Echo Cancellation Core (4-lag) Based on LSM Algorithm

Ankai Liu

2018-09-06

Abstract

This document is a user manual for echo cancellation core (4-lag) based on LSM algorithm.

1 Introduction

This document is a user manual for echo cancellation core (4-lag) based on LSM algorithm. The core is designed on CYCLONE IV FPGAs. It is serving for a SPECIFIC signal type (16 bit binary signal with the first digit represent the sign) and a specific echo type (echo caused by transmission within the circuit).

In the project, we have implemented 4-lag and 16-lag cores that takes continuous sampling. The whole core only require gate-logic calculations. We will provide 3 test bench in order to cover all submodules for potential future modifications.

2 echo_cancelation_full

Top level module.

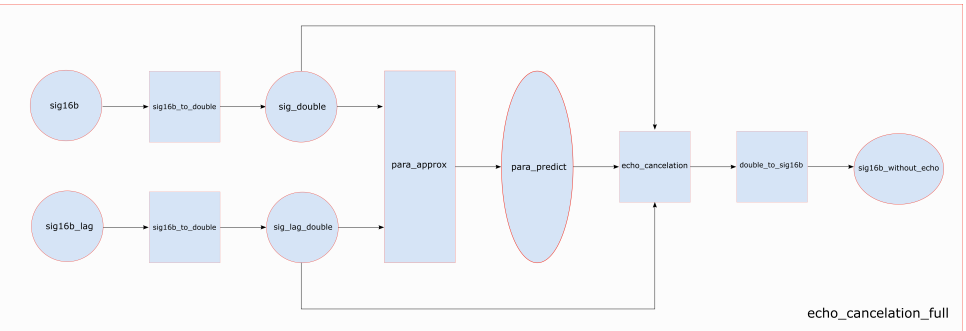

Figure 1: echo_cancelation_full hierarchy

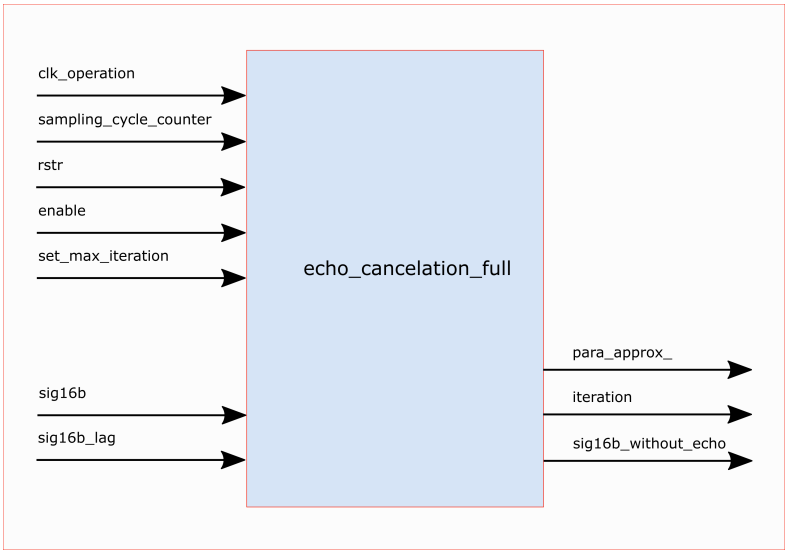

Figure 2: echo_cancelation_full

2.1 Input

clk_operation: Global operation clocks.

sampling_cycle_counter: Global sampling clocks.

rst: Global reset.

enable: Local input. Needs to stay 1 when using the module.

set_max_iteration: Local input. Set to be the numbers of iterations that users wants to achieve.

sig16b: Original signal from sender in 16 bits binary formate.

sig16b_lag: Signal with lag fro receiver in 16 bits binary formate.

2.2 Output

para_approx: Approximate parameters.

iteration: Numbers of iterations that has been done.

sig16b_without_echo: Approximation of original signal/signal without the echo in 16 bits binary formate.

2.3 Important

- The echo_cancelation_full core takes new sample when "sampling_cycle_counter = 0". The reset cycle of sampling_cycle_counter must be more than 1200 of the operation clks/600 of operation cycles.
- The recommenced maximum iteration is 64 for lag 4. See LSM_algorithm_demo.pdf for details.

3 sig16b_to_double

Transforming 16 bit binary signal to double.

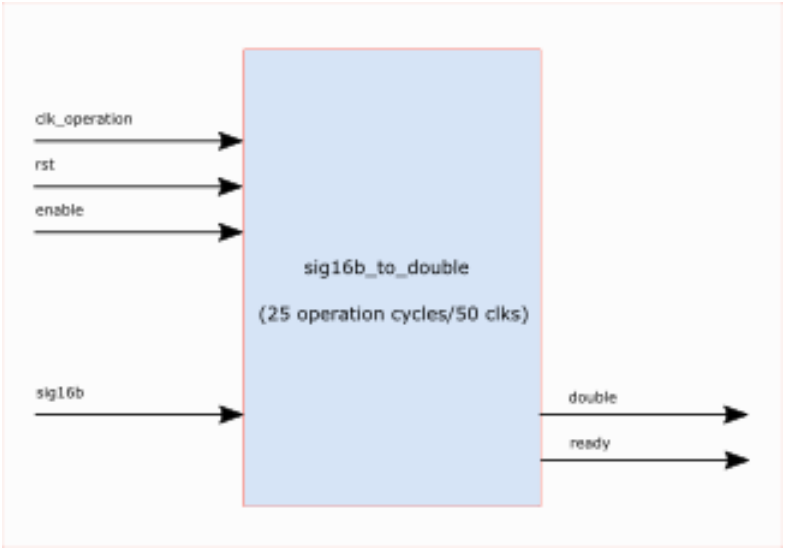

Figure 3: sig16b_to_double

3.1 Input

clk_operation: Global operation clocks.

rst: Global reset.

enable: Local input. Turn on for 2 operation cycles/4 operation clks and then turn off.

sig16b: Input signal in 16 bit binary formate.

3.2 output

double: Output signal in double.

ready: 1 for ready.

4 double_to_sig16b

Transform data types from double to 16 bit binary signal

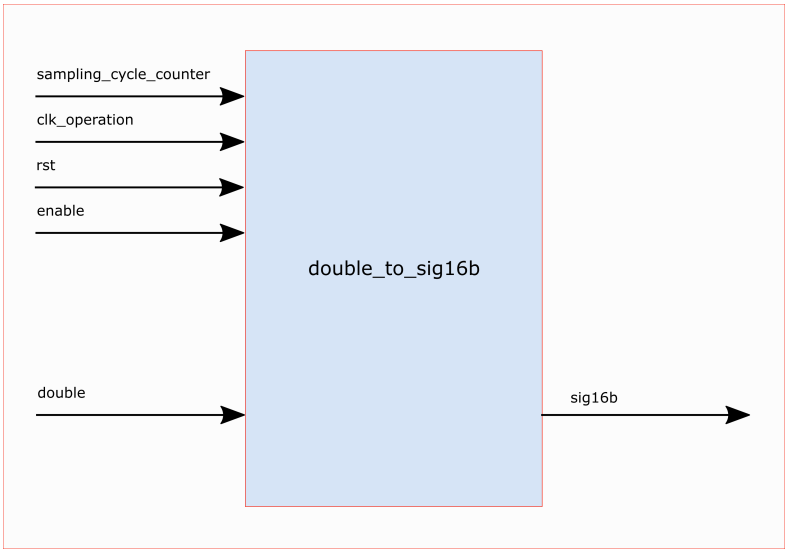

Figure 4: double_to_sig16b

4.1 Input

sampling_cycle_counter: Global sampling clocks.

clk_operation: Global operation clocks.

rst: Global reset.

enable: Local input. Needs to stay 1 when using the module.

double: Input in double.

4.2 Output

sig16b: Output in 16 bit binary formate.

5 para_approx

Estimate parameters for given original signals and signals with lags.

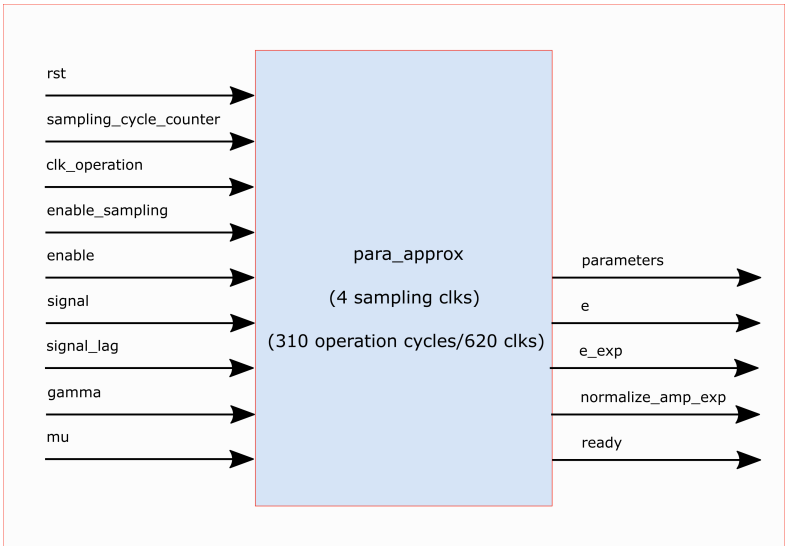

Figure 5: para_approx

5.1 Input

rst: Global reset.

sampling_cycle_counter: Global sampling clocks.

clk_operation: Global operation clocks.

enable_sampling: Local input. In order to make sure the samplings are aligned, needs to stay on even the module is not operating.

enable: Local input. Turn on for 2 operation cycles/4 operation clks and then turn off.

signal: Input signal in double.

signal_lag: Input lag signal in double

gamma: Default is 64'b0011111111010000000000000000000000000000000000000000000000000000(0.01)

mu: Default is
64'b0011111111110000000000000000000000000000000000000000000000000000(1)

5.2 Output

parameters: Estimate parameters in double.

e: Unbiased error of prediction

e_exp: Exponential of unbiased error of prediction

normalize_amp_exp: Exponential of amplitude of normalization. For debug purpose.

ready: 1 for ready

6 echo_cancelation

Preform echo cancellation for given parameters, original signals and signal with lags.

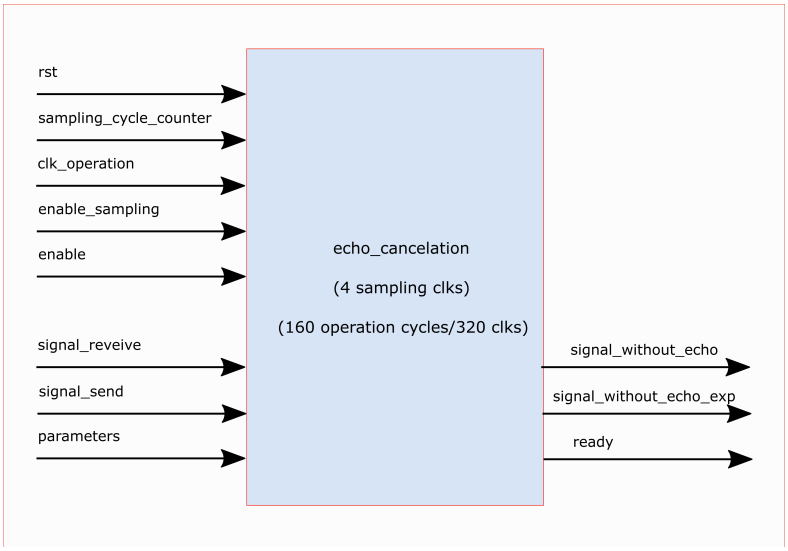

Figure 6: echo_cancelation

6.1 Input

rst: Global reset.

sampling_cycle_counter: Global sampling clocks.

clk_operation: Global operation clocks.

enable_sampling Local input. In order to make sure the samplings are aligned, needs to stay on even the module is not operating.

enable: Local input. Turn on for 2 operation cycles/4 operation clks and then turn off.

signal_reveive: Signals with echo in double.

signal_send: Original signals in double.

parameters: Estimate parameters

6.2 Output

signal_without_echo: Signal after echo cancellation in double

signal_without_echo_exp: Exponential of output signal. For debug purpose.

ready: 1 for ready.

7 lag_generator

Generate lag signal for given original signal and parameters.

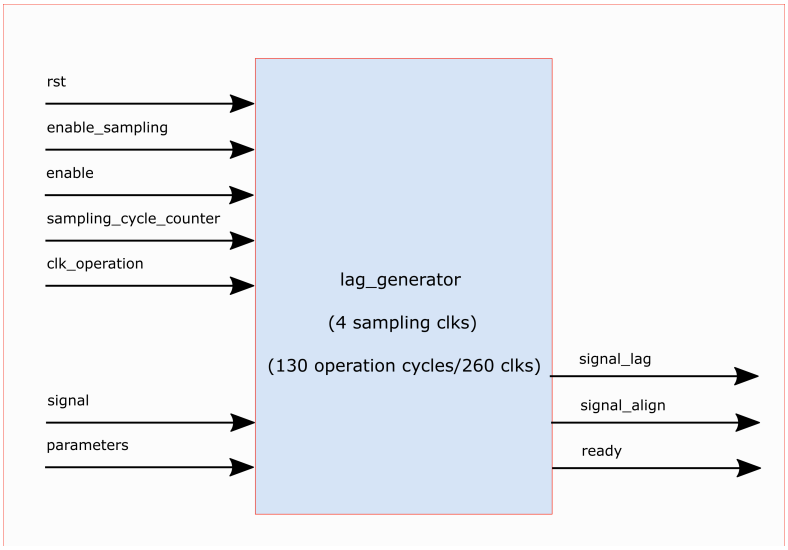

Figure 7: lag_generator

7.1 Input

rst: Global reset.

enable_sampling Local input. In order to make sure the samplings are aligned, needs to stay on even the module is not operating.

enable: Local input. Turn on for 2 operation cycles/4 operation clks and then turn off.

sampling_cycle_counter: Global sampling clocks.

clk_operation: Global operation clocks.

signal: Signals with echo in double.

parameters: Estimate parameters

7.2 Output

signal_lag: Signal after echo cancellation in double

signal_align: Exponential of output signal. For debug purpose.

ready: 1 for ready.

8 double_16b_tb

Test bench for data conversion modules:

double_to_sig16b.v

sig16b_to_double.v

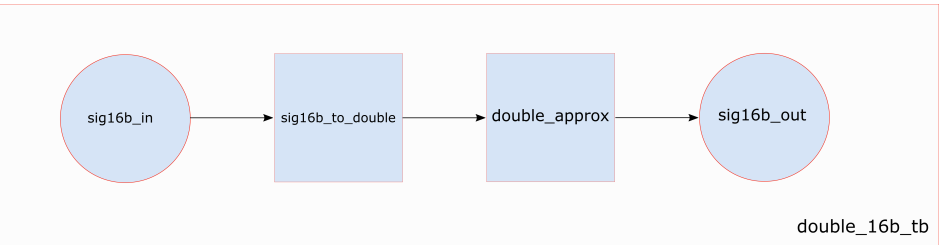

Figure 8: double_16b_tb hierarchy

9 tb_all

Test bench for all sub-level modules:

sig16b_to_double.v

lag_generator.v

para_approx.v

echo_cancelation.v

double_to_sig16b.v

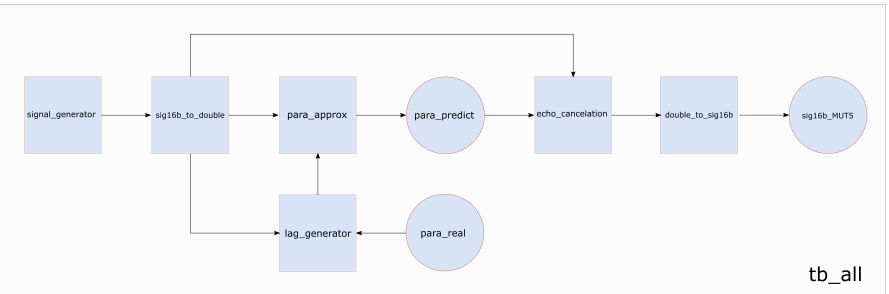

Figure 9: tb_all hierarchy

10 echo_cancelation_full_tb

Test bench for top-level module:

echo_cancelation_full.v

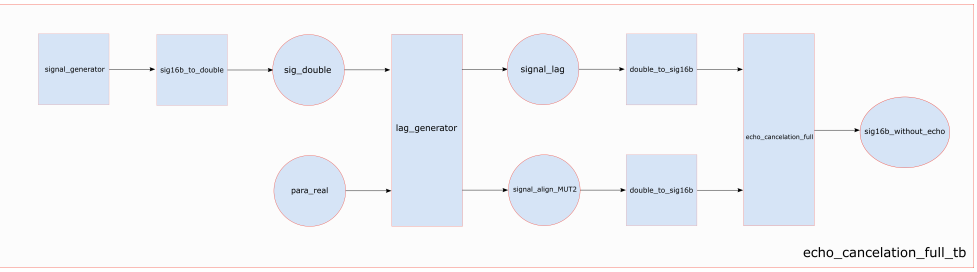

Figure 10: echo_cancelation_full_tb hierarchy