Peak Cancellation Crest Factor Reduction v6.2

LogiCORE IP Product Guide

Vivado Design Suite

PG097 June 6, 2018

Table of Contents

Introduction

Crest Factor Reduction (CFR) is used to limit the dynamic range of the signals being transmitted in wireless communications and other applications. Multi-user and multi-carrier signals often have a high peak-to-average ratio (PAR). This places high demands on the data converters and especially limits the efficiency of operation of the Power Amplifiers (PAs) used in cellular base stations. Reducing the PAR is therefore beneficial in increasing PA efficiency by allowing higher average power to be transmitted before saturation occurs.

Features

- Support for multiple air interface standards.
- Smart Peak Processing mode for supporting wide transmit bandwidth up to 400 MHz, processes incoming samples at >1.2 times instantaneous Bandwidth (iBW) reducing resource utilization.
- User selectable carrier configuration agnostic Window Crest Factor Reduction (WCFR) available as a standalone or a post processing stage.
- Support for power and frequency dynamics.
- Support for dynamic computation of Cancellation Pulse (CP).
- Support for optional hard clipper in post processing stage.
- Meets performance requirements (EVM, PAPR and ACLR) of all air interfaces.
- Configurable clock-to-sample ratio of 1, 2, 3, 4 and 8.
- Configurable number of Cancellation Pulse Generators (CPGs) of 1 to 12 per iteration.
- Support for 1, 2, 4, 8, and 16 antennas.
- Support for 1 to 8 iterations.

LogiCORE™ IP Facts Table	
Core Specifics	
Supported Device Family[1]	UltraScale+™ Families UltraScale™ Architecture Zynq® UltraScale+™ Devices 7 Series
Supported User Interfaces	AXI4-Stream, AXI4-Lite
Resources	PC-CFR web page (registration required)
Provided with Core	
Design Files	Encrypted RTL
Example Design	Not Provided
Test Bench	VHDL
Constraints File	Vivado: XDC
Simulation Model	VHDL and Verilog Structural Simulation Model MATLAB® Model available
Supported S/W Driver	N/A
Tested Design Flows[2]	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx at the Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

Crest Factor Reduction (CFR) is used to limit the dynamic range of the signals being transmitted in Wireless Communications and other applications. Multi-user and multi-carrier signals often have a high peak-to-average ratio (PAR). This places high demands on the data converters and especially limits the efficiency of operation of the Power Amplifiers (PAs) used in cellular base stations. Reducing the PAR is therefore beneficial in increasing PA efficiency by allowing higher average power to be transmitted before saturation occurs.

In a modern transmit chain, CFR is often incorporated with Digital Predistortion (DPD), which acts to linearize the PA, allowing operation at maximum efficiency with spectral compliance. CFR complements DPD because it levels the signal peaks, making accurate correction estimation easier.

The Xilinx® Peak Cancellation-CFR (PC-CFR) core is an efficient, flexible and easy-to-use implementation that supports Virtex®-UltraScale™, Kintex®-UltraScale, Virtex-7, Kintex-7, Artix®-7, and Zynq ® UltraScale+™ Devices (Zynq SoC, MPSoC and RFSoC). It is configurable both in function, supporting all major cellular wireless air interfaces, and in use, supporting many clocking and resource requirements. It can also handle dynamic power and frequency variations in the incoming data by computing the cancellation pulse coefficients dynamically.

Features and General Description

The PC-CFR core processes control and data through industry-standard AXI4 interfaces that allow immediate logic-free connection to other Xilinx IP components and to any general environment. The control interface is AXI4-Lite compliant and the data interface is AXI4-Stream compliant. The control interface provides access to a set of configuration registers and a pulse coefficients RAM and the data interface is used for streaming data in/out of the core. The data flow is unidirectional with no rate or bit-width change. A typical CFR application consists of multiple iterations and multiple antennas that can be configured through the Vivado® Integrated Design Environment (IDE). The core is configured for a particular application through the control interface. In particular, the contents of the pulse coefficients RAM are related to the spectrum of the signal being transmitted. Mixed-mode signal operation is also supported. Pulse coefficients can be pre-configured at generation time through a `.coe` file or configured in operation using the control interface. There is also provision for a shadow bank of coefficients to be loaded, and then activated with a select signal, to cater to applications where fast dynamic switching is required. Functions that can be run with MATLAB ® software are supplied for simulation and design of the cancellation pulse.

The core can be configured for clock-to-sample ratios of 1, 2, 3, 4 and 8 and for algorithmic complexity, allowing FPGA resources to be minimized for a given application.

Feature Summary

- Quantization support for 16 and 18 bits.
- Support for cancellation pulse read back in static mode and base pulse read back in dynamic mode.
- Support for read back of CFR configuration and statistics registers.
- Supports multiple air interface standards - MC-GSM, WCDMA, TD-SCDMA, WiMAX, LTE, CDMA2000, mixed mode (for example, GSM + LTE, TD-SCDMA + LTE, WCDMA + LTE); also supports frequency hopping GSM (FH-GSM) with a maximum of eight carriers.
- Can be clocked at 491.52 MHz or higher for -2 speed grade devices.
- Employs smart peak processing, which operates on incoming samples with sampling rate f_s > 1.2 times iBW for accurate peak detection supporting wide transmit bandwidth up to 400 MHz ($f_s = 491.52$ MHz). This feature helps in running the design close to iBW, enabling a higher clock-to-sample ratio leading to reduced resource utilization. In wide bandwidth cases, the design can be run with a lower clock-to-sample ratio processing the whole combined stream. The core has been tested for performance up to 200 MHz for single RAT and 400 MHz for multi-RAT multi-band cases.
- User-selectable Window Crest Factor Reduction (WCFR) is used as a post processing stage for better peak-to-average ratio (PAR) performance. WCFR post processing is enforced when smart peak processing is selected. WCFR can also be used as a standalone AXI4-Stream and AXI4-Lite protocol compliant CFR block for low cost (such as multi-carrier GSM) or for carrier configuration agnostic scenarios. You can optionally disable the smart peak processing when WCFR is selected as a post processing stage or in a stand-alone WCFR mode.
- Configurable clock-to-sample ratio of 1, 2, 3, 4, and 8 for resource optimization.
- Support for optional hard clipper in post processing stage. CFR stage can also have Hard Clipper as a standalone AXI4-Stream compliant block for low cost low footprint deployment scenarios (e.g. multi-carrier GSM).
- User-selectable cancellation pulse loading mode: single-pulse fixed coefficients, single-pulse configurable coefficients, two-pulses configurable coefficients.

- Cancellation pulse read back support (in configurable coefficients modes only) in static mode and base pulse read back support in dynamic mode enabling faster debug.
- AXI4-Stream compliant for data interface and AXI4-Lite compliant for control interface.
- User-selectable cancellation pulse generators (real or complex) for better resource utilization (DSP48s and block RAMs).
- Configurable number of CPGs 1-12 per iteration; more CPGs per iteration helps to reduce the number of iterations and overall latency of the core.
- Exploits conjugate symmetry of the Cancellation Pulse (CP) coefficients to optimize CP block RAM utilizations.
- Supports configurable maximum CP length of 511, 1023 and 2047, allowing CP length to go up to 2047 for low bandwidth or high sampling rate cases.
- Configurable Peak Detect Window length to avoid detection of multiple peaks in multi-carrier scenarios.
- Enhanced the core to build various architectures for Multi-Radio Access Technology (MRAT) capability with maximum bandwidth support up to 400 MHz. The core has been tested for performance up to 200 MHz in single-RAT and up to 400 MHz in multi-RAT multi-band configurations.
- Support for dynamic power and frequency variation.
- Addition of dynamic Cancellation Pulse (CP) computation mode where the CP is computed internally by the core based on power and frequency for individual carriers. This is called Dynamic mode and is required only if incoming data is frequency hopping and/or power hopping.
- Read back of CFR statistics registers such as, maximum number of CPGs used, missed peaks (in the last iteration of PC-CFR), hard clipped samples and number of peaks processed by WCFR is provided. All AXI4-Lite registers can be read back for testing and faster debug.
- Latency is part of the Implementation Details tab and is computed as per Equation 3-4 and Equation 3-5. Core latency is a function of CP length, CPGs per iteration, number of iterations used and the post-processing stage.
- In Dynamic mode, a maximum of five different RATs are supported. The total number of carriers using all RATs is thirty. LTE 5MHz and LTE 10MHz are considered different RATs because the corresponding base pulses are different.
- Option of slow and fast update in Dynamic mode. Slow update is suitable for power varying configurations where the rate of change is in the order of milliseconds, while fast update is required when the update rate is in hundreds of microseconds (like in FH-GSM where the update is on a per slot basis, that is, 577 µs).
- MATLAB software simulator available for system-level verification (and simulation).
- Optional support for forwarding Out-of-Band signals on TUSER bus of AXI4-Stream interface for data. The TUSER bus is delay matched with the data path.

Licensing and Ordering

License Checkers

If the IP requires a license key, the key must be verified. The Vivado design tools have several license check points for gating licensed IP through the flow. If the license check succeeds, the IP can continue generation. Otherwise, generation halts with error. License checkpoints are enforced by the following tools:

- Vivado design tools: Vivado Synthesis, Vivado Implementation, write_bitstream (Tcl command)

IMPORTANT: *IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.*

License Type

This Xilinx LogiCORE IP module is provided under the terms of the Xilinx Core License Agreement. The module is shipped as part of the Vivado Design Suite. For full access to all core functionalities in simulation and in hardware, you must purchase a license for the core. Contact your local Xilinx sales representative for information about pricing and availability.

For more information, visit the PC-CFR product web page.

Information about other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Evaluation

An evaluation license is available for this core. The evaluation version operates in the same way as the full version for several hours, dependent on clock frequency.

IMPORTANT: *At the end of the evaluation period, data output is forced to zeros while output valid remains unaffected. If you notice this behavior in hardware, it probably means that you are using an evaluation version of the core.*

The Xilinx tools warn that an evaluation license is being used during netlist implementation. If a full license is installed, delete the old XCI file, reconfigure and regenerate the core. More details on evaluation can be found in the PC-CFR evaluation lounge (registration required).

https://www.xilinx.com/member/pc_cfr_eval/index.htm

Product Specification

Standards

The PC-CFR v6.2 core adheres to the AMBA® AXI4-Stream protocol for the data interface and AMBA AXI4-Lite for the control interface.

Resource Utilization

Resource requirements and performance are dependent on the CFR Method (PC-CFR or WCFR), smart peak processing, data rate (clock cycles per sample), number of cancellation pulse generators per iteration, number of antennas, number of iterations, and max cancellation pulse length. They are also susceptible to the version of implementation tools used.

For all configurations, a default data width of 16 bits is used. All Static mode configurations are used with Coefficient Type selected as **Complex** and Coefficient Selection selected as **Two Pulses configurable coefficients** in the Vivado® IDE. **Max Peak Detect Window** is set to 120 when smart peak processing is enabled, and to 197 when smart peak processing is disabled. For more information, see Peak Detect Window Length.

The latest resource utilization and maximum clock frequency tables for various core configurations can be downloaded from PC-CFR evaluation lounge. The resource utilization and performance numbers shown in these tables should only be used as guidance, and should be verified before specific use. Maximum clock frequencies quoted do not take clock jitter into account and should be derated by an amount appropriate to the clock source jitter specification.

Port Descriptions

Figure 2-1 displays the signal names; Table 2-1 defines these signals.

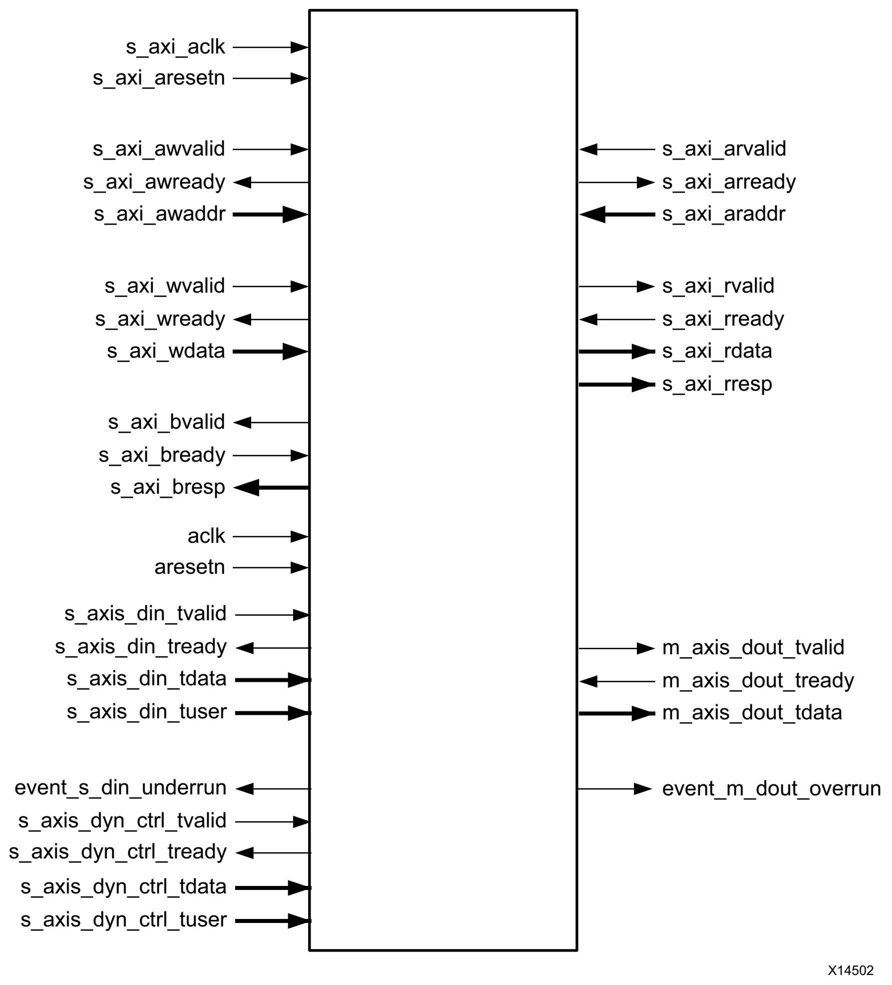


Figure 2-1: **Input and Output Ports**

Table 2-1: **Input and Output Ports**

Port	Direction	Width (bits)	Description
s_axi_aclk	Input	1	Clock for control interface. Synchronous operations on the control interface occur on the rising edge of this clock signal.
s_axi_aresetn	Input	1	Active-Low synchronous reset for the control interface. Resets the control interface logic to idle/power up state. Also resets all the configuration registers, but not the pulse coefficients RAM.

Table 2-1: **Input and Output Ports *(Cont'd)***

Port	Direction	Width (bits)	Description
s_axi_awvalid	Input	1	Active-High. When asserted, the core reads the address on the s_axi_awaddr port, if s_axi_awready is High.
s_axi_awaddr	Input	32	Address to write control data. Address read when s_axi_awvalid and s_axi_awready asserted High on a rising clock edge.
s_axi_awready	Output	1	Active-High. Indicates that the core is ready for a valid address on the s_axi_awaddr port.
s_axi_wvalid	Input	1	Active-High. When asserted, the core reads the data on the s_axi_wdata port, if s_axi_wready is High.
s_axi_wdata	Input	32	Control data to write to memory. Data read when s_axi_wvalid and s_axi_wready asserted High on a rising clock edge.
s_axi_wready	Output	1	Active-High. Indicates that the core is ready for valid data on the s_axi_wdata port.
s_axi_bvalid	Output	1	Active-High. Asserted when the core has completed the current write transaction and the response is available.
s_axi_bresp	Output	2	Write transaction response to the user (00=OK, 1X=ERROR, 01=NA). Data read when s_axi_bvalid and s_axi_bready asserted High on a rising clock edge.
s_axi_bready	Input	1	Active-High. Indicates that the user is ready to read data on the s_axi_bresp port.
s_axi_arvalid	Input	1	Active-High. When asserted, the core reads the address on the s_axi_araddr port, if s_axi_arready is High.
s_axi_araddr	Input	32	Address to read control data. Address read when s_axi_arvalid and s_axi_arready asserted High on a rising clock edge.
s_axi_arready	Output	1	Active-High. Indicates that the core is ready for a valid address on the s_axi_araddr port.
s_axi_rvalid	Output	1	Active-High. Asserted when the core outputs read data on the s_axi_rdata port.
s_axi_rdata	Output	32	Control data read from memory. Data read by user when s_axi_rvalid and s_axi_rready asserted High on a rising clock edge.
s_axi_rresp	Output	2	Read transaction response to the user (00=OK, 1X=ERROR, 01=NA). Data read when s_axi_rvalid and s_axi_rready asserted High on a rising clock edge.
s_axi_rready	Input	1	Active-High. Indicates that the user is ready to read data on the s_axi_rdata port.

Table 2-1: **Input and Output Ports *(Cont'd)***

Port	Direction	Width (bits)	Description
aclk	Input	1	Clock for data interface and core operations. Synchronous operations on the data interface occur on the rising edge of this clock signal.
aresetn	Input	1	Active-Low synchronous reset for the data interface and core operations. Disables any active cancellation pulse generators and resets the data interface to idle/power-up state, awaiting new data.
s_axis_din_tvalid	Input	1	Active-High. When asserted, the core reads the data on the s_axis_din_tdata port, if s_axis_din_tready is High.
s_axis_din_tready	Output	1	Active-High. Indicates that the core is ready for valid data on the s_axis_din_tdata port.
s_axis_din_tdata	Input	C_AXIS_ TDATA_WIDTH	Slave interface data input to the core. Port width specified at configuration based on no. of antennas (after byte-rounding for AXI4-Stream).
s_axis_din_tuser	Input	C_AXIS_TUSER _WIDTH	This port is present only when dynamic mode is enabled or TUSER forwarding is selected. In the case of dynamic mode, each bit used as a slot marker for the corresponding antenna. Number of bits equals the number of antennas present.If TUSER forwarding is selected, the TUSER width is 8-bits per antenna. Hence, TUSER width = 8*No.of Antennas.
s_axis_dyn_ctrl_tvalid	Input	1	Active-High. This port is present only when dynamic mode is enabled. When asserted, the core reads the data on the s_axis_dyn_ctrl_tdata port, if s_axis_dyn_ctrl_tready is High.
s_axis_dyn_ctrl_tready	Output	1	Active-High. This port is present only when dynamic mode is enabled. Indicates that the core is ready for valid data on the s_axis_dyn_ctrl_tdata port.
s_axis_dyn_ctrl_tdata	Input	C_AXIS_DYN_ CTRL_TDATA_ WIDTH	This port is present only when dynamic mode is enabled. When frequency hopping is disabled this port indicates relative power. However when frequency hopping is enabled, this port conveys relative power as well as frequency information. Bits [7:0] are used for power per antenna. Power is an 8-bit unsigned integer; the frequency is in 24-bit twos complement format. Bits [31:8] are used for frequency per antenna with the format given by round (f_c/fs*2^{24}) when smart peak processing is disabled and round ($f_c/(4f_s) * 2^{24}$) when smart peak processing is enabled. For non-frequency hopping case the total number bits is 8*Number of Antennas. For the case of frequency hopping the total number of bits is 32*Number of Antennas.

Table 2-1: **Input and Output Ports *(Cont'd)***

Port	Direction	Width (bits)	Description
s_axis_dyn_ctrl_tuser	Input	C_AXIS_DYN_ CTRL_TUSER_ WIDTH	This port is present only when dynamic mode is enabled. Each bit is used as a power marker for the corresponding antenna. Number of bits equals the number of antennas present.
event_s_din_underrun	Output	1	Error signal to indicate underrun on the slave data input interface.
m_axis_dout_tvalid	Output	1	Active-High. Asserted when the core writes data on the m_axis_dout_tdata port.
m_axis_dout_tready	Input	1	Active-High. When asserted, the downstream slave is ready for valid data on the m_axis_dout_tdata port.
m_axis_dout_tdata	Output	C_AXIS_ TDATA_WIDTH	Master interface data output from the core. Port width specified at configuration based on no. of antennas (after byte-rounding for AXI4-S).
event_m_dout_overrrun	Output	1	Error signal to indicate overrun on the master data output interface.
m_axis_dout_tuser	Output	C_AXIS_ TUSER_WIDTH	TUSER output carrying Out-of-band signals. This is enabled whenever TUSER forwarding is selected.

Register Space

Control Interface

The control interface is used for writing to the configuration registers and loading the pulse coefficients RAM. It is implemented as an AXI4-Lite slave interface. The total memory space (including register and pulse coefficients space) is 512K words, although only the required memory locations are physically implemented in hardware. Even though the required address bus width is only 22 bits, the core has a 32-bit address bus at the interface. Internally, the higher 12-bits (31-22) are ignored. The AXI4-Lite register space is byte addressable and the corresponding word addresses are given in Table 2-3 (the corresponding AXI4-Lite address is four times the word address). The address format and the memory map are given in Table 2-2 and Table 2-3 respectively.

The memory space is divided into 16 partitions (one for each antenna) of 32K words each and bits 18-15 of the address (Ant Sel) are used to select the antenna. Bits 21-19 are only used in Dynamic mode to indicate which base pulse to be written into. Bits 21-19 are used when RAM Sel (Bit 14) is set to 1. Each 32K partition consists of 4K register space and 16K of pulse coefficients space. Bit 14 of the address (Reg/RAM Sel) is used to select between register space and pulse coefficients space (0: register space, 1: pulse coefficients RAM). The Maximum Cancellation Pulse Length parameter in the Vivado IDE is used to optimize block RAM resource usage based on the requirement. Bits 13-2, together with bits 21-19, are used to address a specific configuration register or a specific location in the pulse coefficients RAM.

Table 2-2: **Control Interface Address Format**

Bit 31-22	Bits 21-19	Bits 18-15	Bit 14	Bit 13-2	Bits 1-0
Unused	Static Coefficients/Base Pulse Select	Ant Sel	Reg/RAM Sel	Offset	00

Table 2-3: **Control Memory Map**

Coeff/ Base Pulse Select	Reg/ RAM Sel	Offset (hex)	Register/ RAM Name	Data width (bits)	Description
b'000	0	0x000	Filter Length	9, 10 or 11[(1)]	Input to provide the number of filter taps. This controls the number of coefficient values that are read from memory. Valid only when configurable coefficients have been configured, as otherwise this value is inferred from the .coe file. This is the full length of the CP pulse with conjugate symmetry, that is, when the filter length is 511, there are 256 samples in the first half and the remaining 255 samples are complex conjugate of the first half. In Dynamic mode, this parameter indicates the length of the base pulses. When smart peak processing is enabled, the filter length must be of the form 8n+1, where n is any integer greater than 0.
b'000	0	0x001	Filter Select	1	Input to select which pulse is in use (0 = Pulse 0, 1 = Pulse 1). Valid only when 'Two pulses (selectable) configurable coefficients' is selected.
b'000	0	0x002	Threshold	16	Input to provide the threshold level over which valid peaks can be identified. The computation of threshold is explained in .
b'000	0	0x003–0x00A	Peak Detect Window Length for iterations 1 to 8	10	This is a window (in samples) over which the dominant peak is detected and cancellation applied. Setting a non-zero value (typically 3 or more) helps avoid CPG allocation to nearby multiple peaks in multi-carrier and/or multi-RAT scenarios. Minimum value is 3 and maximum value is Max Peak Detect Window parameter setting.
b'000	0	0x040–0x05D	Frequency[(2)]	32	Used in power Dynamic mode only, for max 30 carriers. The format for frequency is (round (f_c/f_s*2^{24})). See Dynamic CP Computation in Chapter 3 for more information.
b'000	0	0x072	Power Marker Window Start[(2)]	19	Used only when frequency hopping is enabled in Dynamic mode, an additional control for power ramp, so that power values are ignored before the envelope settles. This is measured in clock cycles (not samples).

Table 2-3: **Control Memory Map *(Cont'd)***

Coeff/ Base Pulse Select	Reg/ RAM Sel	Offset (hex)	Register/ RAM Name	Data width (bits)	Description
b'000	0	0x073	Next Slot Compute Start Location(2)	19	Used only when frequency hopping is enabled in Dynamic mode, an additional control for power ramp down, so that CP can be computed now for the next slot. This is measured in clock cycles (not samples).
b'000	0	0x074	Power change threshold(2)	8	Threshold for CP compute trigger in power Dynamic mode. This is an 8-bit value. See Dynamic Power Variation for more information.
b'000	0	0x075	Power Marker Window Stop(2)	19	Used only when frequency hopping is enabled in Dynamic mode, an additional control for power ramp, so that power values are ignored beyond this window. This is measured in clock cycles (not samples).
b'000	0	0x077	Hard clipper threshold	16	Threshold used for hard clipper. The computation of hard clipper threshold is similar to that of the PC-CFR threshold which is explained in .
b'000	0	0x071	Force Static in Dynamic(2)	1	Uses initialized CP instead of the dynamically computed CP.
b'000	0	0x070	CP initialization Enable(2)	1	Loads a new initialized CP into CPG memory.
b'000	0	0x078	WCFR threshold	16	Threshold used for WCFR. The computation of threshold is explained in .
b'000	0	0x0A0	Max CPGs	4	Maximum CPGs used in Iteration 0(3)
b'000	0	0x0A1	Max CPGs	4	Maximum CPGs used in Iteration 1(3)
b'000	0	0x0A2	Max CPGs	4	Maximum CPGs used in Iteration 2(3)
b'000	0	0x0A3	Max CPGs	4	Maximum CPGs used in Iteration 3(3)
b'000	0	0x0A4	Max CPGs	4	Maximum CPGs used in Iteration 4(3)
b'000	0	0x0A5	Max CPGs	4	Maximum CPGs used in Iteration 5(3)
b'000	0	0x0A6	Max CPGs	4	Maximum CPGs used in Iteration 6(3)
b'000	0	0x0A7	Max CPGs	4	Maximum CPGs used in Iteration 7(3)
b'000	0	0x080	Peaks Missed	32	Peaks Missed in the last iteration(3)
b'000	0	0x0C0	Hard Clipped Samples	32	This indicates Hard Clipped samples when hard clipper is present or peaks found in WCFR when WCFR is present as the last stage(3)
b'000	0	0x0E0	Stats Clear	1	Clear Statistics registers
b'000	0	0x07B	Core Configuration Read-back Register 4	8	See Table 2-7(3)

Table 2-3: **Control Memory Map *(Cont'd)***

Coeff/ Base Pulse Select	Reg/ RAM Sel	Offset (hex)	Register/ RAM Name	Data width (bits)	Description
b'000	0	0x07C	Core Configuration Read-back Register 1	32	See Table 2-4[3]
b'000	0	0x07D	Core Configuration Read-back Register 2	32	See Table 2-5[3]
b'000	0	0x07E	Core Configuration Read-back Register 3	32	See Table 2-6[3]
b'000	0	0x07F	Version	12	Indicates the PC-CFR core version. Twelve bits are used to indicate the complete version. Bits 3:0 convey minor version and bits 11:4 convey major version. This currently reads as 0x061[3].
b'000	1	0x000–(Max Cancellation Pulse Length–1)/2	Pulse 0 Coefficients RAM	32	Memory to store Pulse 0 coefficients. Coefficients are stored as 16-bit I concatenated with 16-bit Q (Q in MSW, I in LSW). Only the first half of the taps (first half of the filter length) needs be programmed.
b'000	1	(Max Cancellation Pulse Length+1)/2–Max Cancellation Pulse Length	Pulse 1 Coefficients RAM	32	Memory to store Pulse 1 coefficients. Coefficients are stored as 16-bit I concatenated with 16-bit Q (Q in MSW, I in LSW). Only the first half of the taps (first half of the filter length) needs be programmed.
b'001	1	0x000–(Max Cancellation Pulse Length–1)/2	Base pulse RAT A[2]	32	Base pulse for RAT A, to be written in this field. Coefficients are stored as 16-bit I concatenated with 16-bit Q (Q in MSW, I in LSW). The 16-bit Q values are always zero. Only the first half of the taps (first half of the filter length) needs to be programmed.
b'010	1	0x0000–(Max Cancellation Pulse Length–1)/2	Base pulse RAT B[2]	32	Base pulse for RAT B, to be written in this field. Coefficients are stored as 16-bit I concatenated with 16-bit Q (Q in MSW, I in LSW). The 16-bit Q values are always zero. Only the first half of the taps (first half of the filter length) needs to be programmed.
b'011	1	0x0000–(Max Cancellation Pulse Length–1)/2	Base pulse RAT C[2]	32	Base pulse for RAT C, to be written in this field. Coefficients are stored as 16-bit I concatenated with 16-bit Q (Q in MSW, I in LSW). The 16-bit Q values are always zero. Only the first half of the taps (first half of the filter length) needs to be programmed.

Table 2-3: **Control Memory Map *(Cont'd)***

Coeff/ Base Pulse Select	Reg/ RAM Sel	Offset (hex)	Register/ RAM Name	Data width (bits)	Description
b'100	1	0x0000–(Max Cancellation Pulse Length–1)/2	Base pulse RAT D	32	Base pulse for RAT D, to be written in this field. Coefficients are stored as 16-bit I concatenated with 16-bit Q (Q in MSW, I in LSW). The 16-bit Q values are always zero. Only the first half of the taps (first half of the filter length) needs to be programmed.
b'101	1	0x0000–(Max Cancellation Pulse Length–1)/2	Base pulse RAT E	32	Base pulse for RAT E, to be written in this field. Coefficients are stored as 16-bit I concatenated with 16-bit Q (Q in MSW, I in LSW). The 16-bit Q values are always zero. Only the first half of the taps (first half of the filter length) needs to be programmed.

Notes:

1. Depending on Max Cancellation Pulse Length = 511, 1023 or 2047.
2. Used only in Dynamic mode and can be ignored in Static mode.
3. Read-only register.

Configuration Registers

The configuration registers are defined in Table 2-3. These registers are typically programmed once at start-up and can be changed dynamically if required. No attempt has been made to balance the pipelines of these registers to the different parts of the circuit where they are applied. So if any of these registers are changed during normal core operation without applying a core reset (`aresetn`), then transient behavior will occur at the output for a number of samples up to 2300 times the number of iterations. The configuration registers can be read through the AXI4-Lite interface.

Core Configuration Read-back Registers

Four read-only registers have been provided to show the core parameters that were set at the time of configuring the IP from Vivado IDE. The details of these read-back registers are provided below:

Core Configuration Read-back Register 1

This is a 32-bit register and is described in Table 2-4.

Table 2-4: **Core Configuration Read-back Register 1**[(1)]

Type	Width	Bit Field	Description
CFR Method	2	[1:0]	00: Smart peak processing disable 01: Smart peak Processing enable 10: WCFR stand-alone 11: Stand-alone hard clipper
Number of Antennas	3	[4:2]	000: 1, 001: 2, 010: 4, 011: 8, 100:16 Remaining: Unused
Data Rate	4	[8:5]	000: 1, 001: 2, 010: 3, 011: 4, 101: 8, Remaining: Unused
Number of Iterations	3	[11:9]	000: 1, ...111: 8
Data Width	3	[14:12]	010: 16, 100: 18, Remaining: Unused
Max CP Length	3	[17:15]	010: 511, 011: 1023, 100: 2047, Remaining: Unused
Configurable Coefficients	1	[18]	0: Non-configurable, 1: Configurable
Pulse Mode	1	[19]	0: Single pulse mode using one filter bank. 1: Two pulse mode using two filter banks
Core Mode	1	[20]	0: Static, 1: Dynamic
CP Coefficient Type	1	[21]	0: Real, 1: Complex
WCFR Window Length	3	[24:22]	001: 192, 010: 384, 011: 768, Remaining: Unused
Hard Clipper	1	[25]	0: Disabled, 1: Enabled
Frequency Hopping	1	[26]	0: Disabled, 1: Enabled
CP Engine	1	[27]	0: Serial, 1: Parallel
Reserved	3	[31:28]	Unused

Notes:

1. This register is read-only.

Core Configuration Read-back Register 2

This register indicates the number of CPGs present in each iteration. A maximum of eight iterations can be present in the core. Four bits are used to indicate CPGs per iteration as shown in Table 2-5.

Table 2-5: **Core Configuration Read-back Register 2[(1)]**

Type	Width	Bit Field	Description
Number of CPGs per iteration	4	[3:0]	Number in iteration 1
	4	[7:4]	Number in iteration 2
	4	[11:8]	Number in iteration 3
	4	[15:12]	Number in iteration 4
	4	[19:16]	Number in iteration 5
	4	[23:20]	Number in iteration 6
	4	[27:24]	Number in iteration 7
	4	[31:28]	Number in iteration 8

Notes:

1. This register is read-only.

Core Configuration Read-back Register 3

This register encodes the MIF SIZE, number of carriers per RAT A, B, C and the maximum peak delay as shown in Table 2-6.

Table 2-6: **Core Configuration Read-back Register 3[(1)]**

Type	Width	Bit Field	Description
MIF size	11	[10:0]	Half the number of CP (Cancellation Pulse) Coefficients in MIF (Memory Initialization File) file. That is (Total No.of Coefficients in MIF - 1)/2
Number of carriers for RAT A	4	[14:11]	0000: 0, ...1000:8
Number of carriers for RAT B	4	[18:15]	0000: 0, ...1000:8
Number of carriers for RAT C	4	[22:19]	0000: 0, ...1000:8
Max peak delay	9	[31:23]	This represents Max Peak Delay and it depends upon the Peak Detect window setting

Notes:

1. This register is read-only.

Core Configuration Read-back Register 4

This register captures the selection of number of RATD and RATE carriers as shown in Table 2-7.

Table 2-7: **Core Configuration Read-back Register 4**[(1)]

Type	Width	Bit Field	Description
Number of carriers for RAT D	4	[3:0]	0000:0, ... 1000:8
Number of carriers for RAT E	4	[7:4]	0000:0, ... 1000:8
Reserved	24	[31:8]	

Notes:

1. This register is read-only.

IMPORTANT: *Reading back composite cancellation pulse in dynamic mode is not supported.*

Pulse Coefficients RAM

The control interface is used to load the pulse coefficients RAM only when configurable coefficients are in use. The maximum pulse coefficients address space available per antenna is 4K for the two-pulse setup. The actual block RAM usage is based on the Maximum Cancellation Pulse Length parameter in the Vivado IDE. The pulse coefficients RAM can be read back through the control interface.

RAT Base Pulse RAM

The control interface is used to load the base pulse, the length of which is equal to half of the filter length programmed. The number of base pulses to be written is equal to the number of RATs selected. This is used for computing the CP coefficients dynamically. The PC-CFR core supports a maximum of five RATs. The actual block RAM usage is based on the Maximum Cancellation Pulse length parameter set in the Vivado IDE. The lengths of the base pulses have to be same. If the lengths are unequal, then the shorter pulses have to be zero padded symmetrically so that the peaks of all the base pulses are time aligned. The base pulse coefficients RAM can be read back. Table 3-1 can be used as reference for generation of base pulses.

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the core.

General Design Guidelines

The PC-CFR v6.2 core requires the input sampling rate to be at least 1.2 x iBW when smart peak processing is enabled. Disabling this feature makes it behave like PC-CFR v5.0, which requires 3 to 5x interpolated samples for accurate peak location estimation and cancellation. Thus, with smart peak processing enabled, a transmit signal of 100 MHz iBW requires a minimum sampling rate of $f_S = 122.88$ MHz and when disabled $f_S = 368.64$ MHz.

Clocking

There are two clock domains viz., AXI4-Stream and AXI4-Lite. AXI4-Stream uses input clock `aclk`. Separate clocks are used for AXI4-Lite and AXI4-Stream interfaces. It is expected that the AXI4-Lite uses a much slower clock compared to the AXI4-Stream interface

The PC-CFR core is designed for a fully synchronous operation with `aclk`. The core datapath is synchronous to this clock. This clock is an integral multiple (clock per sample) of the input sampling rate f_S.

Resets

A reset port, `aresetn`, is provided to return the data interface to its power-up state where it is not synchronized and waiting for the first valid input data. `aresetn` does not reset all logic in the core, but deallocates any currently active pulse generators and prevents them being reallocated to anything but new data input. The initial output data after synchronization following a reset depends on the values in the processing pipeline and is not necessarily zero data. To completely flush the core and ensure zero valued output data prior to the processed input data, `aresetn` should be held asserted for a period greater than 2048 cycles. Alternatively, `aresetn` can be held asserted for a shorter period, but the first data sample input to the core must be delayed by a minimum of the difference between the length of the reset pulse and 2048 cycles. The reset latency is two cycles, that is, the core returns to idle state two cycles after `aresetn` is deasserted.

Algorithm Description

Most practical CFR solutions are based in principle on subtracting a correction signal from the original signal as shown in Figure 3-1.

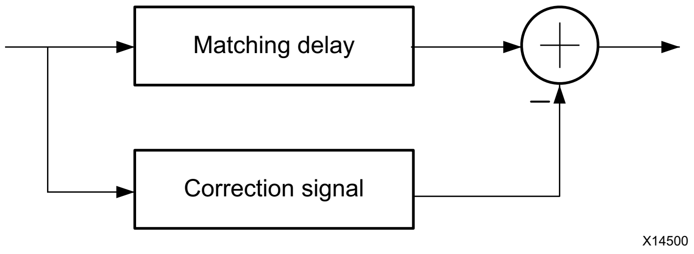


Figure 3-1: **Generalized CFR Block Diagram**

The correction signal is a spectrally compliant signal that matches the signal peaks. For example, in Noise-shaping CFR (see *High Density WiMAX Digital Front End Reference Design* [Ref 1]) the correction signal is a filtered version of the signal after magnitude thresholding, that is, the signal above a clipping threshold is filtered by a noise-shaping filter. In Peak Cancellation CFR, the correction signal is a sum of individual cancellation pulses. The pulses are applied by searching for peaks in the signal (see *Peak Cancellation Crest Factor Reduction in a Multi-Standard Transmit System Application Note* (XAPP1174) [Ref 2] for details of peak detection). Each pulse is the impulse response of a filter that is designed to match the spectral content of the signal. Figures 3-2, 3-3, and 3-4 depict the sequence of events. Figure 3-2 shows the detection of the peak and the identification of the excess amplitude over threshold, (A–Ath). Figure 3-3 shows the cancellation pulse. This pulse is the appropriate unit magnitude impulse response scaled by CP_val = (A–Ath) * $e^{j\vartheta}$ where ϑ is the phase and j is $\sqrt{-1}$.

The cancellation pulse is scaled to the excess magnitude over the desired clipping threshold at the peak with the phase of the signal at the peak. This is because the signals are complex and each cancellation pulse must be rotated to match the phase of the corresponding signal peak. When the cancellation pulse is subtracted from the original signal, it reduces the signal peak magnitude to the threshold value, as indicated in Figure 3-4, while preserving the signal phase. For asymmetric spectra, the cancellation pulse itself is complex but with conjugate symmetry.

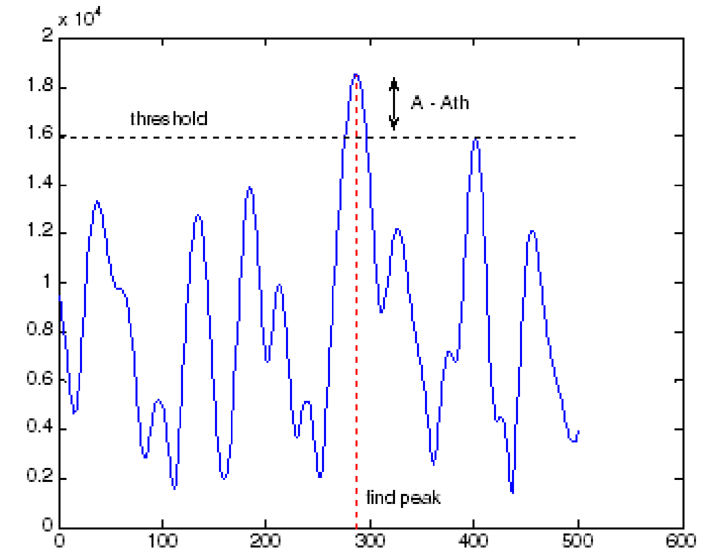


Figure 3-2: **Peak Detection**

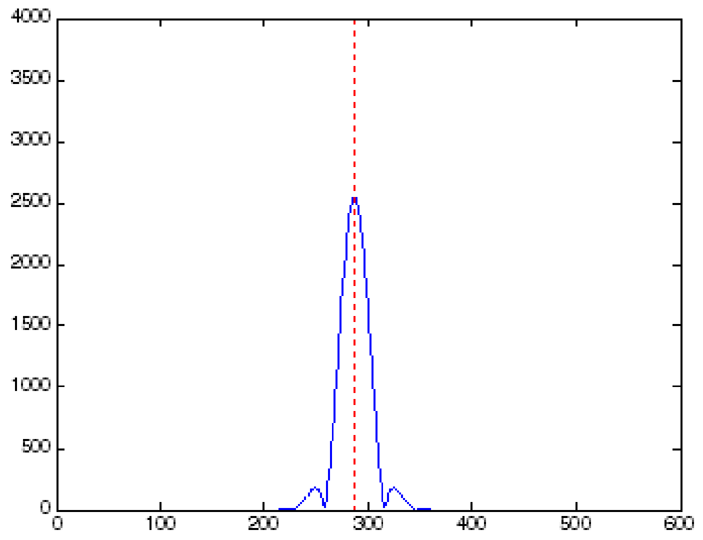


Figure 3-3: **Cancellation Pulse**

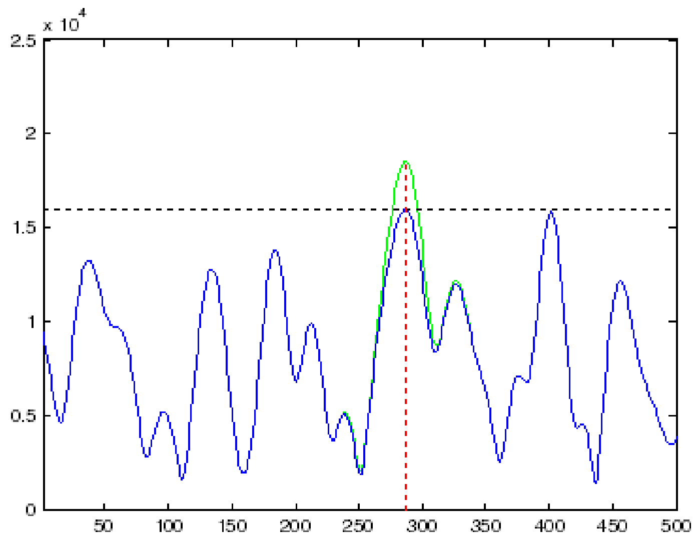


Figure 3-4: **Peak Cancellation**

In hardware, the cancellation pulse is generated by a Cancellation Pulse Generator (CPG) with function as indicated in Figure 3-5. The address counter must be triggered at the appropriate time to play out the pulse and the complex multiplier factors in the correct scaling.

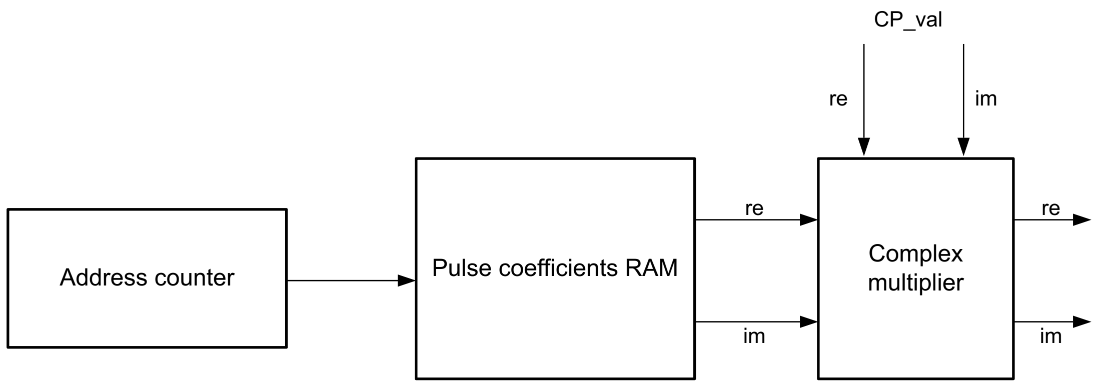


Figure 3-5: **Cancellation Pulse Generator**

The resource complexity of the PC-CFR core is bounded by having a finite number of cancellation pulses available at any one time. This means that, depending on the signals, not all the peaks are guaranteed to be canceled in one pass, so an application typically has multiple iterations.

A detailed structure for the PC-CFR core implementation is shown in Figure 3-6.

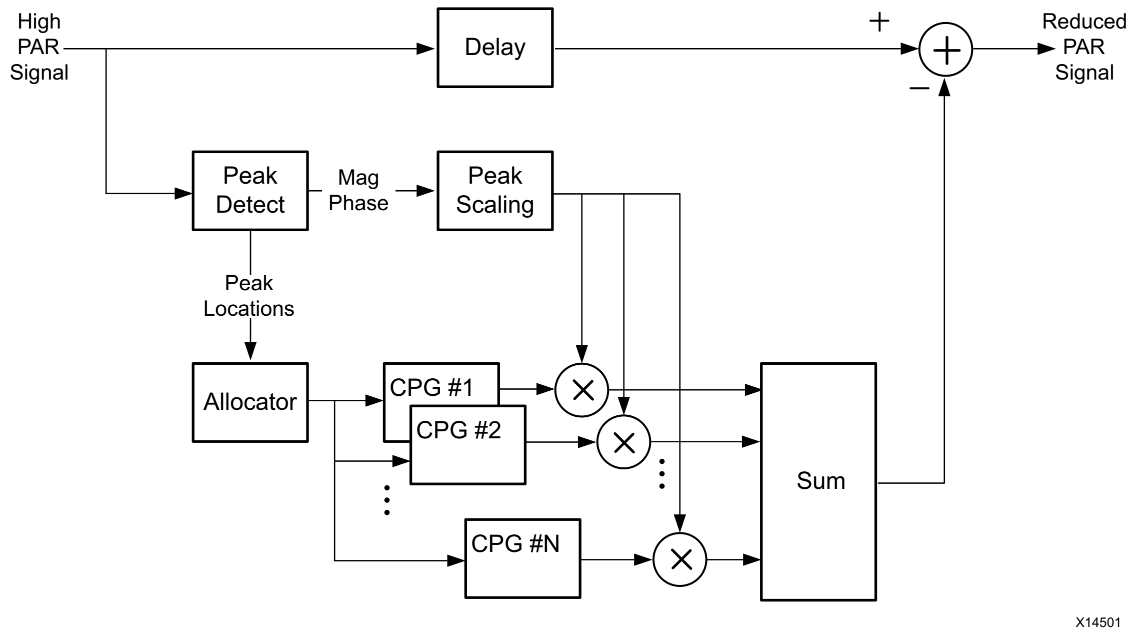


Figure 3-6: **PC-CFR High-level Block View**

The Peak Detect block identifies the signal sample at which the magnitude is a maximum in a region where it exceeds the CFR Threshold, which is a register input. The Peak Scale block forms the complex value for the cancellation pulse as described previously. The peak detect window parameter defines the window over which a peak is detected and careful setting of the value of this parameter can improve the EVM performance as discussed in the Number of CPGs at Each Iteration. The CPG pulse coefficients RAM is as discussed in Features and General Description.

One of the key features of the peak cancellation method is that it is very flexible with respect to supporting different air interfaces. Because the cancellation pulse coefficients are generated offline and then loaded into RAM, it is possible to support a wide variety of carrier configurations and bandwidths using the same hardware.

Smart Peak Processing Based PC-CFR

The PC-CFR algorithm works well if the incoming signal is oversampled by a factor of 3 to 5, as compared to the base sampling rate of the signal. Typically, an oversampling factor of 4 (also called as 4x interpolation factor) is required for good cancellation performance leading to better PAPR numbers. CPG resources can be reduced by folding the CPGs. This is achieved by clocking the core at a frequency which is an integral multiple of the sampling frequency of incoming data. This integral multiple is called Data Rate or Clocks Per Sample (CPS). However, the CPS cannot be increased beyond a certain limit because achieving timing closure with a higher core clock frequency is difficult.

FPGA resource usage tends to increase with higher values of interpolation factor. In the Smart Peak Processing mode, this oversampling requirement is relaxed to only 1.2 times the input bandwidth at the cost of some extra processing per iteration. Thus, input sampling rate can be kept low (no oversampling) leading to higher CPS. Higher values of CPS can be used resulting in higher resource savings.

IMPORTANT: *A cancellation pulse used with smart peak processing should be generated at 4 times the sampling frequency of incoming data. In addition, this pulse length should be of the form 8n+1, where n is any integer greater than 1.*

Window CFR

In the WCFR algorithm, peaks over threshold are identified. From these peaks the gain correction is computed and then filtered to control the splatter. The gain is then applied to the signal to reduce the level to the desired target threshold.

Figure 3-7 shows the top-level diagram of the WCFR approach. The peaks are identified, measured, and a gain correction is computed in the detector block. The gains are filtered and then used to vary the gain of the signal path so that the modified signal is always less than a target threshold.

Window CFR operates by attenuating the signal at peaks over the threshold and is agnostic to the carrier configuration of the input signal. The length of the filter sets the amount of spreading of the signal in the frequency domain. If the filter is made very long, then the splatter is reduced but the added EVM becomes larger. Thus, there is a trade-off between the ACLR and EVM in the selection of the window length. Three different window filter lengths (192, 384, and 768) are supported in the current implementation. A key advantage of the WCFR is it guarantees that no peaks are passed on unlike pulse cancellation methods. However, the WCFR has generally lower performance than PC-CFR. This degradation is proportional to the number of peaks that must be canceled. Therefore, this is an ideal processing stage to place at the end of the CFR chain where the number of peaks is small so that little degradation occurs but guarantees that no peaks escape cancellation. The current implementation supports WCFR as a post-processing stage which further helps in controlling peak EVM and reducing PC-CFR iterations. WCFR can also be instantiated in stand-alone mode for carrier configuration agnostic scenarios (repeater applications, FH MC-GSM, etc). WCFR is mandatory when smart peak processing is enabled.

When WCFR is used in standalone mode, then the smart peak processing can be enabled or disabled. When disabled, a sampling rate of at least 4x the iBW is recommended. When WCFR is used as post-processing stage in PC-CFR mode, then the smart peak processing is either enabled or disabled in accordance to the PC-CFR settings.

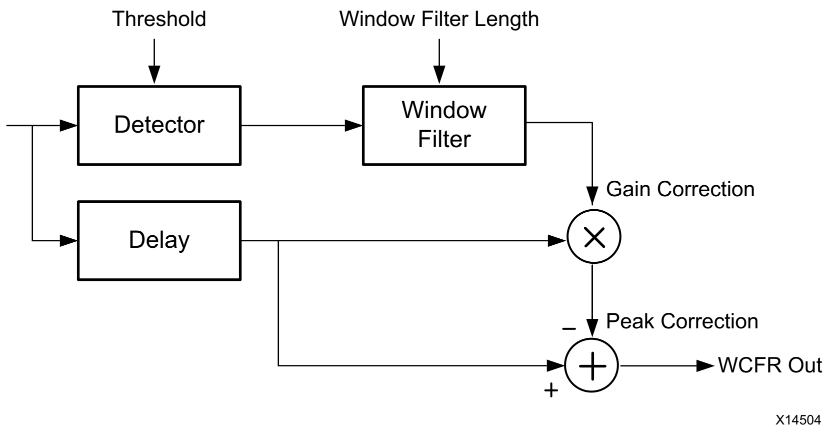


Figure 3-7: **WCFR High-level Block View**

Post-Processing Stage

A post-processing stage is provided after the primary CFR stage. This stage can consist of a hard clipper, a WCFR stage, or neither of these. In many cases, the use of this module can help cut down the number of iterations needed to get the desired EVM and PAR performance. (It is recommended to try this in MATLAB® software simulations first). This module can help reduce the resource utilization and latency for the design. The overall latency when post-processing stage is used is given in Equation 3-4 and Equation 3-5 on page 42. Hard clipper and WCFR use independent threshold values that can be programmed through the AXI4-Lite interface. The format of the threshold value for WCFR and hard clipper is identical to that of the PC-CFR threshold.

IMPORTANT: *WCFR is a necessary post-processing stage when smart peak processing is enabled. It helps to improve peak PAR by cleaning up any leftover peak after cancellation at 1.2 times iBW.*

Register Timing

Control Interface

For write access, both address and data can be written simultaneously, or one can be written before the other. The access is complete only after both address and data have been written and the write response has been received. The write response is a 2-bit signal indicating the success/failure of the current write access (00=OK, 1X=ERROR, 01=NA). It has a fixed latency of two cycles from when the latest of address/data has been accepted on the input. Until the write response is issued, the address and data ready signals are kept deasserted indicating that a new access cannot begin. Figure 3-8 shows the timing diagram for the write operation. The timing diagram firstly shows the case where address and data are written simultaneously (this is the fastest way to write data into the control memory, with one write transaction every three cycles). Secondly it shows an address write before the corresponding data, thirdly a data write before the corresponding address, and lastly a write terminated by reset. There can be any delay between the address and data or the data and address where the write operations occur on different cycles. The reset operation shows a write being terminated after the address has already been written; write operations where the data has been written, but not the address, can also be terminated using reset. The reset latency is one cycle, that is, the core output signals (such as write address ready, write data ready, etc) return to their idle state one cycle after reset is deasserted. In all of the previous cases, the reset operation also resets all the configuration registers (Filter Length, Filter Select, Threshold and Peak Detect Window, etc.), but does not reset the pulse coefficients RAM. Therefore the configuration registers need to be reloaded after a control reset operation.

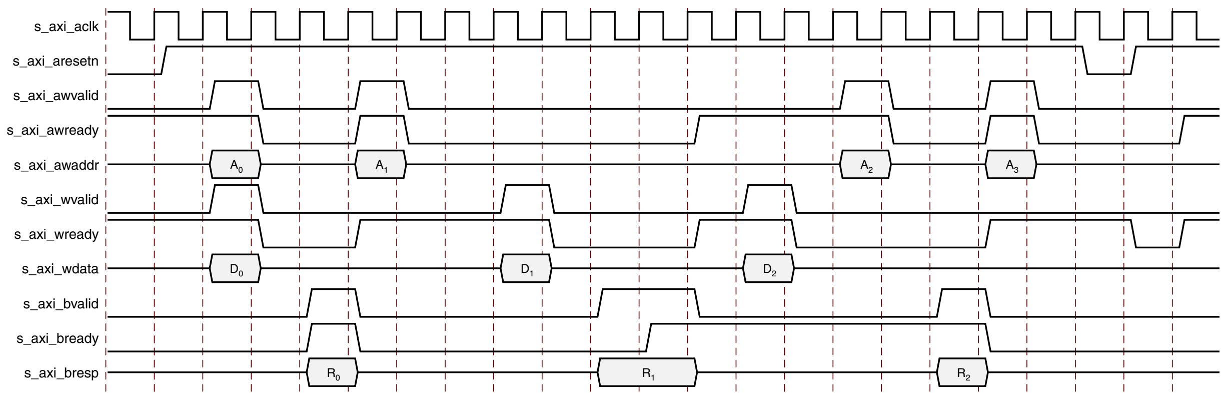


Figure 3-8: **Control Write Timing Diagram**

For read accesses, the address is written first and the access is complete after the read data/response has been issued. The read response is a 2-bit data indicating the success/failure of the current read access (00=OK, 1X=ERROR, 01=NA). It has a fixed latency of five cycles from when the address has been accepted on the input. Until the read data/response is issued, the address ready signal is kept deasserted indicating that a new access cannot begin. Figure 3-9 shows the timing diagram for the read operation. The reset operation shows a read being terminated after the address has already been written. The reset latency is one cycle, that is, the core output signals (such as read address ready, read data valid) return to their normal state one cycle after reset is deasserted.

The core does not support burst accesses or simultaneous write and read accesses. An ongoing write/read access has to be completed/terminated before the next write/read access can begin.

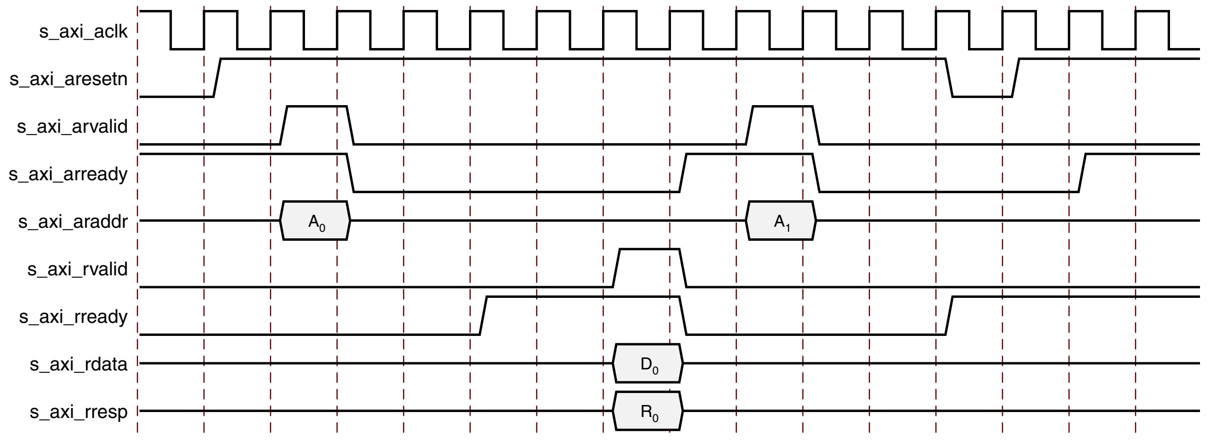


Figure 3-9: **Control Read Timing Diagram**

Data Interface

The core supports multiple data rates (1, 2, 3, 4, and 8 clock-cycles/sample) on the AXI4-Stream interface through READY-VALID handshaking. After power-up or reset, the core is idle until `s_axis_din_tvalid` has been asserted for the first time. `s_axis_din_tready` is asserted two cycles after reset deassertion and held High and `m_axis_dout_tvalid` is held Low until `s_axis_din_tvalid` is asserted. After this, the core is synchronized and expects data input and provides data output every x clock cycles defined by the configuration parameter "data rate" in clock-cycles/sample. `s_axis_din_tvalid` could be asserted at the same time as `s_axis_din_tready`, asserted in advance, or held asserted between `s_axis_din_tready` assertions.

IMPORTANT: *For TDD signals,* `s_axis_din_tdata` *is driven with zeros during the interval when the TX is silent and* `tvalid` *is asserted as it is normally done during the presence of data. If this is not done then an underrun condition occurs*

If `s_axis_din_tvalid` is not asserted when `s_axis_din_tready` is held High, then the underrun condition occurs because valid data was not available when the core expected it. As the core is free-running, underrun causes the preceding data sample to be input for a second time internally. Underrun is signaled by asserting the `event_s_din_underrun` flag. This flag remains set until the next core reset operation (by asserting `aresetn`). Figure 3-10 shows an example timing diagram where data rate = 3 clock-cycles/sample.

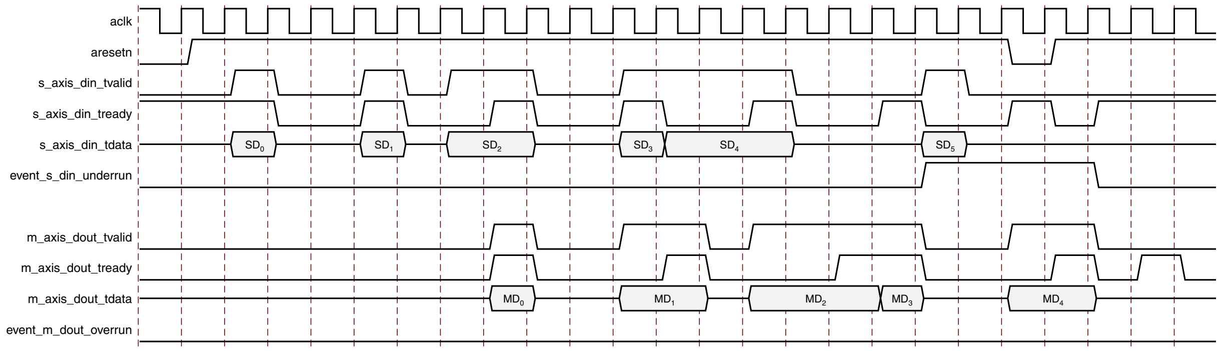


Figure 3-10: **Data Timing Diagram 1**

After the core is synchronized by the slave interface, it asserts `m_axis_dout_tvalid` after a latency given by Equation 3-4 and Equation 3-5 and thereafter periodically asserts `m_axis_dout_tvalid` to indicate valid output samples on the port `m_axis_dout_tdata`. The data rate specified determines how many cycles are available to read this data. The core expects the data to be read by asserting `m_axis_dout_tready` before the next output data sample is available. Output data could be read either immediately, a cycle after, or just prior to the next output data becoming available. After `m_axis_dout_tready` has been asserted to indicate the sample has been read, `m_axis_dout_tvalid` is deasserted unless the next output sample is available on the following clock cycle; in this case M_AXIS_DOUT_TVALID remains asserted while `m_axis_dout_tdata` is updated with new output data.

If an output sample is not read by asserting `m_axis_dout_tready` before the next is available, then the overrun condition occurs. As the core is free-running, overrun causes the next output sample to be dropped internally, as the current one has not been read and remains on the output port `m_axis_dout_tdata`. Overrun is signaled by asserting the `event_m_dout_overrun` flag. This flag remains set until the next core reset operation (by asserting `aresetn`). Figure 3-11 shows an example timing diagram where data rate = 3 clock-cycles/sample.

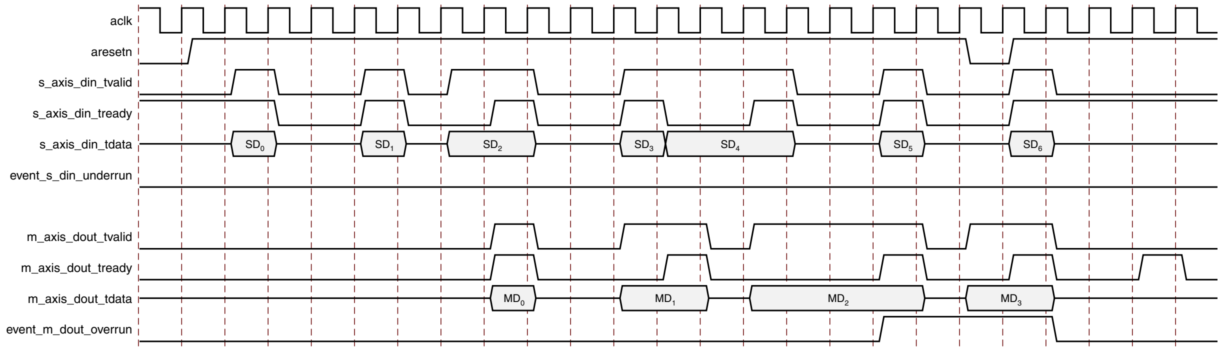


Figure 3-11: **Data Timing Diagram 2**

Two Pulse Configurable Coefficients Mode

In Static mode, two-pulse switching allows some flexibility in adapting to the spectrum of a changing signal. When this option is selected in Static mode, two coefficient memories RAM are available.

When the Filter Select register is set to zero, the first pulse stored in the pulse coefficients RAM (Pulse 0) is active. When Filter Select is set to one, the second pulse stored in the pulse coefficients RAM (Pulse 1) becomes active. For example, if the Filter Select register is zero and the pulse needs to adapt, the new coefficients should be written into the corresponding Pulse 1 RAM locations and then the Filter Select register can be set to one. The converse toggling process can continue for subsequent changes.

Note: In this scenario, the CP coefficients have to be manually loaded and the Filter Select register has to be written to; therefore, this can be used for very slow signal variations only.

The speed at which the pulses can be adapted depends on the controller at the data interface. With, for example, a 50 MHz clock and a cancellation pulse length of 1023, new coefficients can in principle be written in 15 µs; however in practice the time taken to recalculate for the new frequency content needs to be taken into account. If a microcontroller is used, then other factors come into play; however adaptation rates of less than one millisecond would seem reasonable to achieve.

When the active pulse is switched, that is, the Filter Select register is changed, any active CPGs at that time continue using the previous pulse until they have completed their operation and become inactive again. Pulse switching causes a transient in the output data. This should not be an issue for spectral or error-rate considerations, but if there is a blank period in the data, this could be advantageously used as the switching point.

Dynamic Mode

Dynamic mode should be used only when the power or frequency of the incoming data is changing dynamically. In Dynamic mode, sideband information for power and frequency is provided on the AXI4-Stream `TUSER` (used for slot marker) and AXI4-Stream dynamic control (used for power, frequency and power marker) interface. Dynamic Power Variation and Frequency Hopping explain the scenarios with and without frequency hopping (dynamic power variations only) in detail. Dynamic CP Computation provides details of CP calculation. Also, in Dynamic mode, there is a debug register that can mimic the Static mode in Dynamic mode and is explained in Force Static in Dynamic Mode.

IMPORTANT: *Dynamic Power variation refers to power variation in individual carrier powers. However, the variation in "total" input power due to these variations should not be more than 2–3 dB. If the total power reduces further, there are increasingly less peaks to be cancelled and the PC-CFR core passes through (when total input power is below, for example, –18 dBFS). The maximum input power (or rated power) occurs when all the carriers are loaded with full power and this needs to be mapped to –15 dBFS, so that any subsequent variation is less than this power.*

Dynamic Power Variation

For applications where the carrier power changes dynamically, the pulse coefficients must be computed dynamically. This mode is used when frequency hopping is disabled, and is recommended for wideband data where carrier configuration remains unchanged but individual carrier power changes. The slot marker information is input on the AXI4-Stream TUSER (s_axis_din_tuser) interface and is aligned to the data as shown in Figure 3-12. The relative power of each carrier is fed serially-aligned with the power marker pulse (the number of carriers per RAT is known from the core customization). The power marker pulse can arrive independently on each antenna. Relative carrier power can be computed by power meters (PM) external to the PC-CFR core, which can generate sideband signals (8-bit relative power and power marker) needed for power dynamics. The PM can be a simple moving average estimator measuring carrier powers over a preset. Relative carrier power and power marker are part of AXI4-Stream dynamic control channel TDATA and TUSER ports respectively, as detailed in Table 2-1.

The relative power values are on s_axis_dyn_ctrl_tdata[7:0] and the power marker appears on s_axis_dyn_ctrl_tuser[0] for the first antenna. Ports scale with the number of antennas. The slot marker port s_axis_din_tuser[0] can be set to zero if unused or unavailable from the baseband interface. The frequency information can be programmed through the AXI4-Lite slave interface by programming frequency registers. A dynamic CP computation is triggered whenever an individual carrier power is changed by an AXI4-Lite write to the Power Change Threshold register value. If this value is set to 0, every set of power input triggers a CP computation. The minimum time gap between the arrival of any two power markers (T_{update}) is dependent on whether the mode of computation is fast or slow (see Dynamic CP Computation). For a 245.76 MHz clock, the core can respond to power variations every 50 µs with fast computation and every 250 µs with slow computation. In dynamic power mode, up to five RATs are allowed and can have a maximum of 30 carriers that is six carriers per RAT.

RECOMMENDED: *Any modification in the AXI-Lite programmable frequency values take effect only when a power change is detected. For constant power input, it is recommended that the Power Change Threshold register be set to 0.*

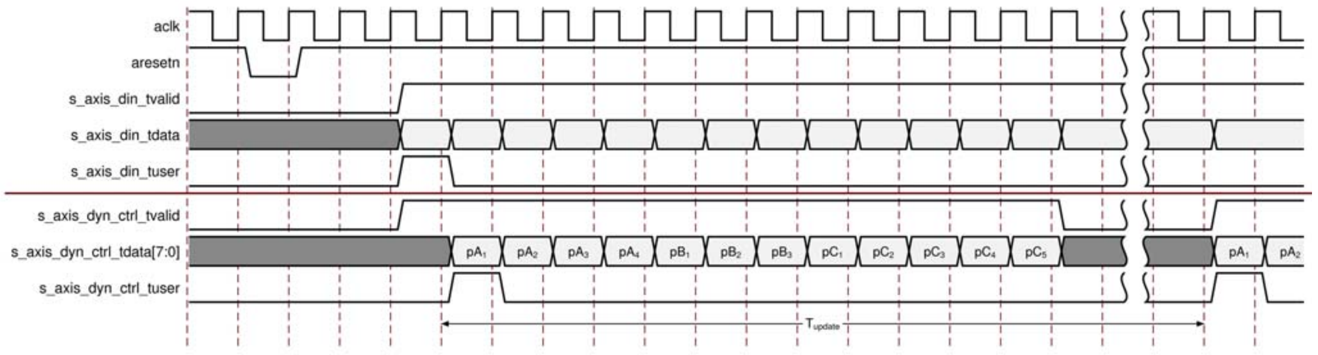


Figure 3-12: **Dynamic Mode with Frequency Hopping Disabled**

Frequency Hopping

For applications where the frequency is changing dynamically (such as in frequency hopping GSM), the pulse coefficients can be computed dynamically. This mode is enabled when frequency hopping is selected. Only one RAT is enabled in the frequency hopping mode and it can have up to eight carriers.

The frequency and power information is input on the AXI4-Stream dynamic control TDATA (`s_axis_dyn_ctrl_tdata`) interface as shown in Figure 3-13. The TDATA and TUSER ports are detailed in Table 2-1. The slot marker is input on `s_axis_din_tuser` aligned with the data. The power marker corresponds to `s_axis_dyn_ctrl_tuser` and 8 bit power values correspond to `s_axis_dyn_ctrl_tdata`[7:0]. The frequency corresponds to `s_axis_dyn_ctrl_tdata`[31:8]. Computation of the 24-bit frequency values is further explained in Dynamic CP Computation.

IMPORTANT: *The frequency information is expected to arrive one slot in advance and arrives aligned to the slot marker. As a result, the very first transmission slot must have all zero data with frequency information for the next slot to avoid any spectral splatter. The power information for the current slot arrives aligned to a power marker with an offset to the slot marker. The power information is expected to arrive within a window that is register-configurable.*

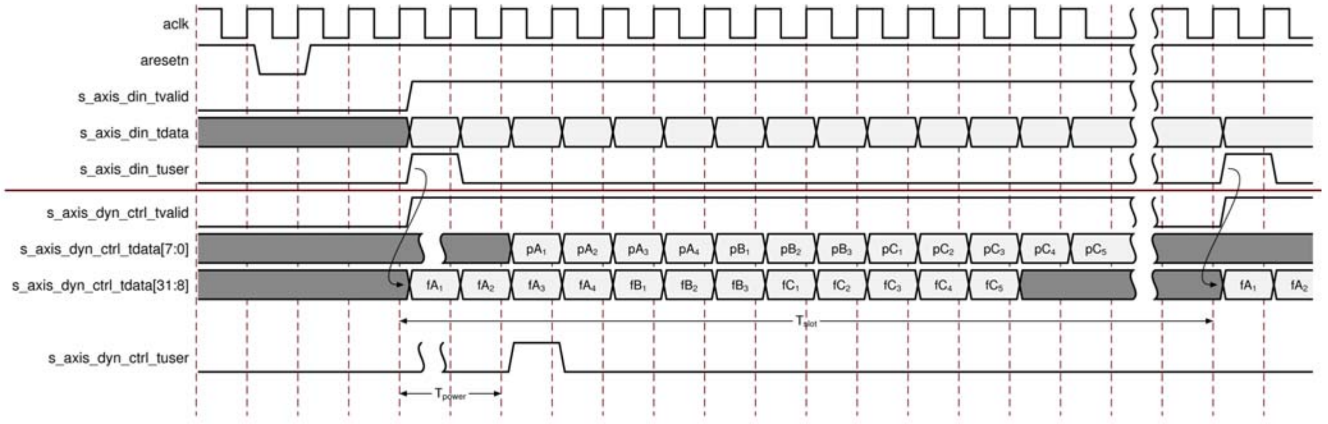

aclk
aresetn
s_axis_din_tvalid
s_axis_din_tdata
s_axis_din_tuser
s_axis_dyn_ctrl_tvalid
s_axis_dyn_ctrl_tdata[7:0]
s_axis_dyn_ctrl_tdata[31:8]
s_axis_dyn_ctrl_tuser
pA1 pA2 pA3 pA4 pB1 pB2 pB3 pC1 pC2 pC3 pC4 pC5
fA1 fA2 fA3 fA4 fB1 fB2 fB3 fC1 fC2 fC3 fC4 fC5 fA1 fA2
Tslot
Tpower

Figure 3-13: **Dynamic Mode with Frequency Hopping Enabled**

Figure 3-14 shows the definition for the AXI-Lite programmable register values used specifically for frequency hopping support with respect to slot marker that occurs at the beginning of each slot. These register values are unused when frequency hopping is disabled. T_{power} corresponds to the register value, Power Marker Window Start.

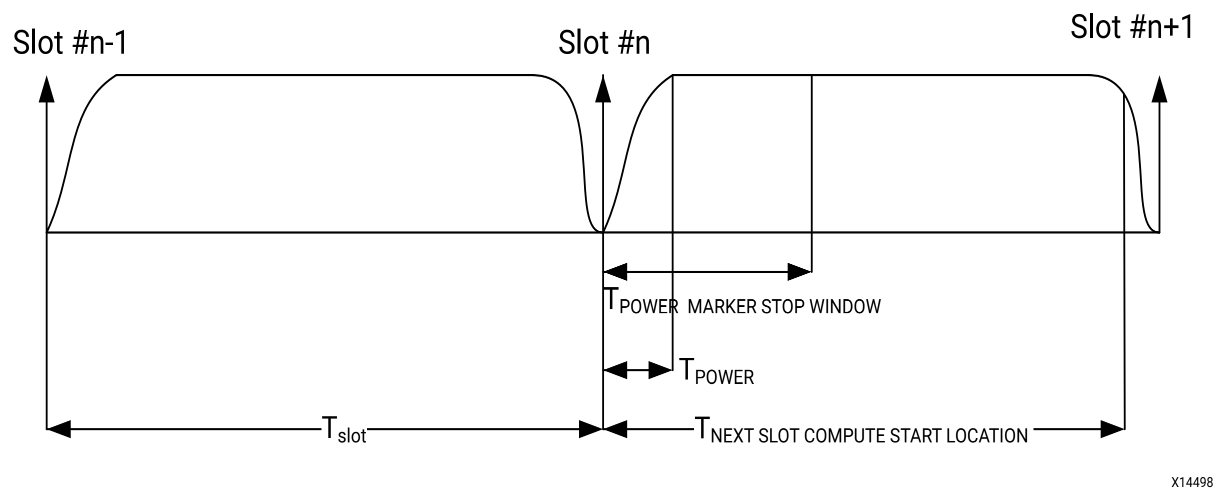

Slot #n-1
Slot #n
Slot #n+1
T POWER MARKER STOP WINDOW
T POWER
T slot
T NEXT SLOT COMPUTE START LOCATION
X14498

Figure 3-14: **Dynamic Mode with Frequency Hopping Enabled**

During each slot, the CP is computed twice; once for the current slot when the power values arrive through the TUSER interface during the register configurable window (this happens only if the power change for any carrier is greater than the register value Power Change Threshold) and once towards the end of the current slot (Next Slot Compute Start Location); this is used for the next slot.

In Figure 3-14 computation for the next slot occurs from “Next Slot Compute Start Location”. The choice of this location is dependent on the nature of the power ramp down towards the end of the slot. Let T_{slot} be the duration of the slot. Assuming $T_{ramp\_down\_locn}$ to be the location (measuring from the slot marker) where the signal power has fallen significantly below the PC-CFR threshold (so that no new peaks from the current slot are picked up),

$$T_{ramp\_down\_locn} < T_{NEXT\ SLOT\ COMPUTE\ START\ LOCATION} + T_{dyn\_impl} < T_{slot}$$

where $T_{dyn\_impl} = ((L_{max} + 1)/2 + 72)/(f_s * \text{data rate})$, L_{max} is the maximum CP length and the term 72 comes from the hardware implementation delays. The T_{dyn_impl} is based on Equation 3-1 in Dynamic CP Computation.

The power marker is sampled by the core only if it arrives between T_{POWER} and $T_{POW\_MARKER\_STOP\_WINDOW}$. The power input for the current slot is available only after a power computation has been done on the current slot using a power meter. This in turn happens only when the current slot has ramped up in power. T_{POWER} should be chosen to account for this delay. In addition, $T_{POWER} > T_{data}^{pc\_cfr}$ for the chosen PC-CFR v6.2 configuration as given in Equation 3-4 and Equation 3-5.

$T_{\text{POWER MARKER STOP WINDOW}}$ should be chosen as $T_{\text{POWER}} < T_{\text{POWER MARKER STOP WINDOW}} < T_{\text{slot}}/2$

The window for taking in power ensures that any unintended power markers (for example from a free running power meter) also get ignored and this can be used to simplify the power meter design. The register values to be fed into the core through the AXI4-Lite interface should be obtained by multiplying the corresponding values with f_s*data rate and flooring the value. It must be noted that these register values are computed with the core clock and not the sampling clock.

Example Computation for FH-GSM

Assume a 245.76 MHz core clock with FH-GSM slot duration = 577 µs. At 574 µs, the signal has ramped down sufficiently so that no new peaks are expected beyond this location. Hence, $T_{\text{ramp\_down\_locn}}$ = 574 µs.

Assume L_{max} = 2047.

The $T_{\text{NEXT SLOT COMPUTE START LOCATION}}$ is:

574-(1024+72)/245.76 = 569.54 µs

This translates to a register value of floor(569.54*245.76) = 139970.

Assume 16 µs is the time required for the signal to ramp up so that the power meter can do a correct power estimation, the T_{POWER} can be set to 16 µs and $T_{\text{POWER MARKER STOP WINDOW}}$ can be set to 20 µs which translates to register values of 3932 and 4915 respectively.

Dynamic CP Computation

The dynamic CP computation module computes the cancellation pulse based on Equation 3-1.

$$c_n = \frac{1}{\sum_{i=1}^{N} g_i} \sum_{i=1}^{N} h_{n,i} \cdot g_i \cdot e^{j\omega_i in}, n \in [-L,L] \qquad \textit{Equation 3-1}$$

Here, c_n is the multi-carrier composite CP, h_{ni} is the base CP for the i^{th} carrier of length 2L+1 with h_{0i}=1, for every carrier i, N is the number of carriers, g_i are the carrier gains and ω_i are the discrete

carrier frequency-offsets ($\omega_i = 2\pi\frac{f_i}{f_s}$, where f_i is the i^{th} carrier frequency offset and f_s is the sampling rate of incoming samples). Without loss of generality, the base CPs can be assumed to be symmetric with a peak at n=0, that is, $(h_{0i} = |h_{0i}| = 1)$ with, $|h_{0i}| > |h_{ni}|, n \neq 0, \forall i$, while 2L+1 is the length of the longest base CP, other shorter CPs must be zero padded suitably. Thus Equation 3-1 forms the basis of composite CP formulation.

The inputs to this module are 24-bit frequency and 8-bit power values. The g_i are internally computed inside the core from the 8-bit power values and are normalized such that they sum to unity.

IMPORTANT: *Whenever smart peak processing is enabled, the base pulses are computed with 4 times* f_s *and the frequency input format is round* $(f_i/(4*f_s)*2^{24})$.

Dynamic CP Computation for Multi-RAT Cases with Power Dynamics

As per the Dynamic CP Computation, the base CPs have to be generated as (h0i = 1). In case of Multi-RAT scenarios this introduces extra attenuation to higher bandwidth channels. So the input 8-bit power values have to be scaled as per the bandwidth of the individual carriers as shown in Equation 3-2.

$$P_i = round\left(\frac{255 \times BW_i \times Pmes_i}{K}\right)$$ *Equation 3-2*

Here P_i is the 8-bit input power data for the ith carrier, BW_i is the bandwidth of that carrier, and $Pmes_i$ is the measured power for that carrier. The normalization factor K should be selected to get an appropriate tradeoff between dynamic range and resolution.

For example, K may be selected such that the maximum power value for P_i (255) corresponds to a power of -15dBFS for a carrier with maximal bandwidth MBW (e.g. MBW = 20MHz):

$$K = MBW \times 10^{-15/10}$$ *Equation 3-3*

Dynamic Mode Computation

PC-CFR v6.2 can be configured in various dynamic modes such as Frequency-Hop (FH), slow computation and fast computation. While FH mode is aimed for FH MC-GSM, where CP has to be computed every 577 µsec based on frequency (one slot ahead) and power information, other modes are for power dynamics assuming carrier configuration does not change on-the-fly. Thus in power dynamic modes, frequency can be programmed through AXI4-Lite memory mapped registers (starting at offset 0x040, round(fc/fs*2^32)) as 32-bit 2's complement values. You have two options to choose Power dynamic modes. They are slow and fast computation. In slow computation, the compute resource is shared across the carriers.In fast computation, each carrier has a dedicated compute engine. Hence, it needs more DSP48s. Each compute engine uses 2 DSP48s for CP coefficient update. Hence, 2N DSP48s will be used for N carriers in fast computation. In slow computation it's just 2 DSP48s for N carriers. Consequently, update time will be more in slow computation. Assuming CP length of 2L+1, compute time for slow computation will be approximately 3N

(L+1)/2 cycles. However, the fast computation needs 3(L+1)/2 cycles only as N compute engines process CP updates concurrently. Figure 3-12 displays the timing diagram of power dynamic mode.

Figure 3-12 displays that the power marker and 8-bit relative power for five RATs with individual carriers are input signals to the core as part of AXI4-Streaming `tuser` bus. Note that the number of RATs and carriers for each of the RATs are generic parameters configured during core creation from Vivado™ IP Catalog. PC-CFR v6.2 supports a maximum of eight carriers per RAT up to three RATs, and six carriers per RAT for four or five RATs. The power marker signal indicates the time that the core has to sample relative power coming from external power meters (PM). CP computation starts after sampling all the carrier power values as shown in figure.

For example,

6c WCDMA with 1023 CP length and core clock fcore = 245.76MHz

Slow computation needs 1.5x6x512/fcore = 18.75 µsec while fast computation needs only 3.125 µsec.

However, power meter may need to average over a few hundred µsec of data samples for a reliable relative power measurement and this is a narrow time window where old CP will get applied to the new peaks. This "dead-zone" window can be made narrower by reducing the PM averaging time which in turn increases the inaccuracy in relative power measurement and impacts CP scaling.

Application of new CP is glitch-free, i.e., any new peaks detected after power marker plus CP compute delay is canceled with newly computed CP (if the CPGs are available) while the peaks prior to this will be canceled with old or previously computed CP. Besides, there is an AXI4-Lite memory mapped register (offset: 0x074) for programming a power change threshold value. If at least one carrier power difference (from the previous CP computation relative power profile) is more than this programmed value, then the CP computation is triggered.

Smart Peak Processing Disabled

The frequency input format is $\mathrm{round}(\frac{f_i}{f_s} \times 2^{24})$. The sampling rate, f_s in this mode is oversampled typically by a factor of 4. Consider a 2-carrier LTE 5 MHz case with carrier centers at –20 MHz and 15 MHz. The recommended sampling rate, f_s is 245.76 MHz. In this case, the cancellation pulse and frequency are computed with the same f_s.

Smart Peak Processing Enabled

The frequency input format is $\mathrm{round}(\frac{f_i}{4f_s} \times 2^{24})$. Sampling rate f_s in this mode is mode does not need any oversampling. Consider a 2-carrier LTE 5 MHz case with carrier centers at –20 MHz and 15 MHz. The recommended sampling rate f_s is 61.44 MHz. In this case, however the cancellation pulse and frequency are computed with $4f_s$.

Dynamic CP computation can be done in fast (parallel) or slow (serial mode). In parallel mode, the computation happens in parallel for each carrier whereas in serial mode, a single engine is multiplexed for the computation across all the carriers. Thus, in the parallel or fast mode, the computation takes approximately $3(L_{max}+1)/4$ cycles and in the serial or slow mode it takes $3*N*(L_{max}+1)/4$, where L_{max} is the maximum CP length setting and N refers to the sum of all the carriers across all the RATs being used. In frequency hopping mode, dynamic CP computation always uses fast mode. Note that the dynamic CP computation operates at the core clock and not the sample rate clock.

Force Static in Dynamic Mode

The debug registers "Force Static in Dynamic" and "CP initialization Enable" can be used to replicate Static mode in Dynamic mode. When this is done, CP coefficients can be loaded exactly as in Static mode and the core uses these AXI4-Lite-configured CP coefficients for pulse cancellation. The sequence of steps is as follows:

1. Set register "Force Static in Dynamic" to 1
2. Load the CP coefficients as done in Static mode (the address space is same as that used in Static mode for Pulse 0 Coefficients RAM)
3. Set register "CP initialization Enable" to 1

Statistics Registers

CFR statistics can be used to study the behavior of the core. Registers have been provided that give access to certain CFR statistics. These registers can be read back using the AXI4-Lite control interface. The following statistics registers are supported:

- Maximum number of CPGs used per iteration.
- Number of peaks missed in the last iteration of PC-CFR.
- Number of hard-clipped samples/number of peaks processed by WCFR.

Applications

To configure the PC-CFR core for a particular application, decisions must be made on:

- Sample Rate of Operation
- Number of Iterations

- Number of CPGs at Each Iteration
- Pulse Shape Coefficients
- Latency
- Threshold Value(s)
- Peak Detect Window Length

The following sections provide guidelines for these topics.

RECOMMENDED: It is recommended that simulation be used prior to configuration for a particular application.

A MATLAB product reference model and supporting functions are provided with the core and several examples are given for use as templates.

General Considerations for Parameter Selection

Sample Rate of Operation

The choice of sampling rate of the incoming data affects the performance of CFR. When using PC-CFR mode with smart peak processing disabled, the sampling rate (f_S) of CFR should be at least three times the signal bandwidth for single carrier and contiguous multi-carrier spectra. In non-contiguous carrier configuration case, `4 x iBW` is recommended sampling rate for the core to function properly. However, with smart peak processing enabled, sampling rate can be brought down to `~1.2 x iBW` for the same performance.

As with all parameters, there is a trade-off in play between the PAR that can be achieved, EVM, and other factors. Typically CFR operates at the sample rate of DPD, that is, it is placed between the Digital up Converter (DUC) and Digital Pre-Distortion (DPD) blocks in the transmit chain. Typically DPD is specified to operate at five to six times the signal bandwidth (although often performance is not compromised at lower rates for contiguous spectra). However, if, for example, simulation shows that three times is sufficient for CFR but DPD still requires a higher rate, there is no reason why the signal should not be interpolated between CFR and DPD. Measuring the achieved PAR at the CFR sample rate is misleading. Ultimately, the signal is upsampled to the analog domain. Therefore CFR should be evaluated by measuring the PAR on a simulated upsampled output signal, upsampled to a high enough rate for the regrowth due to analog conversion to be accounted for. This is the method used in the example results reported.

Number of Iterations

For common applications where 2–3 dB of PAR reduction is required and EVM budgets are 5% or more, typically two iterations are required. More than two iterations might be needed for difficult cases—non-adjacent carriers where a low PAR is needed, and multi-carrier GSM. For a signal where the required PAR reduction is lower, because of EVM constraints such as in the case of WiMAX, one iteration might be sufficient. It is recommended to use WCFR as a 'last iteration' in the post-processing stage. This reduces the total number of required PC-CFR iterations. When using PC-CFR with Smart Peak Processing enabled, a post-processing stage consisting of WCFR is mandatory and is generated by default. This stage detects any peaks missed in the initial iterations and limits them. The parameters given below are applicable only to PC-CFR stage and not WCFR.

Number of CPGs at Each Iteration

IMPORTANT: *This section is applicable only to the PC-CFR stage and not WCFR.*

The typical number is three to four, but there are special cases. The number depends on the signal type, sample rate, and the final PAR that needs to be achieved. The supplied simulator allows for different CPG numbers to be provisioned and reports the maximum number of CPGs used in each iteration. The number of CPGs provisioned can be different at each iteration to save resources if the performance is assured.

Pulse Shape Coefficients

IMPORTANT: *This section is applicable only to the PC-CFR stage and not WCFR.*

The cancellation pulse coefficients are the impulse response of a low-pass filter that matches, to within an approximation, the frequency response of the transmitted signal. It does not need to have the exact pulse shape of the signal, nor does it need the degree of attenuation in the stop band.

The time domain requirements on the cancellation pulse coefficients are that in its main lobe it broadly matches the shape of the signal peaks, as shown in Figure 3-4. This requirement means that the spectra must broadly match. For the out-of-band performance, one might think in terms of clipping noise being introduced in the process of canceling peaks with the clipping noise at some dB level relative to the signal, and that the requirement on the cancellation pulse is to filter the clipping noise down to a spectrally compliant level. Strictly speaking, this is not an accurate view because the PC-CFR core does not actually clip the signal. However, in a particular application this process can be implied. For example, in the single-carrier WCDMA example, the response of the carrier and the signal before and after CFR are shown, and from these plots it can be inferred that there is an equivalent clipping noise approximately 20 dB below the carrier.

For a single-centered carrier, the cancellation pulse is the impulse response of a suitable low-pass filter. There is no general theory as to how this should be designed. Experience shows that any reasonable method can be used. Parameters are shown here for a constrained equiripple method that seems to give good results across the example cases studied. In previous Xilinx literature, the least squares method was specified to equally good effect. The constrained equiripple method is easier to parameterize.

The MATLAB software signal processing toolbox function for constrained equiripple is

firceqrip(order, F_c/(f_s/2), [D_{pass}, D_{stop}], 'slope', 0);

and suitable parameters for the air interface standards explicitly supported are shown in Table 3-1.

Table 3-1: **firceqrip Parameters**

Standard	Order[1]	F_c	D_{pass}(dB)	D_{stop}(dB)
WCDMA	3.3*f_s	1.85	–20	–65
CDMA2000	4*f_s	0.45	–20	–65
WiMAX (10 MHz BW)	2*f_s	4.2	–20	–65
LTE (5 MHz BW)	3.3*f_s	2	–20	–65
LTE (10 MHz BW)	2*f_s	4.3	–20	–65
LTE (20 MHz BW)	1.3*f_s	9.0	–20	–65
TD-SCDMA	4*f_s	0.5	–20	–65
GSM	8.3*f_s	0.1	–20	–70

Notes:

1. The order values should be rounded to the nearest even number below the calculated values.

For the actual call to `firceqrip`, the D_{pass} and D_{stop} values must be converted to linear scaling using, for example: D_{stop} = 10^(D_{stop_dB}/20);

The supplied simulation files contain an Excel spreadsheet with the aforementioned single-carrier filters at 61.44 MSPS for use if firceqrip or equivalent is not available. They can be resampled for other sample rates.

IMPORTANT: *When Smart Peak Processing is enabled, the computation shown above should be done with 4xf_s instead of f_s. Thus the generated pulse corresponds to a sampling rate of 4xf_s and this oversampled pulse should be loaded into the PC-CFR core.*

For multiple carriers, the single-carrier filter should be frequency shifted and summed to match the carrier configuration of the signal. For use in the core, the final pulse must be quantized and scaled such that the center value is 16384. The function rotate_and_scale in the supplied MATLAB software simulation files is a template for this operation. Individual carrier scaling can be introduced for non-uniform power configurations.

The PC-CFR core can be used for mixed-mode signals; pulses for different air interfaces have to be scaled by their relative power levels before combining to form the mixed-mode pulse. The supplied examples serve to further illustrate the processes described here.

Latency

Latency of the core is primarily a function of the length of programmed cancellation pulse (filter length, offset: 0x0) and a parameter called Max Peak Detect Window. The Max Peak Detect Window parameter determines the range of values that can be taken by the peak detect window. A low value of peak detect window can result in missing detection of peaks, whereas a high value can avoid the missing of peaks but results in increased core latency. The detailed computation of latency for the entire CFR module including the post processing stage is discussed below.

When the core is configured in PC-CFR mode, then the range for latency is given by Equation 3-4. When the core is configured in WCFR standalone mode then latency is given by Equation 3-5. Latency is a function of number of iterations (N_{iter}), Data Rate (*data_rate*), cancellation pulse length (L_{CP}) and the Max Peak Detect Window parameter. Note that the first valid output has a pre-defined latency from the first valid input given by Equation 3-6.

Note: Delays mentioned in all equations are in terms of samples.

Max Peak Delay is computed from Max Peak Detect Window setting as follows:

$$T_{max\_peak\_delay} = 32 \text{ (if max\_peak\_detect\_window} < 29)$$

$$T_{max\_peak\_delay} = \text{max\_peak\_detect\_window} + 3 \text{ (if max\_peak\_detect\_window} \geq 29)$$

Therefore,

$$T_{\text{data}}^{\text{pc-cfr}} = \left(\left(\frac{L_{CP}-1}{k}\right) + T_{\text{max\_peak\_delay}} + T_{\text{imp}}\right) \times N_{\text{iter}} + T_{\text{AXI}} + T_{\text{iter}} + T_{\text{pps}} \qquad \textit{Equation 3-4}$$

Note that,

k=2 when smart peak processing is disabled,
k=8 when smart peak processing is enabled.

Therefore,

$$T_{\text{data}}^{\text{wcfr}} = T_{\text{wcfr}} + T_{\text{AXI}} \qquad \textit{Equation 3-5}$$

$$T_{\text{valid}} = N_{\text{iter}} + T_{\text{AXI}} + \left\lfloor \frac{2 \times N_{\text{iter}}}{\text{data\_rate}} \right\rfloor \qquad \textit{Equation 3-6}$$

T_{imp} is a delay specific to implementation and is shown in Table 3-2; T_{AXI} = 2 is a fixed delay in the AXI4-Stream slave and master interfaces. The term T_{pps} in Equation 3-4 represents the delay due to the post processing stage. It does not contribute anything (that is, T_{pps} = 0) if the post processing stage is not present in the core. T_{pps} is computed as shown in Table 3-2 for both Static and Dynamic mode. The delay due to WCFR is denoted by T_{wcfr}. This is same as the column T_{pps} in Table 3-3 in case when SPP is enabled. The delay due to PC-CFR along with WCFR in post processing stage, when smart peak processing is disabled, is provided in

Table 3-4. For WCFR only, the Latency can be calculated using Table 3-5 when smart peak processing is disabled. Note that CPG_i indicates the number of cancellation pulse generators for the ith iteration.

Table 3-2: **Implementation Delay (Smart Peak Processing Disabled/Hard Clipper Post Processing)**

Data Rate	T_{imp}	T_{iter}	T_{pps}
1	45	$\sum_{i=1}^{N_{iter}} CPG_i$	19
2	36	$\sum_{i=1}^{N_{iter}} \left\lfloor \frac{CPG_i}{4} \right\rfloor$	9
3	34	$\sum_{i=1}^{N_{iter}} \left\lfloor \frac{CPG_i}{10} \right\rfloor + \left\lfloor \frac{2 \times N_{iter}}{3} \right\rfloor - 1 + \left\lfloor \frac{(N_{iter}+1) \text{ modulo } 3}{2} \right\rfloor$	6
4	31	$\sum_{i=1}^{N_{iter}} \left\lfloor \frac{CPG_i}{8} \right\rfloor + 3 \times \left\lceil \frac{N_{iter}}{2} \right\rceil - 2 - (N_{iter} \text{ modulo } 2)$	5
8	29	$3 \times \left\lceil \frac{N_{iter}}{2} \right\rceil - 1 - 2 \times (N_{iter} \text{ modulo } 2) - \left\lfloor \frac{N_{iter}+2}{4} \right\rfloor - \left\lfloor \frac{(N_{iter}-1) \text{ modulo } 4}{3} \right\rfloor$	3

Table 3-3: **Implementation Delay (Smart Peak Processing Enabled/WCFR Post Processing)**

Data Rate	T_{imp}	T_{iter}	T_{pps}
1	73	$\sum_{i=1}^{N_{iter}} CPG_i - 2$	$108 + \left(\frac{WCFR\_Window\_Length}{2}\right) + \left[log_2\left\{\frac{WCFR\_Window\_Length}{3}\right\}\right]$
2	51	$\sum_{i=1}^{N_{iter}} \left\lfloor \frac{CPG_i}{4} \right\rfloor - 2$	$87 + \left(\frac{WCFR\_Window\_Length}{2}\right) + \left[log_2\left\{\frac{WCFR\_Window\_Length}{3}\right\}\right]$
3	47	$\sum_{i=1}^{N_{iter}} \left\lfloor \frac{CPG_i}{10} \right\rfloor + \left\lfloor \frac{2 \times N_{iter}}{3} \right\rfloor - 2 + \left\lfloor \frac{(N_{iter} - 1) \text{ modulo } 3 + 2}{3} \right\rfloor$	$81 + \left(\frac{WCFR\_Window\_Length}{2}\right) + \left[log_2\left\{\frac{WCFR\_Window\_Length}{3}\right\}\right]$
4	40	$\sum_{i=1}^{N_{iter}} \left\lfloor \frac{CPG_i}{8} \right\rfloor + 3 \times \left\lceil \frac{N_{iter}}{2} \right\rceil - 2 - 2 \times (N_{iter} \text{ modulo } 2) - ((N_{iter} - 1) modulo 2)$	$76 + \left(\frac{WCFR\_Window\_Length}{2}\right) + \left[log_2\left\{\frac{WCFR\_Window\_Length}{3}\right\}\right]$
8	37	$3 \times \left\lceil \frac{N_{\text{iter}}}{2} \right\rceil - 2 - 2 \times (N_{\text{iter}} \text{ modulo } 2) - \left\lfloor \frac{N_{\text{iter}} + 2}{4} \right\rfloor - \left\lfloor \frac{(N_{iter} \text{ modulo } 4) + 3}{4} \right\rfloor$	$74 + \left(\frac{WCFR\_Window\_Length}{2}\right) + \left[log_2\left\{\frac{WCFR\_Window\_Length}{3}\right\}\right]$

Table 3-4: **Implementation Delay (Smart Peak Processing Disabled/WCFR Post Processing)**

Data Rate	T_{imp}	T_{iter}	T_{pps}
1	45	$\sum_{i=1}^{N_{iter}} CPG_i$	WCFR_Window_Length*2 + 305
2	36	$\sum_{i=1}^{N_{iter}} \left\lfloor \frac{CPG_i}{4} \right\rfloor$	WCFR_Window_Length*2 + 286
3	34	$\sum_{i=1}^{N_{iter}} \left\lfloor \frac{CPG_i}{10} \right\rfloor + \left\lfloor \frac{2 \times N_{iter}}{3} \right\rfloor - 1 + \left\lfloor \frac{(N_{iter}+1) \text{ modulo } 3}{2} \right\rfloor$	WCFR_Window_Length*2 + 274
4	31	$\sum_{i=1}^{N_{iter}} \left\lfloor \frac{CPG_i}{8} \right\rfloor + 3 \times \left\lceil \frac{N_{iter}}{2} \right\rceil - 1 - (N_{iter} \text{ modulo } 2)$	WCFR_Window_Length*2 + 273
8	29	$3 \times \left\lceil \frac{N_{iter}}{2} \right\rceil - 1 - 2 \times (N_{iter} \text{ modulo } 2) - \left\lfloor \frac{N_{iter}+2}{4} \right\rfloor - \left\lfloor \frac{(N_{iter}-1) \text{ modulo } 4}{3} \right\rfloor$	WCFR_Window_Length*2 + 266

Table 3-5: **Implementation Delay (Smart Peak Processing Disabled/WCFR Mode)**

Data Rate	T_{WCFR}
1	(2 x WCFR_Window_Length) + 305
2	(2 x WCFR_Window_Length) + 286
3	(2 x WCFR_Window_Length) + 274
4	(2 x WCFR_Window_Length) + 273
8	(2 x WCFR_Window_Length) + 266

Threshold Value(s)

The threshold should be set at the maximum signal amplitude to be transmitted (to the PA). It might be useful to think of the threshold in relation to the average signal power such that the ratio of the threshold to the mean becomes the desired PAR. In the simulations, the threshold is set by defining the `PAR_target_dB`, and the actual threshold linear number is derived from this in relation to the signal dBFS (db power relative to Full Scale). However, when considering Time Division Duplex (TDD) signals or types that have a non-constant power (in a short-term moving average sense), it might be better to think in terms of the absolute peak power targets and/or consider the signal time domain details.

The core supports only a single threshold setting for all iterations, unless using multiple instantiations. Normally the threshold settings at each iteration are the same, and there is little evidence to suggest that it should be otherwise.

The recommended input level to the core is –15 dBFS. Thus, for 16-bit I and Q inputs, the required signal standard deviation is $2^{15}*10^{(-15/20)}$ or 5827. The input signal needs to be scaled appropriately to maintain a –15 dBFS (or close to this level) input signal. Too low an input signal causes high level of quantization noise and might impact PC-CFR functionality, whereas too high a value might cause saturation/overflow at various iterations. The cancellation threshold needs to be set with respect to the input dBFS level. For example, a 6.5 dB PAR target needs the threshold to be set at $2^{15}*10^{((-15+6.5)/20)}$ or 12,315.

It is recommended that the hard clipper threshold be set 0.5 dB higher than the main PC-CFR iteration threshold to control peak PAR. In smart peak processing enabled mode, the WCFR threshold should to be set 0.1 to 0.2 dB higher than the PC-CFR iteration threshold for effective post processing.

Peak Detect Window Length

IMPORTANT: *This section is applicable only to the PC-CFR stage and not WCFR.*

The peak detect window length determines the window over which a peak is identified and is chosen based on the time correlation of the input signal. This is explained in more detail in the Xilinx Application Note, *Peak Cancellation Crest Factor Reduction in a Multi-Standard Transmit System* (XAPP1174) [Ref 2]. The peak detect window takes a minimum value of three and a maximum value of Max Peak Detect Window.

The peak detect window length for single-carrier and contiguous multi-carrier spectra can be left at the default. Non-default values come into play when non-adjacent carriers are present, and typically they help to reduce the EVM and the number of iterations needed to achieve the desired PAR. Non-default values are required for multi-RAT, GSM/EDGE and multi-carrier configurations. Examples of such carrier configuration specific values can be seen in MATLAB software simulation scripts in the PC-CFR evaluation lounge. A recommended starting point for f_s=245.76 MHz is [40, 20, 20 20] for 4 iterations using multi-carrier configuration. These values have to be scaled linearly with the sampling rate; thus for f_s=61.44 MHz the recommended values become [10, 5, 5, 5].

Protocol Description

AXI4-Stream

The conversion to AXI4-Stream interfaces brings standardization and enhances interoperability of Xilinx IP LogiCORE™ solutions. Other than general control signals such as `aclk` and `aresetn`, and the control interface which is implemented as an AXI4-Lite interface, all data interface inputs and outputs to the PC-CFR core are conveyed using AXI4-Stream channels. A channel consists of `tvalid` and `tdata` always, plus several optional ports and fields. In the PC-CFR core, the only optional port supported is `tready`.

For further details on AXI4-Stream interfaces see the *AMBA® AXI4-Stream Protocol Specification* [Ref 3]and the *Xilinx Vivado AXI Reference Guide* (UG1037)[Ref 4].

Basic Handshake

Figure 3-15 shows the transfer of data in an AXI4-Stream channel. `TVALID` is driven by the source (master) side of the channel and `TREADY` is driven by the receiver (slave). `TVALID` indicates that the value in the payload field, `TDATA`, is valid. `TREADY` indicates that the slave is ready to receive data. When both `TVALID` and `TREADY` are TRUE in a cycle, a transfer occurs. The master and slave set `TVALID` and `TREADY` respectively for the next transfer appropriately.

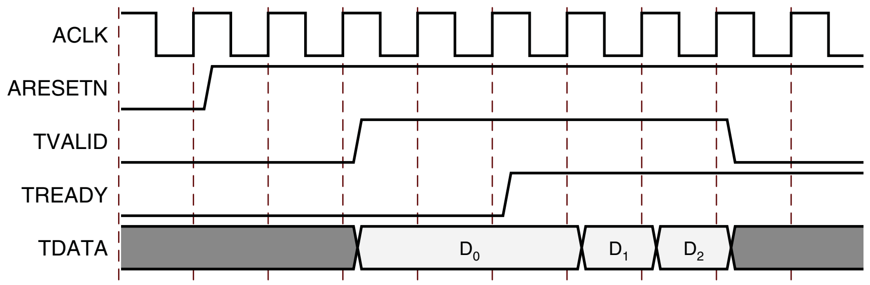


Figure 3-15: **Data Transfer in an AXI4-Stream Channel**

TDATA Packing

The TDATA field of the AXI4-Stream interface carries packed data. In complex data types, the real component is in the least significant position and the imaginary component follows, each are atomic subfields. To ease interoperability with byte-oriented protocols, each subfield within TDATA that could be used independently is first extended to fit a bit field that is a multiple of 8 bits. For example, if the PC-CFR core is configured to have a data width of 18 bits, each of the real and imaginary components of the data is 18 bits wide. The real component would occupy bits 17 down to 0. Bits 23 down to 18 would be ignored. The imaginary component would occupy bits 41 down to 24. Bits 47 down to 42 would also be ignored. Sub-field packing and byte-rounding are shown in Figure 3-16. The bits added by byte orientation are ignored by the core and do not result in additional resource use.

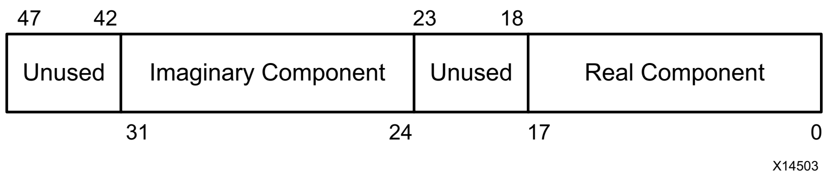


Figure 3-16: **TDATA Packing for DIN and DOUT Ports**

TDATA Structure for DIN and DOUT Channels

Input port DIN and output port DOUT carry packed and byte-rounded data in their respective TDATA fields. Multiple antenna support is implemented by expanding the bus width of TDATA and packing the multiple antenna streams as follows: The N parallel streams of data from N antennas are packed as antenna 1 data (LSW), antenna 2 data, ..., antenna N data (MSW).

Simulation Models

The PC-CFR core has several options for simulation models:

- VHDL UNISIM structural model
- Verilog UNISIM structural model

The models required can be selected in the project options.

Xilinx recommends that simulations utilizing UNISIM-based structural models are run using a resolution of 1 ps. Some Xilinx library components require a 1 ps resolution to work properly in either functional or timing simulation. The UNISIM-based structural models might produce incorrect results if simulation with a resolution other than 1 ps.

Chapter 4

Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 5]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 7]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 8]

Customizing and Generating the Core

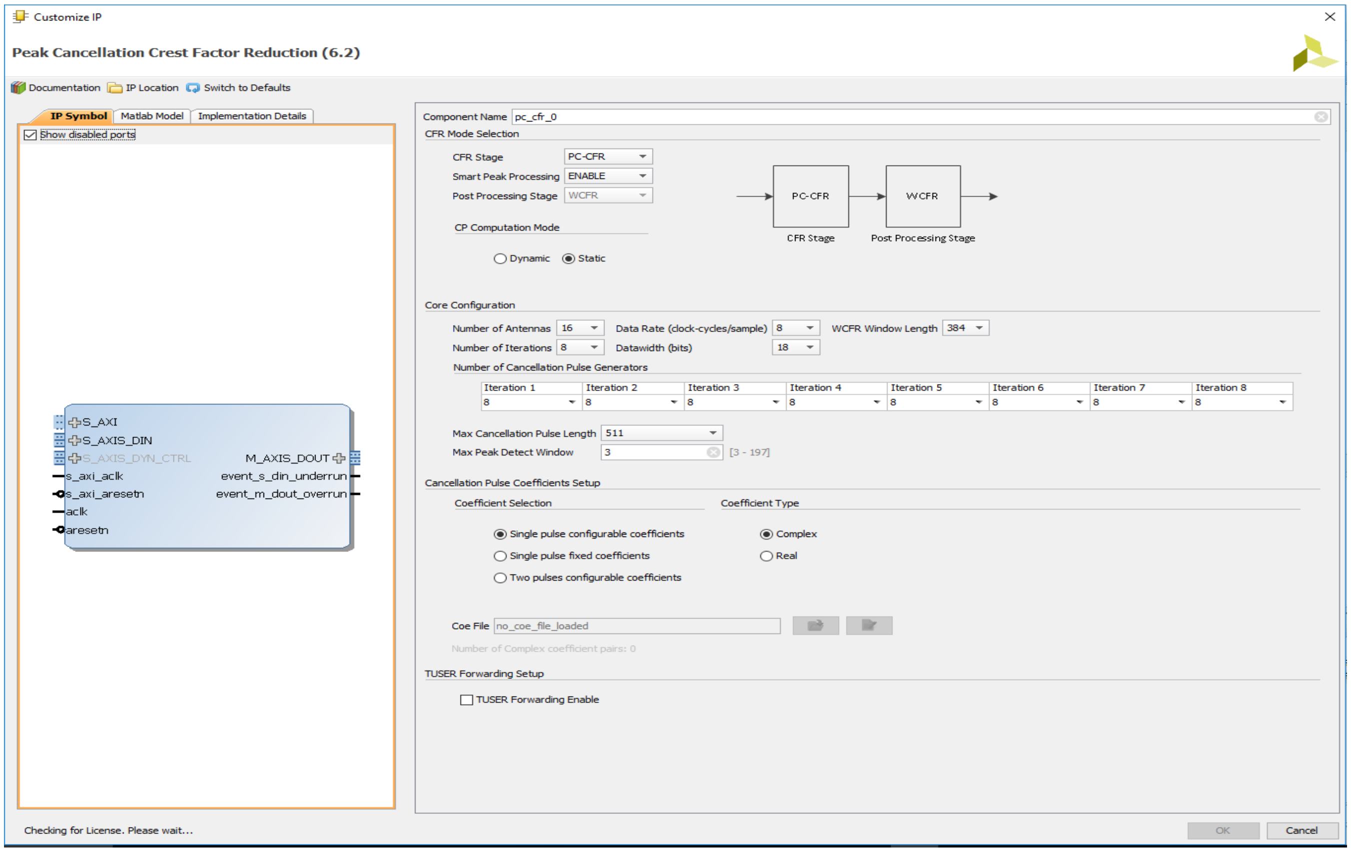
You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 7].

Note: Figures in this chapter are illustrations of the Vivado IDE. This layout might vary from the current version.

Three core configuration windows are shown in Figure 4-1, Figure 4-2, Figure 4-3, and Figure 4-4. Figure 4-1 shows the core interface in Static mode with smart peak processing and WCFR as the post-processing stage.



Peak Cancellation Crest Factor Reduction (6.2)

[✓] Show disabled ports

Component Name: pc_cfr_0

CFR Mode Selection

- CFR Stage: PC-CFR
- Smart Peak Processing: ENABLE
- Post Processing Stage: WCFR

CP Computation Mode: Dynamic [✓ Static]

Core Configuration

- Number of Antennas: 16
- Data Rate (clock-cycles/sample): 8
- WCFR Window Length: 384
- Number of Iterations: 8
- Datawidth (bits): 18

Number of Cancellation Pulse Generators

Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8
8	8	8	8	8	8	8	8

- Max Cancellation Pulse Length: 511
- Max Peak Detect Window: 3 [3 - 197]

Cancellation Pulse Coefficients Setup

Coefficient Selection: [✓ Single pulse configurable coefficients] Single pulse fixed coefficients Two pulses configurable coefficients

Coefficient Type: [✓ Complex] Real

Coe File: no_coe_file_loaded

Number of Complex coefficient pairs: 0

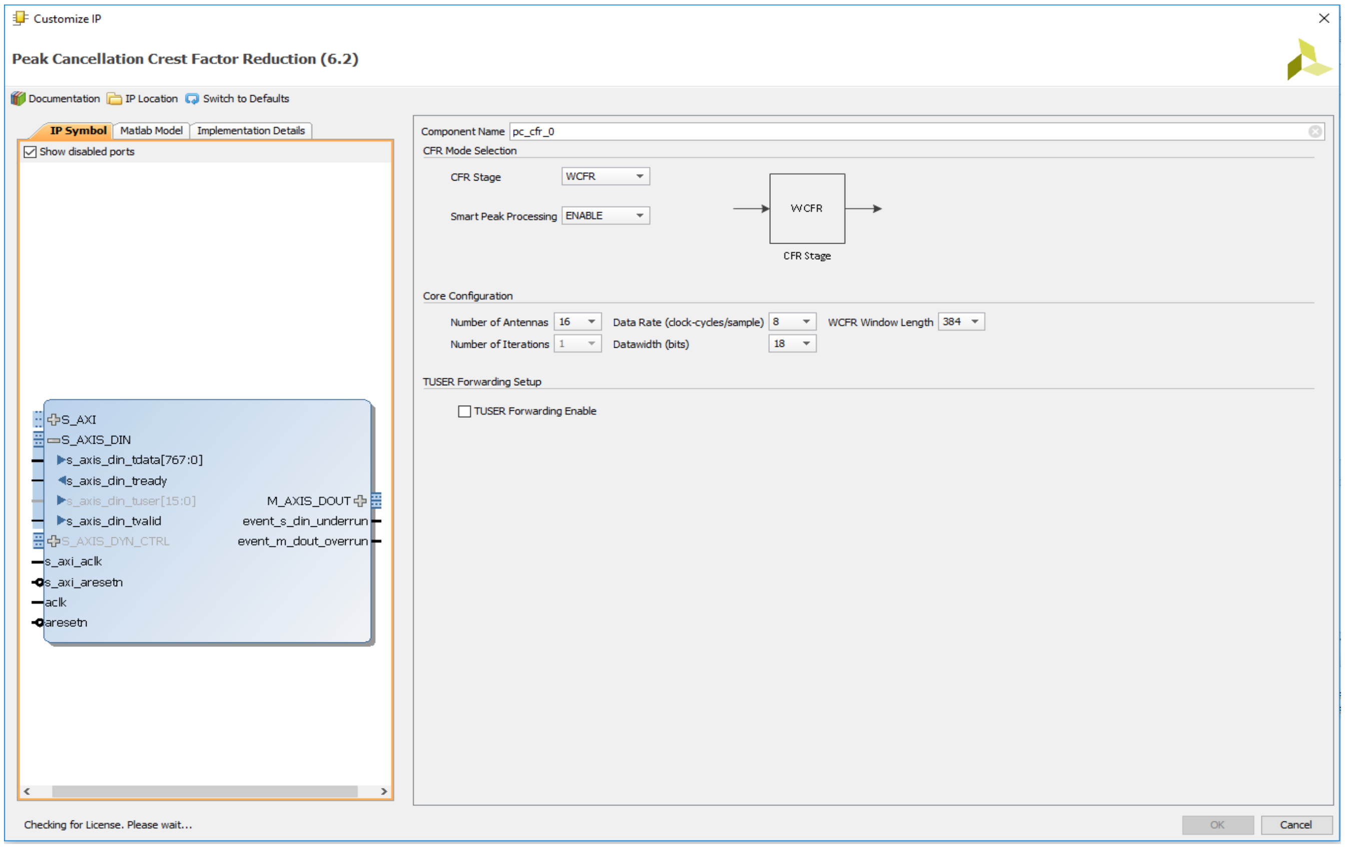
TUSER Forwarding Setup

[ ] TUSER Forwarding Enable

Checking for License. Please wait...

Figure 4-1: **Core Interface in Static Mode**

Figure 4-2 shows the core interface in standalone WCFR mode.



Peak Cancellation Crest Factor Reduction (6.2)

☑ Show disabled ports

Component Name: pc_cfr_0

CFR Mode Selection

- CFR Stage: WCFR
- Smart Peak Processing: ENABLE

Core Configuration

- Number of Antennas: 16
- Number of Iterations: 1
- Data Rate (clock-cycles/sample): 8
- Datawidth (bits): 18
- WCFR Window Length: 384

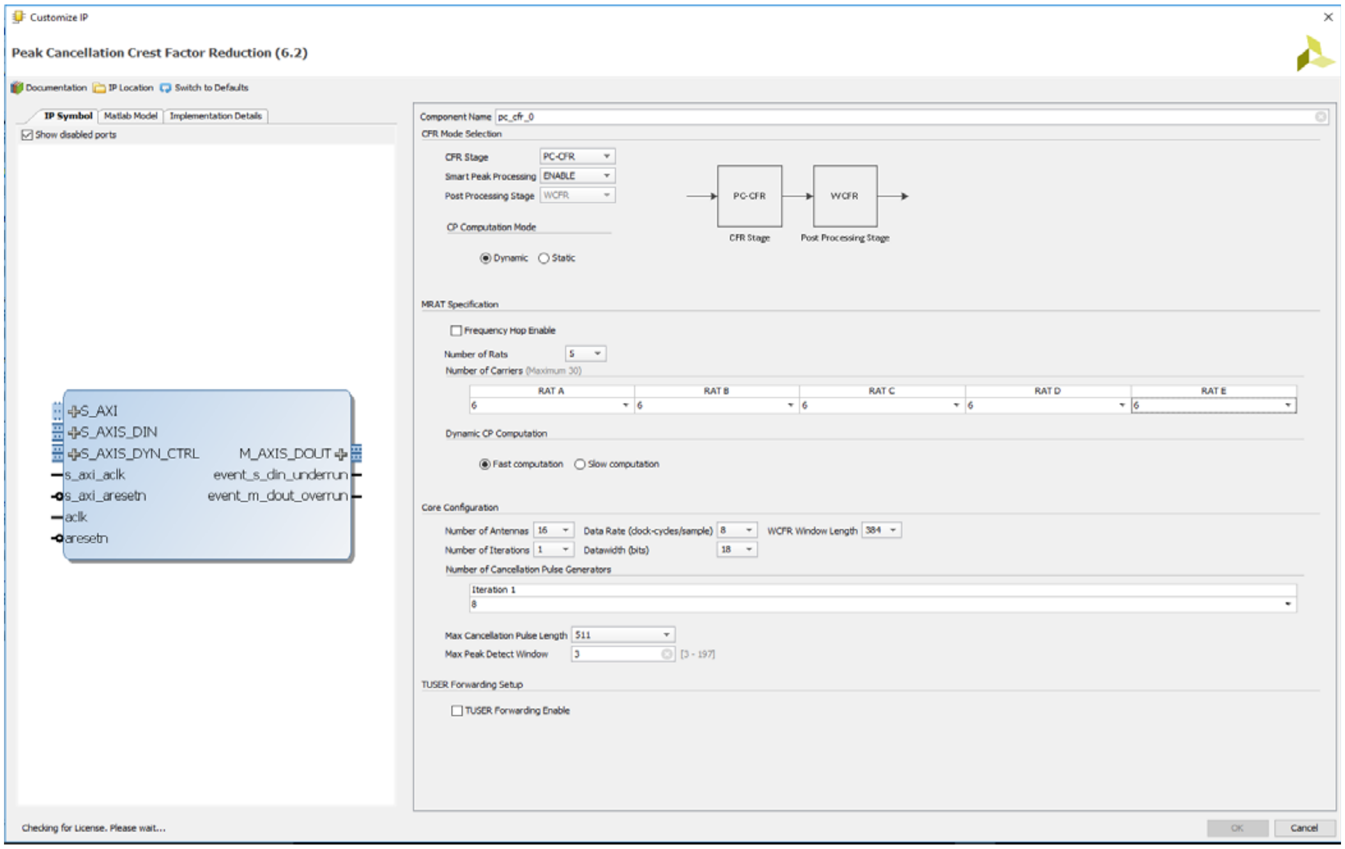
TUSER Forwarding Setup

☐ TUSER Forwarding Enable

Checking for License. Please wait...

Figure 4-2: **Core Interface in Standalone WCFR Mode**

Figure 4-3 shows the core interface in Dynamic mode with smart peak processing and WCFR as the post-processing stage.


Customize IP

Peak Cancellation Crest Factor Reduction (6.2)

Documentation IP Location Switch to Defaults

IP Symbol | Matlab Model | Implementation Details

☑ Show disabled ports

Component Name: pc_cfr_0

CFR Mode Selection

CFR Stage: PC-CFR

Smart Peak Processing: ENABLE

Post Processing Stage: WCFR

CP Computation Mode

[✓ Dynamic] Static

MRAT Specification

☐ Frequency Hop Enable

Number of Rats: 5

Number of Carriers (Maximum 30)

RAT A	RAT B	RAT C	RAT D	RAT E
6	6	6	6	6

Dynamic CP Computation

[✓ Fast computation] Slow computation

Core Configuration

Number of Antennas: 16 Data Rate (clock-cycles/sample): 8 WCFR Window Length: 384

Number of Iterations: 1 Datawidth (bits): 18

Number of Cancellation Pulse Generators

Iteration 1
8

Max Cancellation Pulse Length: 511

Max Peak Detect Window: 3 [3 - 197]

TUSER Forwarding Setup

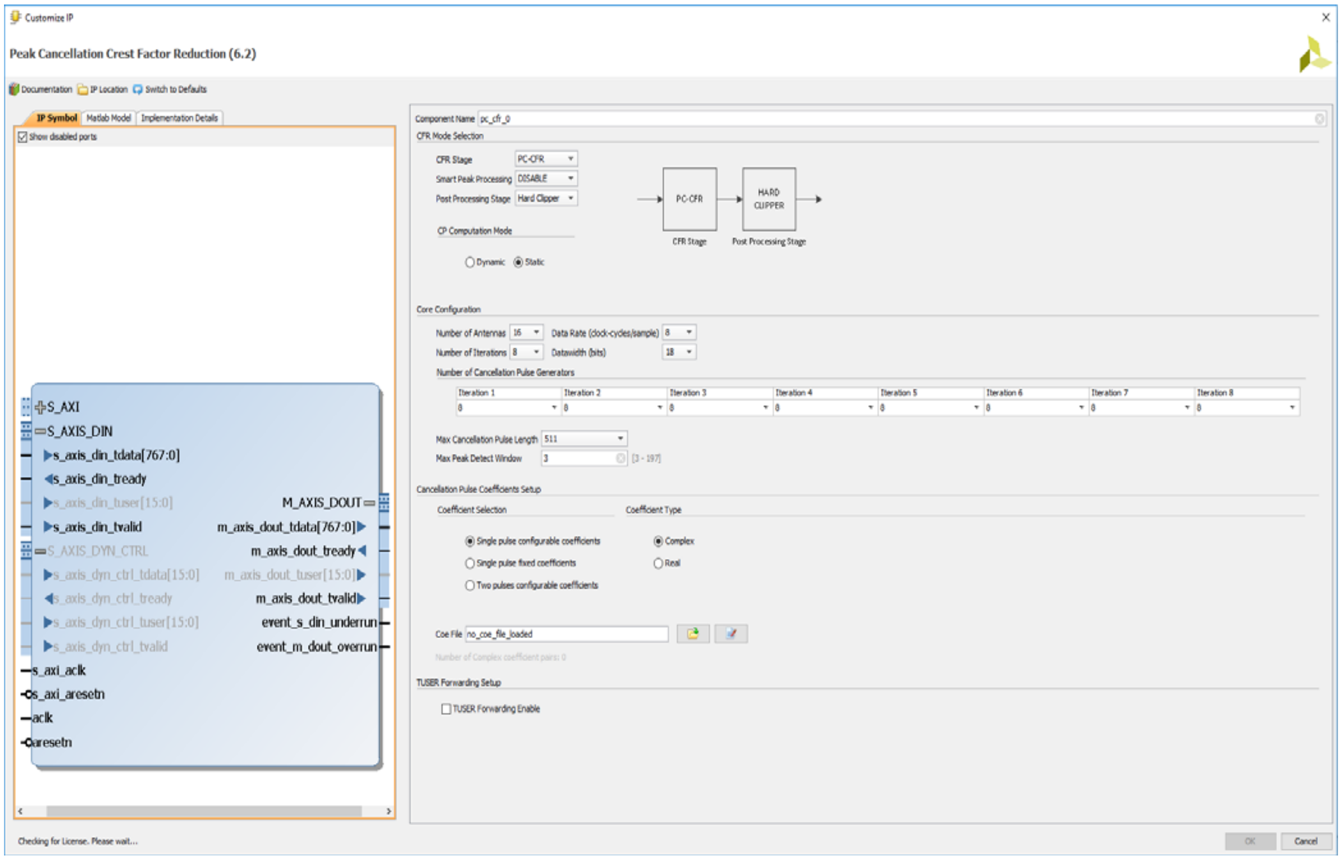
☐ TUSER Forwarding Enable

Checking for License. Please wait...

OK Cancel

Figure 4-3: **Core Interface in Dynamic Mode**

Figure 4-4 shows the core interface with smart peak processing disabled and hard clipper as the post-processing stage.



Peak Cancellation Crest Factor Reduction (6.2)

[✓] Show disabled ports

Component Name: pc_cfr_0

CFR Mode Selection

- CFR Stage: PC-CFR
- Smart Peak Processing: DISABLE
- Post Processing Stage: Hard Clipper

CP Computation Mode: ( ) Dynamic [✓ Static]

Core Configuration

- Number of Antennas: 16
- Data Rate (clock-cycles/sample): 8
- Number of Iterations: 8
- Datawidth (bits): 18

Number of Cancellation Pulse Generators

Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8
8	8	8	8	8	8	8	8

- Max Cancellation Pulse Length: 511
- Max Peak Detect Window: 3 [3 - 197]

Cancellation Pulse Coefficients Setup

Coefficient Selection: [✓ Single pulse configurable coefficients] ( ) Single pulse fixed coefficients ( ) Two pulses configurable coefficients

Coefficient Type: [✓ Complex] ( ) Real

Coe File: no_coe_file_loaded

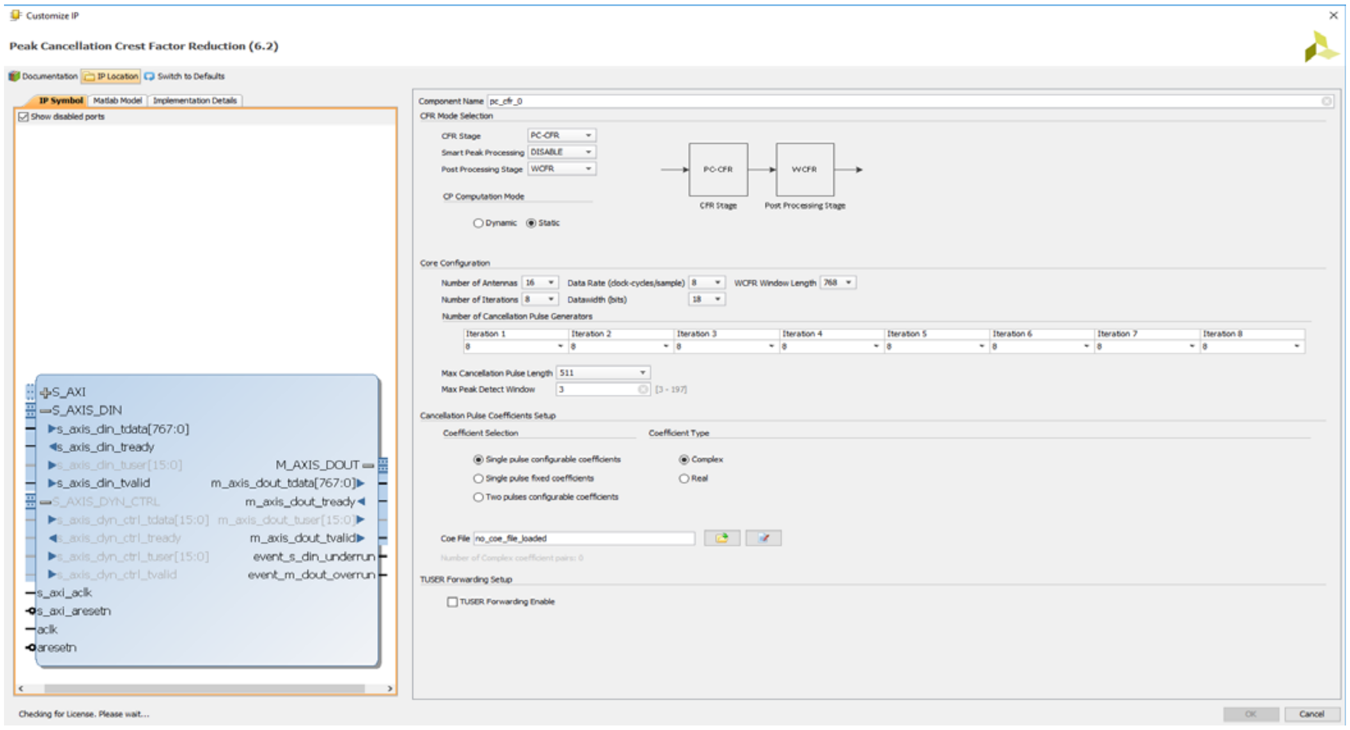
Number of Complex coefficient pairs: 0

TUSER Forwarding Setup

[ ] TUSER Forwarding Enable

Figure 4-4: **Core Interface with Smart Peak Processing Disabled**

Figure 4-5 shows the core interface with smart peak processing disabled and WCFR as the post-processing stage.



Peak Cancellation Crest Factor Reduction (6.2)

[✓] Show disabled ports

Component Name: pc_cfr_0

CFR Mode Selection

- CFR Stage: PC-CFR
- Smart Peak Processing: DISABLE
- Post Processing Stage: WCFR

CP Computation Mode: Dynamic [✓ Static]

Core Configuration

- Number of Antennas: 16
- Data Rate (clock-cycles/sample): 8
- WCFR Window Length: 768
- Number of Iterations: 8
- Datawidth (bits): 18

Number of Cancellation Pulse Generators

Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8
8	8	8	8	8	8	8	8

- Max Cancellation Pulse Length: 511
- Max Peak Detect Window: 3 [3 - 197]

Cancellation Pulse Coefficients Setup

Coefficient Selection: [✓ Single pulse configurable coefficients] Single pulse fixed coefficients / Two pulses configurable coefficients

Coefficient Type: [✓ Complex] Real

Coe File: no_coe_file_loaded

Number of Complex coefficient pairs: 0

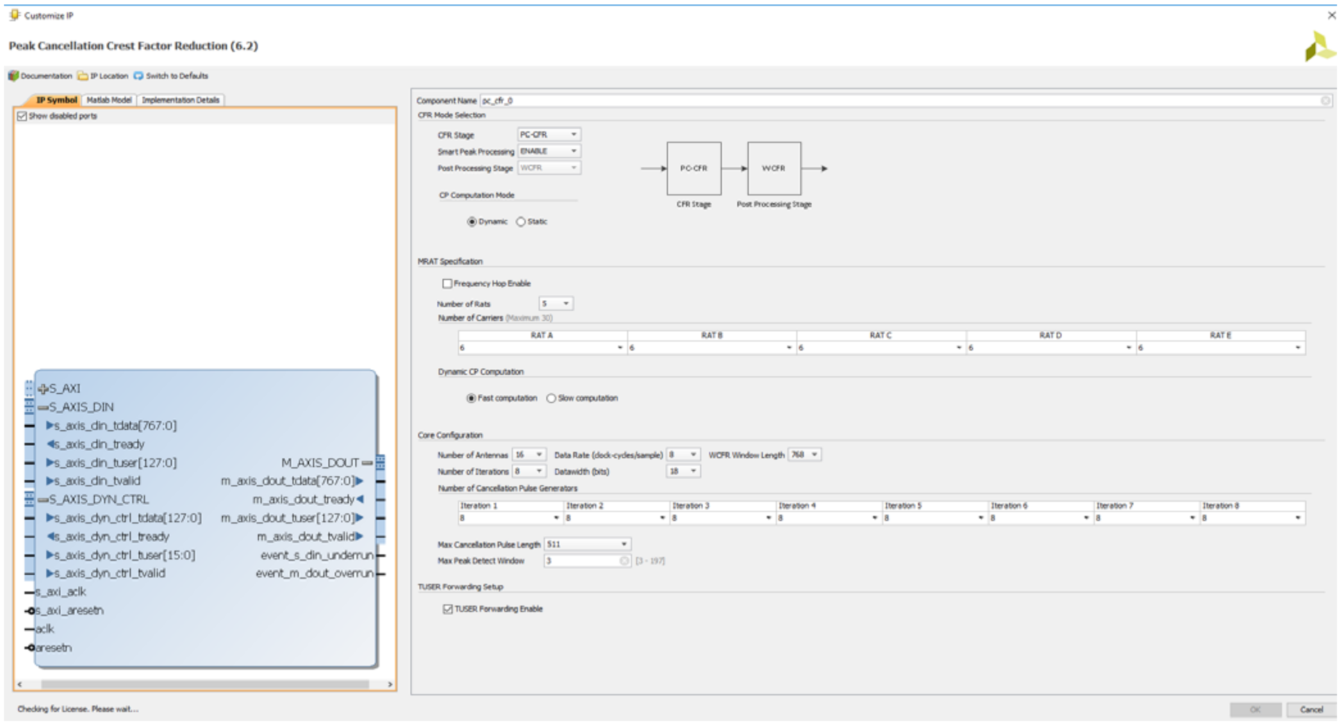
TUSER Forwarding Setup

[ ] TUSER Forwarding Enable

Checking for License. Please wait...

Figure 4-5: **Core Interface with Smart Peak Processing Disabled and WCFR in Post-Processing Stage**

Figure 4-6 shows the core interface with TUSER forwarding enabled



Peak Cancellation Crest Factor Reduction (6.2)

[✓] Show disabled ports

Component Name: pc_cfr_0

CFR Mode Selection

- CFR Stage: PC-CFR
- Smart Peak Processing: ENABLE
- Post Processing Stage: WCFR

CP Computation Mode: [✓ Dynamic] Static

MRAT Specification

[ ] Frequency Hop Enable

Number of Rats: 5

Number of Carriers (Maximum 30)

RAT A	RAT B	RAT C	RAT D	RAT E
6	6	6	6	6

Dynamic CP Computation: [✓ Fast computation] Slow computation

Core Configuration

- Number of Antennas: 16
- Data Rate (clock-cycles/sample): 8
- WCFR Window Length: 768
- Number of Iterations: 8
- Datawidth (bits): 18

Number of Cancellation Pulse Generators

Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8
8	8	8	8	8	8	8	8

- Max Cancellation Pulse Length: 511
- Max Peak Detect Window: 3 [3 - 197]

TUSER Forwarding Setup

[✓] TUSER Forwarding Enable

Checking for License. Please wait...

Figure 4-6: **Core Interface with TUSER Forwarding Enabled**

Component Name

The name of the core component to be instantiated. The name must begin with a letter and be composed of the following characters: a to z, A to Z, 0 to 9, and '_'.

Configuration Settings

CFR Stage

This setting selects between PC-CFR, WCFR, or Hard Clipper.

Smart Peak Processing

This setting selects between legacy PC-CFR mode or Smart Peak Processing based PC-CFR mode.

Post Processing Stage

This can either be Hard Clipper or WCFR or NONE. Note that if Smart Peak Processing is enabled then this stage is greyed out and WCFR is chosen automatically. This option is disabled when WCFR is chosen in the CFR stage.

CP Computation Mode

This setting specifies whether the PC-CFR runs in Static or in Dynamic mode. In Static mode, CFR uses the programmable coefficients from CP memories. In Dynamic mode the CFR estimates the CP coefficients from the given RAT input information. This mode should be used only if the power or frequency of the input is varying dynamically. This option is present only when PC-CFR is chosen as the CFR stage.

Number of Antennas

This setting specifies the number of antennas (transmit paths) required. The options available are 1, 2, 4, 8 or 16. The data interface port width (`C_AXIS_TDATA_WIDTH`) scales with the number of antennas selected. The extra sideband signal width (`C_AXIS_TUSER_WIDTH`) scales with the number of antennas selected in Dynamic mode. This parameter is automatically set in IP integrator but can also be overridden.

Number of Iterations

This setting specifies the number of iterations required. The available range is 1 to 8. The latency of the core increases with the number of iterations selected as given by Equation 3-4. This option is disabled when WCFR is chosen in the CFR stage.

Data Rate (clock-cycles/sample)

This setting specifies the data rate in clocks per sample where the core expects a new sample every n clock cycles based on the value specified. This enables the core to maximize resource savings based on the level of over-clocking. The available range is 1, 2, 3, 4 and 8.

Datawidth

The data width must be 16 or 18 bits. The width specified is used at the external interfaces, but all internal values are represented by 18 bits. If a value less than 18 bits is specified, zero-padding on the LSB is performed at the input interface and symmetric rounding is performed at the output interface. The threshold value is scaled up to 18 bits internally. This parameter is automatically set in IP integrator but can also be overridden.

WCFR Window Length

This field is present when CFR Stage or Post Processing Stage has WCFR. This parameter specifies the length of the window filter in WCFR. It can have one of three distinct values: 192, 384, or 768. All the parameters given below are disabled when WCFR is chosen in the CFR stage.

Number of Cancellation Pulse Generators

This setting specifies the number of cancellation pulse generators available per iteration. The number specified is confined to an integer multiple of the data rate to maximize efficiency. The cancellation pulse generators can all be allocated independently and operate in parallel.

Max Cancellation Pulse Length

This setting specifies the maximum length of the cancellation pulse used. This is used to optimize block RAM utilization for storing the cancellation pulse coefficients. The available options are 511, 1023, and 2047.

Real or Complex Mode

This setting specifies whether the input pulse coefficients are real or complex. This helps in more optimized resource utilization. By default, the input pulse coefficients are assumed to be complex.

Frequency Hopping Enable

This check box used to select whether frequency hopping is supported by the core. Select Frequency Hopping Enable when the incoming signal is FH-GSM.

Number of RATs

This setting specifies the number of RAT to be set for PC-CFR in Dynamic mode. The PC-CFR supports a maximum of five RATs in Dynamic Power mode (frequency hopping disabled) and one RAT with frequency hopping enabled. When number of RATs selected are less than or equal to 3, then each RAT supports 1 to 8 carriers. When number of RATs are more than 3, then each RAT supports 1 to 6 carriers.

Dynamic CP Computation

This setting specifies whether to use a serial or parallel CP computation engine in Dynamic mode. The amount of time the PC-CFR takes to compute the CP coefficients in the serial computation engine is more than the time it takes in parallel computation. This option is set to Fast by default in frequency hopping mode. When frequency hopping is disabled, FPGA resources required for slow computation are less when compared to fast computation.

Cancellation Pulse Coefficients Setup

This setting is visible only in Static mode. In Dynamic mode, two pulse configurable coefficients are used by the core internally.

Selecting configurable coefficients enables the pulse coefficients to be written and updated dynamically by a processor or other source. Selecting two pulses instead of a single pulse increases the coefficient memory required, but enables you to update the pulse that is not in use and then dynamically switch to it after the update is complete.

When selecting fixed coefficients, a COE file is required. The coefficient values in the COE file must all be specified in radix 10 format as interleaved IQ values, that is, sample 1(I), sample 1(Q), sample 2(I), sample 2(Q)... For multiple antenna scenarios, the coefficients of the first antenna must come first, followed by the coefficients of the second antenna, and so on. The number of coefficients per antenna (assumed to be the same across all antennas) is inferred from the COE file and is reported in the core configuration window. The number of coefficients per antenna must always be an odd value in the range 1 to Max Cancellation Pulse Length; each I (or Q) value must be within the range represented by a 16-bit signed number, that is, -32768 to 32767 inclusive.

An example of a COE file:

```
radix=10;
coefdata= 1, -1,
-1, 1,
1, -1;
```

The IDE performs error-checking on all input parameters.

Implementation Details

In the Implementation Details tab the latency information is automatically updated for the current configuration.

Latency

This field shows the latency of the core in number of samples. This is the latency from the first input sample to the first output sample.

User Parameters

Table 4-1 shows the relationship between the fields in the Vivado IDE and the User Parameters (which can be viewed in the Tcl Console).

Table 4-1: **Vivado IDE Parameter to User Parameter Relationship**

Vivado IDE Parameter /Value[1]	User Parameter / Value[1]	Default Value
CFR Mode Selection		
CFR Stage	CFR_METHOD	0
PC-CFR	0	
WCFR	2	
Standalone Hard Clipper	3	
Smart Peak Processing	Smart_Peak_Processing	4
ENABLE	4	
DISABLE	1	
Post Processing Stage	Post_Processing_Stage	0
WCFR	0	0
HARD CLIPPER	1	
NONE	2	
CP Computation Mode	Mode	0
Static	0	
Dynamic	1	
Core Configuration		
Number of Antennas	C_NUM_ANTENNAS	1
Data Rate	Data_Rate	4
Number of Iterations	C_NUM_ITERATIONS	4
WCFR Window Length	WCFR_Window_Length	384
Iteration 1	Cancellation_Pulse_Generator_0	4
Iteration 2	Cancellation_Pulse_Generator_1	4
Iteration 3	Cancellation_Pulse_Generator_2	4
Iteration 4	Cancellation_Pulse_Generator_3	4
Iteration 5	Cancellation_Pulse_Generator_4	4
Iteration 6	Cancellation_Pulse_Generator_5	4
Iteration 7	Cancellation_Pulse_Generator_6	4

Table 4-1: **Vivado IDE Parameter to User Parameter Relationship *(Cont'd)***

Vivado IDE Parameter /Value[(1)]	User Parameter / Value[(1)]	Default Value
Iteration 8	Cancellation_Pulse_Generator_7	4
Max Cancellation Pulse Length	Max_Cancellation_Pulse_Length	511
Max Peak Detect Window	Max_Peak_Detect_Window	3
Frequency Hop Enable	Frequency_Hop_Enable:	FALSE
Unchecked	FALSE	
Checked	TRUE	
Dynamic CP Computation	Dynamic_CP_Computation	0
Fast Computation	0	
Slow Computation	1	
Number of RATs	Number_of_Rats	1
RAT A	Number_of_Carriers_RATA	1
RAT B	Number_of_Carriers_RATB	1
RAT C	Number_of_Carriers_RATC	1
RAT D	Number_of_Carriers_RATD	1
RAT E	Number_of_Carriers_RATE	1
Cancellation Pulse Setup		
Coefficient Selection	Coefficient_Selection	
Single pulse configurable coefficients	Single_pulse_configurable_coefficients	Single_pulse_configurable_coefficients
Single pulse fixed coefficients	Single_pulse_fixed_coefficients	
Two pulses configurable coefficients	Two_pulses_configurable_coefficients	
Coefficient Type	C_REAL_COMPLEX_CP	1
Complex	1	
Real	0	
Coe File	Coe_File	no_coe_file_loaded
TUSER Forwarding Setup		
TUSER Forwarding Enable		0
Enable	1	
Disable	0	

Notes:

1. Parameter values are listed in the table where the Vivado IDE parameter value differs from the user parameter value. Such values are shown in this table as indented below the associated parameter.

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

Constraining the Core

This section contains information about constraining the core in the Vivado Design Suite.

Generate Example Constraints File

To generate the example constraints file, right-click the core name in Project Manager, Sources. Select **Generate Output Products...** In the **Manage Output Products** dialog box select **Generate** in the **Miscellaneous** drop-down selection.

<project_directory>/<component_name> Directory

Table 4-2: **Component Name Directory**

Name	Description
`<project_directory>/<component_name>`	
`<component_name>_impl.xdc`	Constraints file for use at the implementation stage (ensures all BRAMs are in WRITE_FIRST mode) for better timing closure.
`<component_name>_clocks.xdc`	Example file showing the false path settings across the AXI4-Lite and AXI4-Stream clocks.
`<component_name>_ooc.xdc`	This file contains clock period constraints for AXI4-Lite and AXI4-Stream clocks to be used in OOC mode.

Required Constraints

This section is not applicable for this IP core.

Device, Package, and Speed Grade Selections

This section is not applicable for this IP core.

Clock Frequencies

This section is not applicable for this IP core.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP core.

Simulation

For comprehensive information about Vivado simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 8].

IMPORTANT: *For cores targeting 7 series or Zynq-7000 devices, UNIFAST libraries are not supported. Xilinx IP is tested and qualified with UNISIM libraries only.*

Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

Test Bench

This chapter contains information about the test bench provided in the Vivado® Design Suite environment.

Demonstration Test Bench

When the core is generated using the Vivado IDE, a demonstration test bench (and example test vectors file) can be created by following these steps:

1. Right-click on the core name in **Project Manager, Sources**.
2. Select **Generate Output Products...**
3. In the **Manage Output Products** dialog box, select **Generate** in the Test Bench drop-down menu.

This is a simple VHDL test bench that exercises the core. The demonstration test bench source code is one VHDL file: `demo_tb/tb_<component_name>.vhd` in the Vivado output directory. The source code is comprehensively commented.

Using the Demonstration Test Bench

The demonstration test bench instantiates the generated PC-CFR core. If the IDE project options are set to generate a structural model, a VHDL or Verilog netlist named `<component_name>.vhd` or `<component_name>.v` is generated. This file can be generated after synthesizing the core with the following Tcl command:

```
write_vhdl -mode funcsim <component_name>.vhd:
```

Compile the netlist and the demonstration test bench into the work library (see your simulator documentation for more information on how to do this). Then simulate the demonstration test bench. View the test bench signals in your simulators waveform viewer to see the operations of the test bench.

Demonstration Test Bench in Detail

The demonstration test bench performs the following tasks:

- Instantiates the core
- Generates the clock and reset signals
- Drives the core input signals to demonstrate core features
    - Control write/read using AXI4-Lite interface
    - Data I/O using AXI4-Stream interface
- Writes the output data into a file and exits simulation.

More information on the demonstration test bench is available in the `tb_readme.txt` file that is generated as part of the test bench.

Upgrading

This appendix contains information about migrating a design from ISE® to the Vivado® Design Suite, and for upgrading to a more recent version of the IP core. For customers upgrading in the Vivado Design Suite, important details (where applicable) about any port changes and other impact to user logic are included.

Migrating to the Vivado Design Suite

For information about migrating to the Vivado Design Suite, *see the ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 9].

Upgrading in the Vivado Design Suite

This section provides information about any changes to the user logic or port designations that take place when you upgrade to a more current version of this IP core in the Vivado Design Suite.

Register Space Changes

Register Space is extended to 512K to support 16 antennas. The Ant Sel bits in 32-bit field are changed from 17-15 to 18-15. As a result, the Static Coefficients/Base Pulse Select address bits are changed from 20-18 to 21-19. See Table 2-2 for more details.

Other Changes

A C model is no longer available for this core. MATLAB® software simulator can be downloaded from the PC-CFR evaluation lounge.

Instructions for Minimum Change Migration

Migrating to v6.2

You can migrate PC-CFR v6.1 to v6.2 while retaining the functionality of v6.1. PC CFR v6.2 supports 16 antennas as a new feature, forcing changes to the Register Space. Address bits for Static Coefficients/Base Pulse Select and Ant Sel bits are impacted. Therefore, you need to be careful while migrating your software to work with PC CFR v6.2. See Register Space section and Table 2-2 for more details.

Debugging

This appendix includes details about resources available on the Xilinx Support website and debugging tools.

TIP: *If the IP generation halts with an error, there may be a license issue. See License Checkers in Chapter 1 for more details.*

Finding Help on Xilinx.com

To help in the design and debug process when using the PC-CFR, the Xilinx Support web page contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

Documentation

This Product Guide is the main document associated with the PC-CFR. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page or by using the Xilinx Documentation Navigator.

You can download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core can be located by using the Search Support box on the main Xilinx support web page. To maximize your search results, use proper keywords such as

- the product name
- tool message(s)

- summary of the issue encountered.

A filter search is available after results are returned to further target the results.

Master Answer Record for the PC-CFR

AR: 54486

Technical Support

Xilinx provides technical support in the Xilinx Support web page for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support, navigate to the Xilinx Support web page.

Debug Tools

There are tools available to address PC-CFR core design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx.

The Vivado logic analyzer is used with the logic debug IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 10].

Hardware Debug

Hardware issues can range from link bring-up to problems seen after hours of testing. This section provides debug steps for common issues. The Vivado Lab Edition is a valuable resource to use in hardware debug.

General Checks

Ensure that all the timing constraints for the core were properly incorporated from the example design and that all constraints were met during implementation.

- Does it work in post-place and route timing simulation? If problems are seen in hardware but not in timing simulation, this could indicate a PCB issue. Ensure that all clock sources are active and clean.
- If your outputs go to 0; check your licensing.

Interface Debug

AXI4-Lite Interfaces

Read from a register that does not have all 0s as a default to verify that the interface is functional. See Figure 3-9 for a read timing diagram. Output `s_axi_arready` asserts when the read address is valid and output `s_axi_rvalid` asserts when the read data/response is valid. If the interface is unresponsive ensure that the following conditions are met.

- The `S_AXI_ACLK` and `ACLK` inputs are connected and toggling.
- The interface is not being held in reset, and `S_AXI_ARESETN` is an active-Low reset.
- The interface is enabled, and `s_axi_aclken` is active-High (if used).
- Ensure that the main core clocks are toggling and that the enables are also asserted.
- Has a simulation been run? Verify in simulation and/or a Vivado Lab Edition capture that the waveform is correct for accessing the AXI4-Lite interface.

AXI4-Stream Interfaces

If data is not being transmitted or received, check the following conditions.

- If transmit `<interface_name>_tready` is stuck Low following the `<interface_name>_tvalid` input being asserted, the core cannot send data.

- If the receive `<interface_name>_tvalid` is stuck Low, the core is not receiving data.
- Check that the ACLK inputs are connected and toggling.
- Check that the AXI4-Stream waveforms are being followed (see Figure 3-15 and Figures 3-8 to 3-11).
- Check core configuration.

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado® IDE, select **Help > Documentation and Tutorials**.
- On Windows, select **Start > All Programs > Xilinx Design Tools > DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on Documentation Navigator, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this product guide:

1. *High Density WiMAX Digital Front End Reference Design* (XAPP966), registration required
2. *Peak Cancellation Crest Factor Reduction in a Multi-Standard Transmit System* (XAPP1174), registration required
3. *AMBA® AXI4-Stream Protocol Specification* (ARM IHI 0051A)
4. *Xilinx Vivado AXI Reference Guide* (UG1037)
5. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
6. *Vivado Design Suite User Guide: Designing with IP* (UG896)
7. *Vivado Design Suite User Guide: Getting Started* (UG910)
8. *Vivado Design Suite User Guide - Logic Simulation* (UG900)
9. *ISE® to Vivado Design Suite Migration Guide* (UG911)
10. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
11. 3GPP Technical Specification GSM/EDGE, TS 45.005 V7.12.0 "Radio transmission and reception (Release 7)"

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/06/2018	6.2	• Added 16 antenna support • Complex multiplier used in CPGs is changed from 4-DSP 48s to 3-DSP 48s based implementation
04/04/2018	6.1	• Updated the IP core latency numbers • Added Dynamic CP Computation for Multi-RAT Cases with Power Dynamics section
12/20/2017	6.1	Updated supported information on number of RATs and carriers
10/05/2016	6.1	• Added support to work as stand-alone hard clipper • Optional feature to operate WCFR without smart peak processing • Added TUSER forwarding feature. TUSER is delay matched with data-path. • Support for additional RATs (RATD and RATE) in dynamic CP computation mode
11/18/2015	6.0	Added support for UltraScale+ families
04/01/2015	6.0	• Production release of PC-CFR v6.0 core
10/01/2014	6.0	• Version 6.0 is Early Access, lounge delivered to customers • Added UltraScale architecture support • New smart peak processing added to enable inputs at 1.2 x iBW • New post processing stage to support optional hard clipper or WCFR • Support for WCFR standalone mode (without PC-CFR) • Support of data-rate 8 added, data-width of 16 and 18 bits only • New AXI4-Stream control channel for dynamic mode • New CFR statistics registers and version register added • Read back support for all AXI4-Lite registers • Read back support for base CPs in dynamic mode • Drives zero upon hardware time out • Support for C model has been removed, MATLAB(R) model can be downloaded from the secure lounge
12/18/2013	5.0	• Revision number advanced to 5.0 to align with core version number. • Added dynamic support (dynamic power variation and frequency hopping) • Hard clipper support • Change in tool settings for characterization • New TUSER port added in Dynamic mode

Date	Version	Revision
03/20/2013	2.0	PC-CFR v4.0 is available only with Vivado; other changes are • Improved f_{max}, core can be clocked at 491.52 MHz for -2 devices • Core latency has changed
12/18/2012	1.0	This Product Guide is derived from DS846 and UG808. The changes for 14.4/2012.4 are as follows: • Real/Complex CP selection added • Number of CPGs/iteration increased to 12 • Core is fully supported in Vivado 2012.4 • Peak detect window is used in place of allocator spacing

Please Read: Important Legal Notices