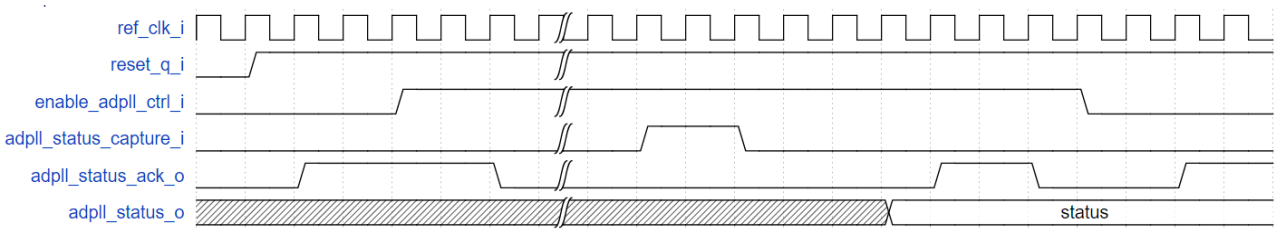
9 ADPLL Status bits

4.8 Status Capture

To get the current status you have to set *adpll_status_capture_i* which is internally synchronized to capture the status.

With the rising edge of *adpll_status_ack_o* the status data are valid (see figure 7).

The acknowledge bit *adpll_status_ack_o* is synchon to *ref_clk_i* but not the status data itself.



8 Status capture

5 Physical Specification

5.1 Dimension

Width in µm	Height in µm	Area in µm2
52.2	89.6	4677.12

6 Electrical Specification

6.1 Operating conditions

Condition	Unit	min	typ	max
Temperature	°C	-40	25	150
Process		ssg	tt	ffg
VNW	V	0.0	0.0	0.0
VPW	V	0.0	0.0	0.0
0.80V operation				
VDD	V	0.76	0.80	0.84

6.2 Reference Clock Frequency

Nominal Voltage	Min. REF Frequency [MHz]	Max. REF Frequency [MHz]
0.80V operation	25	150

6.3 Lock-In Time

The following table shows lock-in times for different reference frequencies without fast lock feature

Reference frequency	Lock-In Time
100MHz	<25us
50MHz	<45us
25MHz	<85us

6.4 DCO Frequency range

Nominal Voltage	Min. DCO Frequency [MHz]	Max.DCO Frequency [MHz]
0.80V operation	800	2400

6.5 DCO Tuning Step Size

Condition	TDCO [ns]	CT value	Average FT stepsize [ps]		
wc_0d76V_150C			1.250	16	2.2
			0.416	48	0.6
tc_0d80V_25C			1.250	12	2.1
			0.416	35	0.7
bc_0d84V_m40C			1.250	10	2.1
			0.416	27	0.7

6.6 Jitter

DCO Period Jitter examples

Accumulated Jitter examples: <15ps

6.7 Power

Condition	DCO	Digital Controller	Sum	
zbb_bc_0d84V_150C		@800MHz: 0.86mW	1.08mW	
		@1500MHz: 1.18mW		
		@2400MHz: 1.61mW		

7 LDO Specification

7.1 Physical Specification

Width in µm	Height in µm	Area in µm2
69.00	60.00	4140

Pins are located in C3 metal.

All devices are within the Deep N-Well (T3) which is biased at VDDA18. The p-substrate connected to VSSA and enclosed by SXCUT.

7.2 Functional Specification

7.2.1 Power pins

Pin	type	description
VDD	power	Always-on power domain
VSS	ground	Ground for always-on power domain
VDDA18	power	Nominal 1.8V input voltage
VDD_DCO	power	Nominal 0.8V output voltage
VSSA	ground	Ground (shared for VDDA18 and VDD_DCO)

7.2.2 Functional pins

Pin	type	direction	power domain	description
ldo_en_i	signal	in	VDD/VSS	LDO enable. If high, the regulated VDD_DCO output is enabled
bypass_en_i	signal	in	VDD/VSS	bypass mode enable. If high, VDDA18 will be bypassed to VDD_DCO.
vref06_i	analog	in	VDDA18/VSSA	analog reference voltage

- ldo_en_i and bypass_en_i may not be active at the same time.
- bypass_en_i is safety critical, as it may cause an VDD_DCO voltage level that may be above spec and could damage connected circuits.

7.3 Electrical Specification

7.3.1 Operating conditions

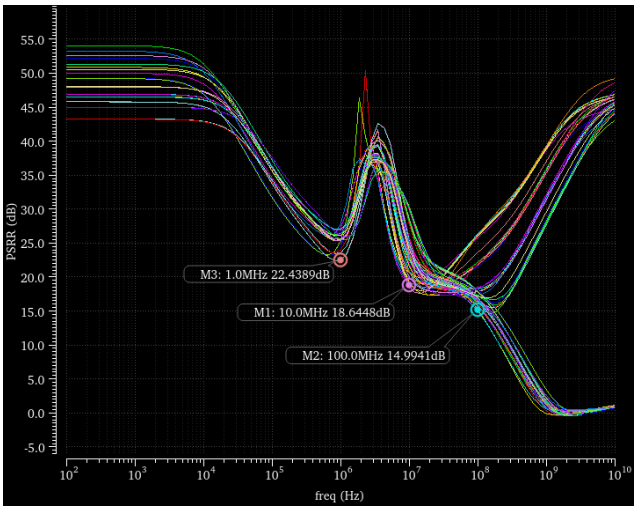
Condition	Unit	min	typ	max	note
Temperature	°C	-40	25	150	
Process		ssg	tt	ffg	
VNW	V	0.0	0.0	0.0	internally tied to VSS
VPW	V	0.0	0.0	0.0	internally tied to VSS
VDDA18	V	1.71	1.80	1.89	when in LDO mode

Condition	Unit	min	typ	max	note
	V	1.00			when in bypass mode
vref06_i	V	0.576	0.6	0.624	= +/-4%
Load current	mA	0.5	1.5	2.5	required: up to 1.5mA for PLL and 0.6mA for PI
Load capacitance	pF	0		40	

Additionally to the load capacitance, a 10...100pF probe can be connected using a 5...15kOhm series resistor.

7.3.2 **Performance**

Condition	Unit	min	typ	max	note
VDD_DCO steady state value	V	0.76	0.8	0.84	Includes +/-4% from untrimmed vref06_n + 3sigma<8mV
PSRR at 10MHz	dB	18.6			
PSRR at 100MHz	dB	14.9			
Static internal losses	uA		150	200	LDO on
Load regulation	mV/mA			0.1	average steepness of V-I-plot in allowed current range
Startup time	us			2	from bypass_en_i 0->1 until VDD_DCO is settled, assuming that VDDA18 and vref06_i are already stable (longer when external probe is connected)
Leakage from VDDA18	uA			2.6	LDO off, VDD_DCO tied to 0.0V
Bypass switch resistance	Ohm			120	(preliminary)



9 PSRR plot (post-layout corner simulation))