# iniADPLL

***All Digital Phase Locked Loop***

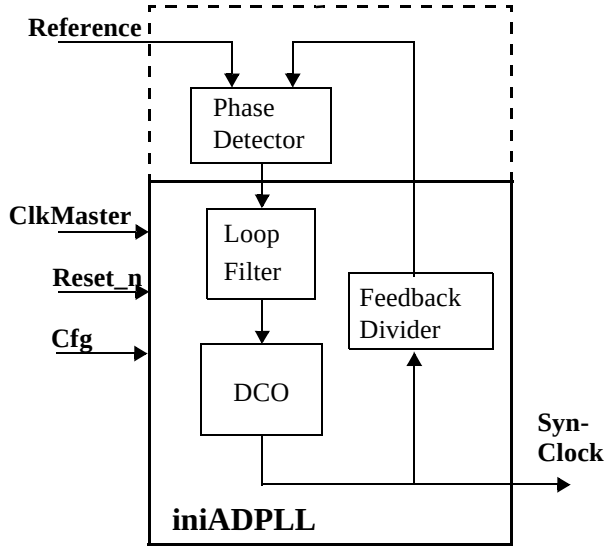
## Features

- All Digital Implementation
- Linear Frequency Range from 0 to n MHz
- Infinite Frequency Hold Time
- Programmable Center Frequency
- Programmable Filter Characteristics (Cut-Off Frequency, Loop Gain)
- Adaptable Phase Detector
- Structured, Synchronous HDL RTL Design
- Gate Count: ~5000
- 100% Technology Independent

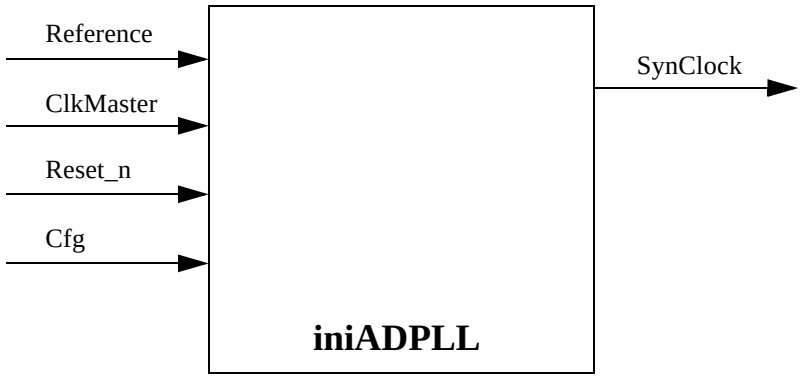
## General Description

The iniADPLL is an all digital implementation of a phase locked loop. PLLs are widely used in telecom applications for clock recovery, clock generation and clock supervision. The all digital solution needs no external components, is fully programmable and can be started already locked. Other advantages can be found in application where long time constants are required or where the loop filter have to be programmed depending on current operating conditions.

Different phase detectors (FIFO fill level, phase errors, etc.) may be used and can be adapted to perfectly fit the application.




Individual implementations usually differ considerably with respect to the phase detector and loop filter characteristics. To help to select the optimal implementation we provide an ADPLL questionnaire.




## I/O Description

This section describes the iniADPLL core input and output ports, and provides a short overview of their functionality.

### General Inputs

Two pins (ClkMaster and Reset_n) are used to clock and initialize the entire iniHDLC core. There are no other clocks used in this core.

| Pin Name | Type | Description |
|---|---|---|
| ClkMaster | in | System clock for the whole iniUART |
| Reset_n | in | Asynchronous system reset, active LOW |

### Configuration

The configuration pins are static inputs used to set the iniADPLL operating mode.

| Pin Name | Type | Description |
|---|---|---|
| Cfg.CenterFreq[1] | in | Center frequency of DCO |
| Cfg.TimeConst[1] | in | Time Constant of loop filter |
| Cfg.LoopGain[1] | in | Gain of loop filter |
| Cfg.FeedbackDiv[1] | in | Feedback divider |
| Cfg.PLL_Run | in | '1': PLL on<br>'0': PLL off: DCO runs at center frequency |

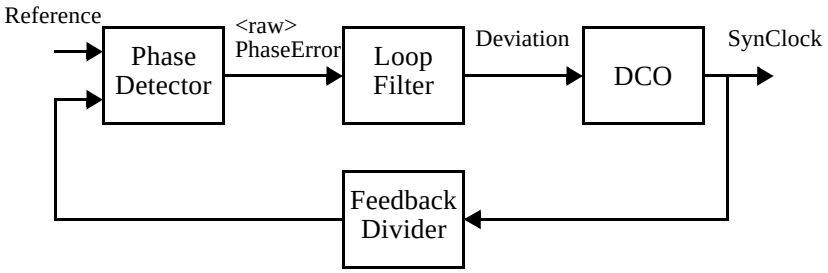
[1]) The signal width depends on the required system parameters.

### Inputs and Outputs

| Pin Name | Type | Description |
|---|---|---|
| Reference[1] | in | Reference input for phase detector |
| SynClock | out | Synthesized clock signal |

[1]) The signal width depends on the required system parameters.

## System Description




The phase-detector is not part of the standard iniADPLL core. A customer specific phase-detector may be added on demand.

**System Clock**

Note that the stability of this clock source directly influences the jitter-performance of the `SynClock` output, as any `ClkMaster` jitter adds to the basic DCO Jitter, which is - by principle - half a `ClkMaster` cycle peak-peak.
The precision of the `ClkMaster` is important because the precision of the deviation limit entirely depends on the precision of the `ClkMaster`.

For application where the DCO can not be used (e.g. high frequencies), an external VCO may be used.

**Phase Detector**

The phase-detector should present permanently a valid phase-difference signal. Therefore a sample-and-hold type phase detector is to be used. `<raw>PhaseError` is sampled by the iniADPLL core on the rising clock edge of `ClkMaster`.
For typical applications it is recommended to use a phase-detector which measures the phase difference in ‘unit intervals’ (UI), where one UI corresponds to one `ClkMaster` period.

**Loop Filter**

This implementation consist of a first order low-pass filter. An architecture was selected which provides for all time-constants a dc-gain of unity. For adjustment of the PLL-loop gain there is a separate function providing a proportional gain.

An other function provides the required setup to select the loop filter time constant.

**Digital Controlled Oscillator**

The DCO is a programmable clock synthesizer. Its center frequency is programmable. Further the DCO can be forced to run at the programmed center frequency (Cfg.PLL_Run=’0’).

## Questionnaire

The iniADPLL-Core can be easily adapted by INICORE to specific customer requirements. Individual implementations usually differ considerably with respect to the phase detector and loop filter characteristics.

With this questionnaire we can rapidly consider your requirements.

### Digitally Controlled Oscillator (DCO)

| Parameter | Value | Units | Tolerance |
|---|---|---|---|
| Available local Clock(s) | | | |
| Required Center Frequency of DCO | | | |
| Do you wish a software-configurable Center Frequency?<br>If yes, specify range and resolution. | | | |
| Required Deviation Range of DCO | | | |
| Do you wish a software-configurable deviation limit?<br>If yes, specify range and resolution. | | | |
| Tolerable edge jitter:<br><br>a) observation period = 1 us<br><br>b) observation period = 100 us | | | |

### Loop Filter and Loop Gain

| Parameter | Value | Units | Tolerance |
|---|---|---|---|
| The loop filter and loop gain parameters are usually adjusted for critical damping.<br>Is this o.k.? | | | |
| Desired corner frequency of the jitter transfer function. | | | |
| Do you wish software programmable parameters for loop-filter time-constant and loop-gain?<br>If yes, specify range relative to the nominal corner-frequency and critical damping. | | | |

## Phase Detector

| Parameter | Value | Units | Tolerance |
|---|---|---|---|
| You may implement the phase detector (e.g. buffer fill level) yourself. Do you want that?<br><br>If yes, specify range of phase-error. | | | |
| Do you need a feedback divider?<br><br>If yes, do you wish to include it in the ADPLL core?<br><br>If yes, specify division ratio. | | | |
| What is the repetition interval of the phase-detection process? | | | |
| What is the maximum jitter amplitude of the phase-error? | | | |
| What is the maximum jitter amplitude where the linear jitter filtering model must apply? | | | |

## Initialization and Lock-In

If the time-constants are very large, lock-in may take an excessive time (up to minutes). In many applications, this can be avoided by proper initialization. With some of these methods we may even specify that the PLL is never out of lock, which considerably simplifies the phase-detector.

- Method 1: Defined Start-up of DCO and Loop-Filter
  During initialization, the Loop-Filter is preset to the nominal DCO frequency. The feedback-divider, if available, is synchronized to the incoming phase reference. The initialization may take place during power-on reset, or may be controlled by a synchronous signal during operation. The sync signal is suitable for holding the PLL in a defined state as long as no valid reference signal is available.

- Method 2: Loop-Filter Hold
  In some systems invalid phase information may occur intermittently. If this is detectable, it is helpful to freeze the Loop-Filter until valid phase information becomes available.

| Parameter | Yes/No | Comment |
|---|---|---|
| Do you want synchronous initialization of DCO and Loop-Filter? | | |
| Do you want to freeze the Loop-Filter state? | | |
| Do you need a phase/frequency detector providing lock-in?<br>(If no, this means that thanks to controlled initiations the PLL is never out of lock). | | |