PLL Circuit for the Evaluation of Analog Defect Simulation Tools

Mailto: adstestsuite@infineon.com

May 24, 2021

Note:
The CMOS transistor models used, were freely available models downloaded from: http://ptm.asu.edu.

Contents

List of Figures

List of Tables

Abstract:

This document describes the PLL and its test suite. It should enable the reader to perform defect simulations on the PLL circuit and to adapt the testbench if needed.

Audience: Tool evaluators

1 Description of the PLL Circuit

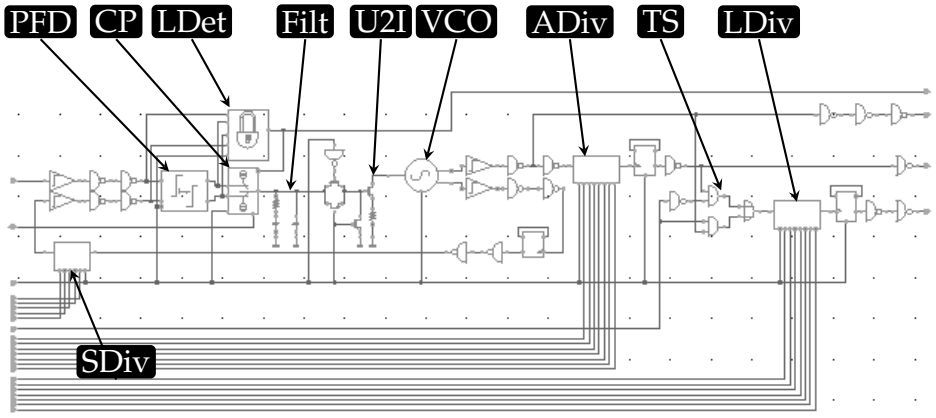

Figure 1: PLL schematic overview.

Figure 1 shows the top level schematic of the PLL. The PLL is used as a clock generation block for different circuits, it generates three clocks: a system clock that runs nominally at 206.4 MHz, an ADC clock that runs nominally at 20.6 MHz, and finally an LDO clock running at 413 kHz.

The individual blocks of the PLL are annotated in Figure 1, the block PFD is the Phase-Frequency-Detector that measures the phase difference between the divided system clock and the reference clock, the measured phase difference controls the charge pump CP, the output of which is fed into the block Filt wich filters the charge pump output. The filtered voltage controls the gate of the NMOS in the U2I stage which generates a current to control the oscillation frequency of the VCO. The VCO output is then fed into the frequency divider SDiv which generates a lower (divided) frequency. This lower frequency is then fed into the PFD which closes the control loop.

Additional circuit elements pertain to the generation of the needed frequencies, the signal coming out of the VCO is divided (Nominal division ratio: 14) to $\approx$ 20.5 MHz by ADiv to generate the ADC-Clock, the ADC-Clock is further divided by LDiv to generate the LDO-Clock, as deriving it directly from the Sys-Clock would imply a high division ratio that would increase the needed area for the circuit.

The block labelled LDet is the lock detector, it asserts the Lock signal when the PLL has reached a locked state. It is not needed for the function of the PLL but is important for its use, as the system can only rely on the clocks generated by the PLL until the PLL is in the locked state. This means that circuits using clocks generated by the PLL should only be

enabled when Lock is asserted. To speed up the locking of the PLL, the charge pump runs at a higher current as long as the PLL is not in lock, after locking the current is reduced to increase the PLL's phase margin and reduce the Jitter of the generated clocks.

The TS block allows to feed the undivided system clock to the LDO frequency divider, it is used during testmode, as measuring the frequency of an $\approx$ 400 kHz clock is consuming. Therefore, during test, the frequency divider is fed with the system clock to reduce test time.

Table 1: List of module signals.

Name	Type	Min.	Nom.	Max.	Unit
VDD	Power	1.35	1.5	1.65	V
fref	Input	-	7.373	-	MHz
disable	Input	0		VDD	V
ibias5u	Input	4.5	5.0	5.5	μA
lock	Output	0		VDD	V
divtest	Input	0		VDD	V
sys_clk	Output	0		VDD	V
sysdiv$\{4\cdots0\}$	Input	0		VDD	V
adc_clk	Output	0		VDD	V
adcdiv$\{6\cdots0\}$	Input	0		VDD	V
ldo_clk	Output	0		VDD	V
ldodiv$\{6\cdots0\}$	Input	0		VDD	V

2 Description of Operation of the PLL circuit

The nominal operation of the PLL is to generate a 206.4 MHz system clock, an ADC clock at 20.6 MHz and an LDO clock at 413 kHz.

The nominal reference frequency is set to be 7.373 MHz.

Please note that any changes to the reference frequency will lead to a change in the frequency response of the PLL. Therefore the loop filter has to be adapted in order to place the poles and zero accordingly for the whole loop to remain stable and still have acceptable transient characteristics.

The generated frequencies can be set using the appropriate dividers according to the following table:

Table 2: Frequency Settings & Divider Values.

Signal	Equation	Divider	Min.	Nom.	Max.
`sys_clk`	$2 \cdot f_{ref} \cdot$ `sysdiv`	`sysdiv`	00100_b	01110_b	11111_b
`adc_clk`	$\frac{f_{sys}}{2} \cdot$ `adcdiv`	`adcdiv`	0000100_b	0000101_b	1111111_b
`ldo_clk`	$\frac{f_{adc}}{2} \cdot$ `ldodiv`	`ldodiv`	0000100_b	0011001_b	1111111_b

Please note that the frequency setting equation for `ldo_clk` is only valid when `divtest` $= 0$.

2.1 Description of Some Individual Signals and Their Effects

`fref`: This is the reference frequency of the PLL. Any changes to its value should be followed by a stability verification of the loop.

`disable`: When set to 1 this signal disables all PLL modules. All signals are set to 0 when the PLL is disabled.

`ibias5`μ: This is the biasing current of the charge pump. As it influences the pole position of the loop, any changes to its values should be followed by a stability verification of the loop.

`lock`: This signal goes high when the PLL reaches the locked state. After each change of the `sysdiv` the PLL will lose lock, thus `lock` should go low and return to high as soon as the PLL has locked to the new frequency.

`divtest`: This signal is used during test. If it is set to high, the divider generating the `ldo_clk` signal is fed by the undivided `sys_clk` allowing for a faster test of its clock divider, especially useful when the ldo clock divider is set to 1111111_b.

`sys_clk`: This is the main system clock generated by the PLL. Its nominal value is: 206.4 MHz.

`sysdiv{4···0}` : Binary divider inputs to the system clock divider. The LSB is `sysdiv0`. The minimum value that the divider operates at is 4.

`adc_clk`**:** This is the ADC clock generated by the PLL. Its nominal value is: 20.6 MHz.

`adcdiv{6···0}` : Binary divider inputs to the ADC clock divider. The LSB is `adcdiv0`. The minimum value that the divider operates at is 4.

`ldo_clk`**:** This is the LDO clock generated by the PLL. Its nominal value is: 413 kHz.

`ldodiv{6···0}` : Binary divider inputs to the LDO clock divider. The LSB is `ldodiv0`. The minimum value that the divider operates at is 4.

3 Description of The Testbench

The testbech goes sequentially through the following stages and exits at the first failed test it encounters:

1. After powerup the `disable` signal is asserted, the clock is started. The first measurements check that all output signals are logical zero (`gnd`)

   Fail if: Any output signal $\geq V_{\mathrm{dd}}/2$

2. The PLL is enabled and and all dividers are set to maximum values. The `divtest` signal is asserted and connects the input of Ldiv to `sys_clk`. After 10 μs the testbench starts checking for lock every 100 ns. When `lock` goes high, the frequencies at `sys_clk`, `adc_clk`, and `ldo_clk` are computed

   Fail if: `lock` doesn't go high within 15 μs of the start of lock check, or if the frequencies at `sys_clk`, `adc_clk`, and `ldo_clk` do not comply to the equations of Table 2 within 2%. Please note that in this case `divtest` is set, and that the input to Ldiv is not $f_{\mathrm{adc}}/2$ but the undivided f_{sys}

3. All dividers are set to minimum values. The `divtest` signal is deasserted. After 10 μs the testbench starts checking for lock every

100 ns. When `lock` goes high, the frequencies at `sys_clk`, `adc_clk`, and `ldo_clk` are computed

Fail if: `lock` doesn't go high within 15 μs of the start of lock check, or if the frequencies at `sys_clk`, `adc_clk`, and `ldo_clk` do not comply to the equations of Table 2 within 2%.

4. All dividers are set to typical values. The `divtest` signal is de-asserted. After 10 μs the testbench starts checking for lock every 100 ns. When `lock` goes high, the frequencies at `sys_clk`, `adc_clk`, and `ldo_clk` are computed

   Fail if: `lock` doesn't go high within 15 μs of the start of lock check, or if the frequencies at `sys_clk`, `adc_clk`, and `ldo_clk` do not comply to the equations of Table 2 within 2%.

5. The `disable` signal is asserted again. The testbench checks whether all PLL output signals go to logical 0 (`gnd`)

   Fail if: Any output signal $\geq V_{dd}/2$