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.

In a modern transmit chain, CFR is also often incorporated with Digital Predistortion (DPD), which acts to linearize the PA, allowing operation at maximum efficiency with spectral compliance. CFR is helpful to DPD because it levels the signal peaks, making accurate correction estimation easier [Ref 1].

The Xilinx PC-CFR core is an efficient, flexible and easy-to-use implementation that supports Virtex®-5, Virtex-6, and Spartan®-6 FPGA families. It is configurable both in function, to support all major cellular wireless air interfaces, and in usage, to support a variety of clocking and resource requirements.

Features and General Description

The Peak Cancellation-CFR core processes control and data through standardized interfaces that allow immediate logic-free connection to other Xilinx IP components and, through simple rules, to any general environment. The data flow is unidirectional with no rate or bit-width change. The control interface consists of constant parallel register inputs and an optional memory-mapped interface to a pulse coefficient RAM (described later). A typical CFR application consists of more than one instance of the core chained in series and in parallel (where multiple transmit paths are to be serviced). The core is configured for a particular application through the control interfaces. In particular, the contents of the pulse coefficient RAM are related to the spectrum of the signal being transmitted. Later sections of this data sheet discuss the general principles behind producing these coefficients, and specific detail is given for the WCDMA, CDMA2000, WiMAX, TD-SCDMA, GSM and E-UTRAN (LTE) air interfaces. Mixed-mode signal operation is also supported. Pulse coefficients may be pre-configured at generation time through a .coe file or configured in operation via the control interface. There is also provision for a shadow bank of coefficients to be loaded, then and activated with an immediate signal, to cater to applications where fast dynamic switching is required. Functions that can be run with MATLAB® are supplied for simulation and design of the cancellation pulse.

The core may be configured for clock-to-sample ratios between 1 and 8, and for algorithmic complexity, allowing FPGA fabric resources to be minimized for a given application. The algorithmic complexity is the number of hardware resources available to cancel the signal peaks. These are called Cancellation Pulse Generators.

Algorithm Description

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

The correction signal is a spectrally compliant signal that matches the signal peaks. For example, in Noise-shaping CFR [Ref 3] 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 cor-
rection signal is a sum of individual cancellation pulses. The pulses are applied by searching for peaks in the signal. Each pulse is the impulse response of a filter that is designed to match the spectral content of the signal. Figures 2, 3, and 4 depict the sequence of events. Figure 2 shows the detection of the peak and the identification of the excess amplitude over threshold, \((A - \text{Ath})\). Figure 3 shows the cancellation pulse. This pulse is the appropriate unit magnitude impulse response scaled by \(\text{CP\_val} = (A - \text{Ath}) \cdot e^{\vartheta}\) where \(\vartheta\) is the phase.

That is to say, 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 4, while preserving the signal phase. For asymmetric spectra, the cancellation pulse itself is complex.

![Figure 2: Peak Detection](image)

![Figure 3: The Cancellation Pulse](image)
In hardware, the cancellation pulse is generated by a cancellation pulse generator (CPG) with function as indicated in Figure 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.

The resource complexity of the PC-CFR 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 cancelled in one pass, so an application typically has multiple (identical) iterations.

A detailed structure for the PC-CFR core implementation is shown in Figure 6.
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 parameter at a constant register input. The Peak Scale block forms the complex value for the cancellation pulse as described previously. The Allocator block takes care of assigning CPGs to peaks. It has an Allocator Spacing as a parameter/register input. This restricts the frequency of peak assignment and is sometimes used to improve EVM performance for certain signals, as discussed in the "Applications" section. 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.
CORE Generator™ Software Parameters

A screen shot of the core configuration window is shown in Figure 7.

**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**

**Data Rate (clock-cycles/sample)**

This setting specifies the data-rate in clocks per sample where the core will expect 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.

**Number of Cancellation Pulse Generators**

This setting specifies the number of cancellation pulse generators available. 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.

**Datwidth**

The datwidth must be in the range 11 to 18 bits. The width specified is used at the external interfaces, but all internal values are represented by 16 bits. If a value other than 16 bits is specified, then internal symmetric rounding or padding takes place at both the input and output interfaces. It is expected that..
16 bits will be the most common usage, and the internal design of the core is optimized for 16 bits. The other datawidth selections are included for compatibility, and the resource usage will not change by very much when the datawidth is changed.

Cancellation Pulse Coefficients Setup

Selecting configurable coefficients results in a memory map interface being provided on the core. This 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 the user 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 number of coefficient or filter taps is inferred from the number of IQ pairs found in the COE file and will be reported in the cores configuration window. The 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) and so on. The number of IQ pairs must always be an odd value in the range 1 to 511; each 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;
```

Input/Output Ports

Figure 8 displays the signal names; Table 1 defines these signals.
<table>
<thead>
<tr>
<th>Port</th>
<th>Direction</th>
<th>Port Width (bits)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aclk</td>
<td>Input</td>
<td>1</td>
<td>Main core clock. Synchronous operations associated with all interfaces apart from \texttt{sreg}_occur on the rising edge of this clock signal.</td>
</tr>
<tr>
<td>areset</td>
<td>Input</td>
<td>1</td>
<td>Active high synchronous reset. Disables any active cancellation pulse generators and resets core interface to power up state.</td>
</tr>
<tr>
<td>mstat_data</td>
<td>Output</td>
<td>32</td>
<td>Status output. Bit 0 represents the under-run condition on the \texttt{sdata}_interface, and bit 1 represents the over-run condition on the \texttt{mdata}_interface. All other bits are unused.</td>
</tr>
<tr>
<td>sparam_threshold</td>
<td>Input</td>
<td>16</td>
<td>Input to provide the threshold level over which valid peaks can be identified.</td>
</tr>
<tr>
<td>sparam_alloc_spacing</td>
<td>Input</td>
<td>10</td>
<td>Input to provide the minimum number of samples required between valid peaks in order for a cancellation pulse generator to be allocated. The core has an inherent minimum spacing of 2, so the actual value will be the value specified plus 2.</td>
</tr>
<tr>
<td>sparam_filter_n_taps</td>
<td>Input</td>
<td>9</td>
<td>Input to provide the number of filter taps. This controls the size of the cancellation pulse and therefore the number of coefficient values that are read from memory. Present only when configurable coefficients have been configured, as otherwise this value is inferred from the COE file.</td>
</tr>
<tr>
<td>sparam_filter_select</td>
<td>Input</td>
<td>1</td>
<td>Input to select which pulse is in use. Present only when 'Two pulses (selectable) configurable coefficients’ is selected.</td>
</tr>
<tr>
<td>sreg_aclk</td>
<td>Input</td>
<td>1</td>
<td>Clock for \texttt{sreg}_interface. Synchronous operations on the \texttt{sreg}_interface occur on the rising edge of this clock signal.</td>
</tr>
<tr>
<td>sreg_areset</td>
<td>Input</td>
<td>1</td>
<td>Active high synchronous reset. Resets \texttt{sreg}_interface.</td>
</tr>
<tr>
<td>sreg_awvalid</td>
<td>Input</td>
<td>1</td>
<td>Active high. When asserted, the core reads the address on the \texttt{sreg_awaddr} port, if \texttt{sreg_awready} is high.</td>
</tr>
<tr>
<td>sreg_awready</td>
<td>Output</td>
<td>1</td>
<td>Active high. Indicates that the core is ready for a valid address on the \texttt{sreg_awaddr} port.</td>
</tr>
<tr>
<td>sreg_awaddr</td>
<td>Input</td>
<td>32</td>
<td>Address to write next filter coefficient data. Address read when \texttt{sreg_awvalid} and \texttt{sreg_awready} asserted high on a rising clock edge. With 'Single pulse configurable coefficients' the bottom 9 bits are used, and for 'Two pulses (selectable) configurable coefficients' the bottom 10 bits are used. All others are ignored.</td>
</tr>
<tr>
<td>sreg_wvalid</td>
<td>Input</td>
<td>1</td>
<td>Active high. When asserted, the core reads the data on the \texttt{sreg_wdata} port, if \texttt{sreg_wready} is high.</td>
</tr>
<tr>
<td>sreg_wready</td>
<td>Output</td>
<td>1</td>
<td>Active high. Indicates that the core is ready for valid data on the \texttt{sreg_wdata} port.</td>
</tr>
<tr>
<td>sreg_wdata</td>
<td>Input</td>
<td>32</td>
<td>Coefficient data to write to memory. Data read when \texttt{sreg_wvalid} and \texttt{sreg_wready} asserted high on a rising clock edge. The lower 16 bits [15:0] correspond to the I component, and the upper 16 bits [31:16] correspond to the Q component.</td>
</tr>
</tbody>
</table>
Table 1: Input and Output Port Definitions (Cont’d)

<table>
<thead>
<tr>
<th>Port</th>
<th>Direction</th>
<th>Port Width (bits)</th>
<th>Description                                                                 -------------</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdata_valid</td>
<td>Input</td>
<td>1</td>
<td>Active high. When asserted, the core reads the data on the sdata_i and sdata_q ports, if sdata_ready is high.</td>
</tr>
<tr>
<td>sdata_ready</td>
<td>Output</td>
<td>1</td>
<td>Active high. Indicates that the core is ready for valid data on the sdata_i and sdata_q ports.</td>
</tr>
<tr>
<td>sdata_i</td>
<td>Input</td>
<td>11-18</td>
<td>Slave interface for I component of data input to the core. Port width equal to DATAWIDTH parameter specified at configuration.</td>
</tr>
<tr>
<td>sdata_q</td>
<td>Input</td>
<td>11-18</td>
<td>Slave interface for Q component of data input to the core. Port width equal to DATAWIDTH parameter specified at configuration.</td>
</tr>
<tr>
<td>mdata_valid</td>
<td>Output</td>
<td>1</td>
<td>Active high. When asserted, indicates that the current output data on the mdata_i and mdata_q ports is valid.</td>
</tr>
<tr>
<td>mdata_ready</td>
<td>Input</td>
<td>1</td>
<td>Active high. When asserted, indicates that any valid data on the mdata_i and mdata_q ports has been read.</td>
</tr>
<tr>
<td>mdata_i</td>
<td>Output</td>
<td>11-18</td>
<td>Master interface for I component of data output from the core. Port width equal to DATAWIDTH parameter specified at configuration.</td>
</tr>
<tr>
<td>mdata_q</td>
<td>Output</td>
<td>11-18</td>
<td>Master interface for Q component of data output from the core. Port width equal to DATAWIDTH parameter specified at configuration.</td>
</tr>
</tbody>
</table>

**General Operation**

**Memory-Mapped Coefficient Interface**

This interface is present only where configurable coefficients are in use. The address space available is 512 for a single-pulse setup and 1024 for the two-pulse setup. The coefficient for the first pulse should be written starting at address location 0, and the coefficient for the second pulse, if present, should be written starting from address location 512. Either both address and data can be written simultaneously, or one can be written before the other. Figure 9 shows the timing diagram for these operations.

![Figure 9: Memory Map Timing Diagram](image)

The timing diagram firstly shows the case where address and data are written simultaneously, which can occur on back-to-back clock cycles and therefore utilizes the full bandwidth of the interface. Secondly it shows an address write before the corresponding data, thirdly a data write before the corre-
sponding address, and lastly a write terminated by reset. Where either the address or data is written in advance of the other, the corresponding ready will be deasserted by the core until the other write is complete, as there is only the ability to buffer one address or data within the core itself. 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. When the interface is idle or the address and data have both successfully been written on a previous cycle, the reset will have no effect.

**Parameter Interface**

The ports that make up this interface form direct connections to the core. These can either be tied off to a fixed value or connected to an external register to allow them to be changed dynamically. No attempt has been made to balance the pipelines of these values to the different parts of the circuit where they are applied. So if any of the parameters prefixed with `sparam_` are changed during normal core operation without applying a reset, then there will be transient behavior at the output for up to 1024 samples. See Table 1 for the definition of these ports.

**Data Interface**

Figure 10 shows the timing diagram for both the slave interface for data input and the master interface for data output with a Data-Rate of three clock cycles per sample.

After power-up or reset, the core is idle until `SDATA_VALID` has been asserted for the first time; `SDATA_READY` is held high and `MDATA_VALID` is held low until this occurs. 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 per sample. Figure 10 shows data being read from the slave interface every three cycles on and after the first assertion of `SDATA_VALID`. The core indicates that it is ready to read data by setting `SDATA_READY` high and at the same time expects `SDATA_VALID` to be asserted, indicating there is valid data on the ports `SDATA_I` and `SDATA_Q`. Figure 10 shows various acceptable combinations of `SDATA_VALID` where it is asserted at the same time as `SDATA_READY`, asserted in advance, or held asserted between `SDATA_READY` assertions. If `SDATA_VALID` is not asserted when `SDATA_READY` is held high, then the under-run condition occurs because valid data was not available when the core expected it. The core can read a new input data sample only when both `SDATA_VALID` and `SDATA_READY` are held high. As the core is free-running, under-run will cause the preceding data sample to be input for a second time internally. Under-run is flagged by setting bit 0 on the `MSTAT_DATA` port; this bit will remain set till the next reset operation.
After the core is synchronized by the slave interface, it periodically asserts MDATA_VALID to indicate valid output samples on the ports MDATA_I and MDATA_Q. The Data-Rate specified determines how many cycles are available to read this data. The core expects the data to be read by asserting MDATA_READY before the next output data sample is available. Figure 10 shows various acceptable combinations where the output data is read either immediately, a cycle after, or just prior to the next output data becoming available. Once MDATA_READY has been asserted to indicate the sample has been read, MDATA_VALID is deasserted unless the next output sample is available on the following clock cycle; in this case MDATA_VALID remains asserted while MDATA_I and MDATA_Q are updated with new output data. If an output sample is not read by asserting MDATA_READY before the next is available, then the over-run condition occurs. As the core is free-running, over-run causes the next output sample to be dropped internally, as the current one has not been read and will remain on the output ports MDATA_I and MDATA_Q. Over-run is flagged by setting bit 1 of the port MSTAT_DATA; this bit remains set until the next reset operation.

The design of the interface is such that the master interface of one instance can be directly connected to the slave interface of another instance running from the same clock and with the same number of clocks per sample. In a chain, MDATA_I and MDATA_Q of the first instance are connected to SDATA_I and SDATA_Q of the second instance, MDATA_VALID of the first instance is connected to SDATA_VALID of the second instance, and MDATA_READY of the first instance is connected to SDATA_READY of the second instance, and so on. This method can also be used to connect to other Xilinx cores that use the same protocol.

For general connectivity, if SDATA_VALID of an instance is tied high, then data can be streamed into the SDATA_I and SDATA_Q inputs at the appropriate rate. In this case, the output SDATA_READY is pulsed high one clock period in n to indicate data being read by the core and can therefore be used to enable an input register or FIFO.

Similarly, MDATA_I and MDATA_Q will stream output data continuously at the defined rate where MDATA_READY is tied high, and in this case MDATA_VALID is available to enable the data.

Reset Operation

A reset port, ARESET, is provided to return the interface to its power-up state where it is not synchronized and waiting for the first valid input data. ARESET 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 will not necessarily be zero data. To completely flush the core and ensure zero valued output data prior to the processed input data, ARESET should be held high for a period greater than 512 cycles. Alternatively, ARESET can be held high 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 512 cycles.

Dynamic Cancellation Pulse Switching

For applications where the spectrum of the signal is changing, the pulse coefficients can be adapted using the two-pulse setup. Normal dynamics in a multi-carrier configuration are unlikely to require this, but situations where carriers are added, taken away or moved, for example in frequency-hopping GSM, may.

The signal SPARAM_FILTER_SELECT, when set to zero, makes the pulse stored in locations starting at zero in the pulse coefficients RAM active. When SPARAM_FILTER_SELECT is set to one, the pulse
stored in locations starting at 512 becomes active. So for example if `SPARAM_FILTER_SELECT` is zero and the pulse needs to adapt, the new coefficients should be written into RAM locations starting at 512, and then `SPARAM_FILTER_SELECT` can be set to one to activate. The converse, toggling process can continue for subsequent changes.

The speed at which the pulses can be adapted depends on the controller at the data interface. With, for example, a 50MHz clock, new coefficients can in principle be written in 10 microseconds; however in practice the time taken to recalculate for the new frequency content will need to be taken into account. If a microcontroller is used, then other factors will come into play; however adaptation rates of less than one millisecond would seem reasonable to achieve.

When the active pulse is switched, there will be a transient. The duration is short and should not be a problem 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.

**Applications**

In the "Data Interface" section, it was shown how to configure and connect the PC-CFR instances and how to chain them together to form a multiple-iteration unit.

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

- The sample rate of operation
- The number of iterations
- The number of CPGs at each iteration
- The pulse shape coefficients
- The threshold value(s)
- The allocator spacing values

The following sections deal with each of these in turn and give guidelines. It is also 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**

**The Sample Rate of Operation**

The sampling rate \( (f_s) \) of CFR should be at least three times the signal bandwidth for single carrier and contiguous multi-carrier spectra. For non-adjacent carrier configuration, a higher ratio is required – at least five times for two non-adjacent carriers at the edge of the bandwidth in use. As with all parameters, there is a trade-off in play between the PAR that can be achieved, EVM, and other factors. Typically CFR will operate at the sample rate of DPD, that is, it will be placed between the DUC and 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 will be 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.
The 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 may be needed for “difficult” cases (some of which will be exposed in the "Examples" section) – 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 is sufficient.

The Number of CPGs at Each Iteration

The typical number is three to four, but there will be 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 may be different at each iteration to save resources if the performance is assured.

The Pulse Shape Coefficients

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 4. This requirement means that the spectra must broadly match. As regards 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 in not an accurate view because PC-CFR does not actually clip the signal. However, in a particular application this process can be implied. For example, in the single-carrier WCDMA example that follows, 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 20dB below the carrier.

For a single-centered carrier, the cancellation pulse is the impulse response of a suitable low-pass filter. There is no apparent general theory as to how this should be designed. Experience shows that any reasonable method can be used. Here we show the parameters 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 signal processing toolbox function for constrained equiripple is `firceqrip(ntaps, Fc/(fs/2), [Dpass, Dstop], 'slope', 0);`

and suitable parameters for the air interface standards explicitly supported are shown in Table 2.

<table>
<thead>
<tr>
<th>Standard</th>
<th>ntaps (see text)</th>
<th>Fc</th>
<th>Dpass_db</th>
<th>Dstop_db</th>
</tr>
</thead>
<tbody>
<tr>
<td>WCDMA</td>
<td>2*fs</td>
<td>1.85</td>
<td>-20</td>
<td>-55</td>
</tr>
<tr>
<td>CDMA 2000</td>
<td>4*fs</td>
<td>0.45</td>
<td>-20</td>
<td>-55</td>
</tr>
<tr>
<td>WiMAX (10 MHz BW)</td>
<td>2*fs</td>
<td>4.2</td>
<td>-20</td>
<td>-55</td>
</tr>
<tr>
<td>LTE (5 MHz BW)</td>
<td>2*fs</td>
<td>2</td>
<td>-20</td>
<td>-55</td>
</tr>
<tr>
<td>LTE (10 MHz BW)</td>
<td>2*fs</td>
<td>4.3</td>
<td>-20</td>
<td>-55</td>
</tr>
</tbody>
</table>
For the actual call to `firceqrip`, the Dpass and Dstop values must be converted to linear scaling via, for example:

\[ D_{\text{stop}} = 10^{(D_{\text{stop\_dB}}/20)} \]

The `ntaps` values should be rounded to the nearest even number below the calculated values.

The value of `ntaps` given for GSM exceeds the 511 maximum provisioned in the core for sample rates greater than 61.44 Msps. The filter design given allows the single carrier spectral mask specified in GERAN [Ref 4] to be met after CFR. In the event that higher sample rates are required (for wider total bandwidth, for example), a more relaxed specification can be met or more iterations can be used with increased allocator spacing (see the "The Allocator Spacing Values" section), which will have the effect of reducing the filtering required at each iteration. Careful simulation studies are required. At the time of writing this data sheet, it is not known whether the single-carrier mask will actually be mandatory for multi-carrier GSM signals.

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.

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.

PC-CFR can be used for mixed-mode signals; pulses for different air interfaces can be combined in the same way as multiple carriers for the same interface.

The supplied examples serve to further illuminate the processed described here.

**The Threshold Value(s)**

The threshold should be set at the maximum signal amplitude that is desired to be transmitted (to the PA). It may 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. However, when considering TDD signals or types that have a non-constant power (in a short-term moving average sense), it may be better to think in terms of the absolute peak power targets and/or consider the signal time domain details (this is done in the WiMAX example that follows).

Normally the threshold settings at each iteration are the same, and there is little evidence to suggest that it should be otherwise. However that is not to say that a special case will not arise to modify this view. In this situation, logic would suggest setting thresholds that decrease along the iterations chain.
The Allocator Spacing Values

The allocator spacing for single-carrier and contiguous multi-carrier spectra may be left at the zero default. Non-zero values come into play when non-adjacent carriers are present, and typically they help to reduce the EVM. Non-zero values are used for GSM and for two WCDMA or LTE carriers 10MHz apart, in the supplied examples.

MATLAB Simulation

The file pc_cfr_v2_0_msim.zip should be obtained and extracted. The functions require version R2009a or a compatible version. The examples require the Signal Processing Toolbox for the pulse design. If this is not available, the scripts should be modified to use the supplied example pulses. The folder pc_cfr_v2_0_msim appears with a subfolder lib and files named and described in Table 3. The contents of the lib folder are named and described in Table 4. The examples can be run, modified as required, and used as templates for other applications and further investigations.

Table 3: pc_cfr_v2_0_msim Contents

<table>
<thead>
<tr>
<th>File Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>examples_cdma2k.m</td>
<td>Simulation of three adjacent CDMA 2000 carriers with fs = 30.72 Msp and PAR_target = 6 dB</td>
</tr>
<tr>
<td>examples_gsm6c_narrow.m</td>
<td>Simulation of 6 GSM carriers spaced by 600KHz with fs = 30.72 Msp and PAR_target = 5.5 dB</td>
</tr>
<tr>
<td>examples_gsm6c_wide.m</td>
<td>Simulation of 6 GSM carriers spaced by 2MHz with fs = 61.44 Msp and PAR_target = 5.5 dB</td>
</tr>
<tr>
<td>examples_lte1001.m</td>
<td>Simulation of 2 x 5MHz LTE carriers spaced 10MHz apart with fs = 122.88 Msp and PAR_target swept from 5.5 to 8 dB in 0.5 dB steps</td>
</tr>
<tr>
<td>examples_lte20MHz.m</td>
<td>Simulation of one 20MHz LTE carrier spaced with fs = 122.88 Msp and PAR_target = 7 dB</td>
</tr>
<tr>
<td>examples_mixed_mode.m</td>
<td>Simulation of one 10MHz LTE carrier mixed with 4 GSM carriers in a total of 20MHz of spectrum with fs = 122.88 Msp and PAR_target = 6.5dB</td>
</tr>
<tr>
<td>examples_tds12c.m</td>
<td>Simulation of twelve TD-SCDMA carriers spaced by 1.6MHz with fs = 61.44 Msp and PAR_target = 7 dB</td>
</tr>
<tr>
<td>examples_tds5c9.m</td>
<td>Simulation of TD-SCDMA carriers occupying five of nine positions spaced by 1.6MHz with fs = 92.16 Msp and PAR_target = 7 dB</td>
</tr>
<tr>
<td>examples_tds6c.m</td>
<td>Simulation of six TD-SCDMA carriers spaced by 1.6MHz with fs = 61.44 Msp and PAR_target = 7 dB</td>
</tr>
<tr>
<td>examples_wcdma1001.m</td>
<td>Simulation of two WCDMA carriers spaced 10MHz apart with fs = 122.88 Msp and PAR_target swept from 5.5 to 8 dB in 0.5 dB steps</td>
</tr>
<tr>
<td>examples_wcdma1111.m</td>
<td>Simulation of four adjacent WCDMA carriers with fs = 122.88 Msp and PAR_target = 6 dB</td>
</tr>
<tr>
<td>examples_wcdma1c.m</td>
<td>Simulation of one WCDMA carriers with fs = 30.72 Msp and PAR_target swept from 5.5 to 8 dB in 0.5 dB steps</td>
</tr>
<tr>
<td>examples_wimax.m</td>
<td>Simulation a 10MHz WiMAX TDD frame at fs = 61.44 Msp and PAR_target = 3.5dB in the preamble</td>
</tr>
</tbody>
</table>
Table 4: pc_cfr_v2_0_msilib Contents

<table>
<thead>
<tr>
<th>File Name</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>pc_cfr_v2_0_iteration.p</td>
<td>Bit-true MATLAB model of the pc-cfr core</td>
</tr>
<tr>
<td>pc_cfr_v2_0_cpulses.xls</td>
<td>Tables of cancellation pulse coefficients for the cases given in &quot;The Pulse Shape Coefficients&quot; section.</td>
</tr>
<tr>
<td>rotate_and_scale.p</td>
<td>Forms correctly scaled single and multi-carrier cancellation pulses</td>
</tr>
<tr>
<td>make_cdma2k_data.p</td>
<td>Function to generate multi-carrier CDMA 2000 upsampled data</td>
</tr>
<tr>
<td>make_gsm_data.p</td>
<td>Function to generate multi-carrier GSM (GMSK) upsampled data</td>
</tr>
<tr>
<td>make_lte_data.p</td>
<td>Function to generate multi-carrier E-URAN (LTE) upsampled data</td>
</tr>
<tr>
<td>make_tdsdcdma_data.p</td>
<td>Function to generate multi-carrier TD-SCDMA upsampled data</td>
</tr>
<tr>
<td>make_wcdma_data.p</td>
<td>Function to generate multi-carrier WCDMA upsampled data</td>
</tr>
<tr>
<td>make_wimax_data.p</td>
<td>Function to generate 10MHz BW WiMAX TDD upsampled data</td>
</tr>
<tr>
<td>TM1_64chips_15dBFS.mat</td>
<td>Data file used by make_wcdma_data</td>
</tr>
<tr>
<td>wimax_tdd_75_per.mat and wimax_duc_10MHz_filters.mat</td>
<td>Data files used by make_wimax_data</td>
</tr>
<tr>
<td>gsm_gmsk_15p36.mat</td>
<td>Data file used by make_gsm_data</td>
</tr>
<tr>
<td>cCDF.p</td>
<td>CCDF utility</td>
</tr>
<tr>
<td>centered_psd.p</td>
<td>PSD utility</td>
</tr>
<tr>
<td>magz_whole.p</td>
<td>Frequency response utility</td>
</tr>
</tbody>
</table>

Output and discussion of some example runs follow in the subsequent sections. For brevity, only a selection is presented – cases which act to exemplify particular design principles.

**Examples**

- "WCDMA: One Carrier"
- "WCDMA: Two Carriers 10MHz Apart"
- "WiMAX"
- "Twelve TD-SCDMA Carriers"
- "Six GSM Carriers"
- "Mixed-mode LTE + GSM"
WCDMA: One Carrier

Selected Simulation Output

Figure 11: WCDMA: One Carrier, Selected Simulation Output
Discussion

The simulation produces the output shown along with additional graphical output (the signal magnitudes before and after each iteration) and test output (diagnostic information from each iteration).

The top left pane of Figure 11 plots the cancellation pulse shape (red) against the original spectrum, and the top right pane shows the signal spectrum before and after (green) CFR at the selected PAR_target stated. From these plots it can be inferred that the equivalent clipping noise is around -20dBc as discussed previously.

The CCDF and PAR versus EVM plots are self-explanatory.

The text output shown gives the core physical parameters and in text form, the information given in the EVM and CCDF plots, and also three other key pieces of information:

1. The PAR at fs followed by the interpolated PAR. Here we see small differences that tell us that the fs chosen is sufficient for the signal bandwidth, and the signal may be safely interpolated after CFR.
2. The number of CPGs actually employed at each iteration. Four were provisioned (as stated in the "General Considerations for Parameter Selection" section) but, for example, if the application required only a 7dB PAR target, then this information says that the core need be configured for only three CPGs per iteration.
3. Less than 10% EVM can be achieved with PAR less than 6dB. The data in the first row of the simulator output table shows this, and also indicates that the CCDF will be similar to that shown in the bottom hand pane of Figure 11.
WCDMA: Two Carriers 10MHz Apart

Selected Simulation Output

---

Core parameters
---

Number of iterations = 3
iteration 1: Number of CPGs = 6, allocator_spacing = 20
iteration 2: Number of CPGs = 6, allocator_spacing = 10
iteration 3: Number of CPGs = 6, allocator_spacing = 5

fs = 122.88 Msps, PAR before CFR = 9.88 dB

---

key to columns
---

PAR_target_dB
PAR achieved at fs
PAR of interpolated signal (245.76 Msps)
EVM

CPGs used at each iteration
CCDF: PAR as fs for 10^-1, -2 -3, -4 at -6 probability

| 5.50  | 6.02  | 6.13  | 12.32 | 6  | 6 | 3.69  | 5.73  | 5.99  | 6.02  | 6.07  | 6.42
| 6.00  | 6.38  | 6.49  | 9.95  | 6 | 6 | 3.70  | 5.97  | 6.35  | 6.38  | 6.41  | 6.69
| 6.50  | 6.76  | 6.88  | 7.85  | 6 | 6 | 3.70  | 6.19  | 6.74  | 6.76  | 6.79  | 7.01
| 7.00  | 7.17  | 7.28  | 6.03  | 6 | 4 | 3.69  | 6.38  | 7.15  | 7.17  | 7.19  | 7.23
| 7.50  | 7.60  | 7.70  | 4.49  | 5 | 3 | 2 | 3.68  | 6.54  | 7.59 | 7.60  | 7.62  | 7.63
| 8.00  | 8.06  | 8.12  | 3.22  | 4 | 3 | 2 | 3.67  | 6.66  | 7.93 | 8.06  | 8.08  | 8.09

---
**Discussion**

The core parameters were selected to give good EVM and CCDF performance down to 6.5 dB achieved PAR. The reader can readily determine, by running the simulation with adjusted parameters, that the non-zero allocator spacing values act to reduce the EVM, and that if the PAR requirements were higher, for example 7.5 dB, then only two iterations would be required.

**WiMAX**

*Selected Simulation Output*

![Graphs showing signal magnitude, block average dBFS, and block average PAR](image)

*Figure 13: WiMAX, Selected Simulation Output*
Figure 14: WiMAX, Entire Signal Amplitude before and after CFR

Figure 15: WiMAX, Spectra and CCDFs in the Preamble
Discussion

The signal here is one in which there is a need to take into account that power (in some short-term average sense) is not constant across the frame. The signal here is a 75% downlink ratio signal in a typical maximum power configuration.

The first plot of the simulation graphical output is the first millisecond of the 5mS frame and shows the mean power and PAR in 4096 sample blocks. The preamble is the highest power, lowest PAR portion of the frame, and is scaled to -15dBFS. The full frame is at -20dBFS.

The objective of CFR is to limit the peak power for a given mean power, and the limiting factor in choosing the threshold in this instance is the EVM budget. The ideal is to limit the peak power in the entire frame while maintaining about 2% EVM. It turns out that setting the PAR target to 3.5dB achieves this.

Figure 14 shows the entire signal amplitude before and after CFR, and Figure 15 shows the spectra and CCDFs in the preamble. The corresponding numbers for the preamble appear in the text output. So the PAR of the preamble is reduced from 4.3 to 3.7, that is, only 0.7dB. But further analysis shows that the PAR of the entire frame (including the zero section) is reduced from 10.3 to 8.7. The message here is to set the threshold in relation to the maximum power segment, and then the peak reduction of the entire frame follows.

Only one iteration is used in this simulation because the number of peaks that needs to be cancelled is relatively small. There is a residual time-domain ripple apparent in the preamble. If necessary, this can be removed with a second iteration.

<table>
<thead>
<tr>
<th>Core parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of iterations = 1</td>
</tr>
<tr>
<td>iteration 1: Number of CPGs = 4, allocator_spacing = 0</td>
</tr>
<tr>
<td>iteration 1: 350 peaks out of 350 were allocated max cpgs used = 3</td>
</tr>
</tbody>
</table>

| fs = 61.44 Msps, PAR before CFR = 4.32 dB |

<table>
<thead>
<tr>
<th>PAR_target_db</th>
<th>PAR achieved at fs</th>
<th>PAR of interpolated signal (245.76 Mps)</th>
<th>EVM</th>
<th>CPGs used at each iteration</th>
<th>CCDF: PAR as fs for 10^-1, -2 -3, -4 at -6 probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.50</td>
<td>3.74</td>
<td>3.77</td>
<td>1.82</td>
<td>3</td>
<td>2.82</td>
</tr>
<tr>
<td>3.61</td>
<td>3.71</td>
<td>3.74</td>
<td>3.74</td>
<td>3</td>
<td>3.74</td>
</tr>
<tr>
<td>3.74</td>
<td>3.74</td>
<td>3.74</td>
<td>3.74</td>
<td>3</td>
<td>3.74</td>
</tr>
</tbody>
</table>
Twelve TD-SCDMA Carriers

Selected Simulation Output

Figure 16: Twelve TD-SCDMA Carriers, Selected Simulation Output

Core parameters

Number of iterations = 2
iteration 1: Number of CPGs = 6, allocator_spacing = 0
iteration 2: Number of CPGs = 3, allocator_spacing = 0
iteration 1: 940 peaks out of 1062 were allocated max cpgs used = 5
iteration 2: 191 peaks out of 193 were allocated max cpgs used = 3

fs = 61.44 Msp, PAR before CFR= 9.68 dB

key to columns

PAR_target_dB
PAR acheived at fs
PAR of interpolated signal (245.76 Mps)
EVM
CPGs used at each iteration
CCDF: PAR as fs for 10^-1, -2 -3, -4 at -6 probability
7.00 7.08 7.35 4.64 5 3 3.66 6.49 7.05 7.08 7.23 7.23

Discussion

This signal which is 20MHz wide is processed at 61.44 Msp. The only ill effect of this is that the interpolated PAR is about 0.3dB higher than the PAR at fs. The PAR can be reduced (to about 7.1) for the same EVM by processing at 122.88, but the hardware resource for this would be more (for the same clock rate).
Six GSM Carriers

**Selected Simulation Output**

---

**Core parameters**

---

- Number of iterations = 3
- iteration 1: Number of CPGs = 4, allocator_spacing = 160
- iteration 2: Number of CPGs = 4, allocator_spacing = 90
- iteration 3: Number of CPGs = 4, allocator_spacing = 0
- iteration 1: 962 peaks out of 4267 were allocated max cpgs used = 3
- iteration 2: 888 peaks out of 2050 were allocated max cpgs used = 4
- iteration 3: 242 peaks out of 245 were allocated max cpgs used = 4

---

fs = 61.44 Msps, PAR before CFR = 7.97 dB

---

**key to columns**

---

- PAR_target_dB
- PAR achieved at fs
- PAR of interpolated signal (245.76 Mps)
- EVM
- CPGs used at each iteration
- CCDF: PAR as fs for $10^{-1}$, $10^{-2}$, $10^{-3}$, $10^{-4}$ at $10^{-6}$ probability

5.50 5.83 5.87 8.17 3.44 3.82 5.49 5.80 5.83 5.85 5.85
**Discussion**

The key point in this case is that GSM is a ‘difficult signal’ – the peaks are very dense because they are entirely due to carrier mixing, as GMSK is a constant modulus signal. Therefore three iterations are used with carefully selected allocator spacing values. Frequency-hopping GSM can be accommodated with dynamic cancellation pulse switching as discussed in the “Dynamic Cancellation Pulse Switching” section.

**Mixed-mode LTE + GSM**

**Selected Simulation Output**

![Figure 18: Mixed-mode LTE + GSM, Selected Simulation Output](image1)

![Figure 19: GSM Single-carrier Mask](image2)
Core parameters

Number of iterations = 3
iteration 1: Number of CPGs = 4, allocator_spacing = 350
iteration 2: Number of CPGs = 4, allocator_spacing = 120
iteration 3: Number of CPGs = 4, allocator_spacing = 60
iteration 1: 609 peaks out of 944 were allocated max cpgs used = 2
iteration 2: 285 peaks out of 332 were allocated max cpgs used = 2
iteration 3: 51 peaks out of 53 were allocated max cpgs used = 2

---

Discussion

Figure 19 plots the GSM single-carrier mask derived from [Ref 4] by the topmost GSM carrier. The single-carrier mask is just met at a 76.8 Msps sample rate used here (with 511 pulse coefficients, the 6.144 Msps limit discussed previously allowed for margin to mask), but the PAR of 7.8 dB cannot be improved without using a higher sample rate. At 122.88 Msps, 6.8 dB PAR can be achieved, but the single-carrier mask is broken (with 511 pulse coefficients). As discussed previously, further detail needs to be provided in respect of the specifications, and further parameter optimization could be performed. Nonetheless, it is clear the PC-CFR core is feasible for this emerging application.

Resource Requirements and Performance

Resource requirements and performance are dependent mainly on the Data-Rate and Number of Cancellation Pulse Generators used in the configuration. They are also susceptible to the version of implementation tools used. Therefore, these example figures should be used as a guide only and 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.

Table 5, Table 6 and Table 7 provide resource requirements and performance for the families Virtex-5, Spartan-6 and Virtex-6 respectively. For all configurations, the default datawidth of 16 bits has been used with the coefficient setup selected as 'Single pulse configurable coefficients.'
**Note:** The size and performance of the core depend on the specific customer application and the tool version used. These sizes and clock rates are obtained with a single core placed in an otherwise empty Xilinx FPGA and should be used as a guide only.

### Glossary

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAR</td>
<td>Peak to Average Ratio. The comparison between the peak signal amplitude that occurs with 10^-4 probability and the mean. Also known as “Peak to Mean ratio” and PAPR (peak to average power ratio).</td>
</tr>
<tr>
<td>EVM</td>
<td>Error Vector Magnitude. The values reported here are obtained by comparing the total signals before and after CFR. Standards compliance measurement usually required this to be performed on demodulated baseband data. However good correlation is observed between the EVM values reported here and, for example, the values seen on the Agilent PSA for the WCDMA signals.</td>
</tr>
<tr>
<td>dBFS</td>
<td>dB power relative to full scale. The (FDD) signals used in the simulation here are at -15dBFS within a 16 bit range, that is, the standard deviation is 32768*10^{-15/20} = 5827. For TDD signals, this applied to the maximum power segment.</td>
</tr>
<tr>
<td>FDD</td>
<td>Frequency Division Duplex. The inference here is that the signal frame has constant power within some timescale average.</td>
</tr>
<tr>
<td>TDD</td>
<td>Time Division Duplex. The signal frame is characterized by some blank periods and/or a non-uniform power profile.</td>
</tr>
</tbody>
</table>
References

1. XAPP1128, “Digital Predistortion v2.0”
2. XAPP921, “High Density WCDMA Digital Front End Reference Design”
3. XAPP961, “High Density WiMAX Digital Front End Reference Design”
4. 3GPP Technical Specification GSM/EDGE, TS 45.005 V7.12.0 “Radio transmission and reception (Release 7)”

Evaluation

An evaluation license is available for this core. The evaluation version operates in the same way as the full version for several hours, dependant on clock frequency. Allocation of the cancellation pulse generators is then completely disabled, and the data output will comprise a delayed version of the data input. If you notice this behavior in hardware, it probably means 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 XCO file, reconfigure and regenerate the core.

Support

Xilinx provides technical support at www.xilinx.com/support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

Refer to the IP Release Notes Guide (XT025) for further information on this core. There will be a link to all the DSP IP and then to the relevant core being designed with.

For each core, there is a master Answer Record that contains the Release Notes and Known Issues list for the core being used. The following information is listed for each version of the core:

- New Features
- Bug Fixes
- Known Issues

Ordering Information

This Xilinx LogiCORE IP module is provided under the terms of the Xilinx Core Site License and the core is generated using the Xilinx ISE® CORE Generator software. The CORE Generator software is shipped with the Xilinx ISE Design Suite software.

For full access to all core functionality in simulation and in hardware, you must purchase a license for the core. Please contact your local Xilinx sales representative for information on pricing and availability of Xilinx LogiCORE IP modules. Information about additional modules is also available at the Xilinx IP Center.
Revision History

The following table shows the revision history for this document:

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description of Revisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/02/09</td>
<td>1.0</td>
<td>Initial Xilinx release.</td>
</tr>
</tbody>
</table>

Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter “Information,” to you “AS IS” with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.