University of North Carolina at Charlotte
Department of Electrical and Computer Engineering

Qorvo Radio

ADF4351 LO C++ Configurator

Contents

Module Overview

The **ADF4351 LO C++ Configurator** is a C++ class designed to generate the necessary register settings for configuring an Analog Devices ADF4351 fractional-N PLL frequency synthesizer chip. This software simplifies calculating frequency settings required to program the ADF4351, supporting both integer-N and fractional-N modes for synthesizing precise frequencies from 35 MHz up to 4.4 GHz.

Class Functionality

The class computes six 32-bit registers (R0–R5) based on user-defined parameters and frequencies. It allows simple integration into systems requiring dynamically adjustable local oscillator frequencies, such as software-defined radios (SDR).

Usage Steps

1. Instantiation

$ADF4351\ synth(10e6, false, false, 1);$

- This initializes the synthesizer class for a 10 MHz reference frequency without doubling or dividing and an R-counter of 1.

2. Set Frequency

$bool\ success\ =\ synth.setFrequency(desiredFreqHz, channelSpacingHz);$

- For integer-N mode, set `channelSpacingHz` to 0.
- For fractional-N mode, provide a nonzero channel spacing.

3. Retrieve Registers

$ADF4351::Regs\ regs\ =\ synth.getRegisters();$

- Registers (R0–R5) are returned for use in hardware configuration.

4. Program Hardware

- Write registers sequentially from R5 down to R0 via SPI to configure the hardware PLL device.

Operational Details

- Automatically selects optimal output divider for VCO frequency range (2.2–4.4 GHz).

- Determines integer and fractional components (INT, FRAC, MOD) for PLL programming.
- Supports prescaler selection based on INT value.

Key Parameters and Functions

1. Constructor Parameters

- `refFreqHz`: Reference frequency in Hz (default 10 MHz).
- `enableRefDoubler`: Enable internal reference frequency doubler.
- `enableRDIV2`: Enable internal reference divide-by-2.
- `RCounter`: Reference counter division factor.

2. Methods

- setFrequency(double freqHz, double channelSpacingHz): Configures frequency.
- getRegisters(): Retrieves the register configuration for hardware.

Error Handling

- Returns a boolean status from setFrequency() indicating success or failure based on frequency compatibility with the synthesizer.