HIGH-DENSITY DIGITAL DOWNCONVERTER AND UPCONVERTER

FEATURES

- **Optimized for CDMA2000−1X and UMTS**
- **Up to 12 UMTS or 24 CDMA2000 Downconverter and Upconverter Channels**
- **Mixed CDMA2000−1X and UMTS Operation**
- **DDC Input and DUC Output Rates to 125 MSPS**
- **Any DDC Can Connect to Any of Four Input Ports**
- **Any DUC Can Sum into Any of Four Output Ports**
- **Real/Complex DDC Inputs and DUC Outputs**
- **Programmable AGC on DDC Outputs**
- **Rx Filtering: 6 Stage CIC, 48 Tap CFIR, 64 Tap PFIR**
- **Tx Filtering: 6 Stage CIC, 47 Tap CFIR, 63 Tap PFIR**
- **115-dB SFDR**
- **16-Bit DDC Inputs, 18-Bit DUC Outputs**
- **1.5-V Core, 3.3-V I/O**

1 Description

The GC5316 is a high-density multi-channel communications signal processor integrated circuit that provides both digital downconversion and digital upconversion optimized for cellular base transceiver systems. The device supports both UMTS and CDMA2000 (CDMA) air interface cellular standards.

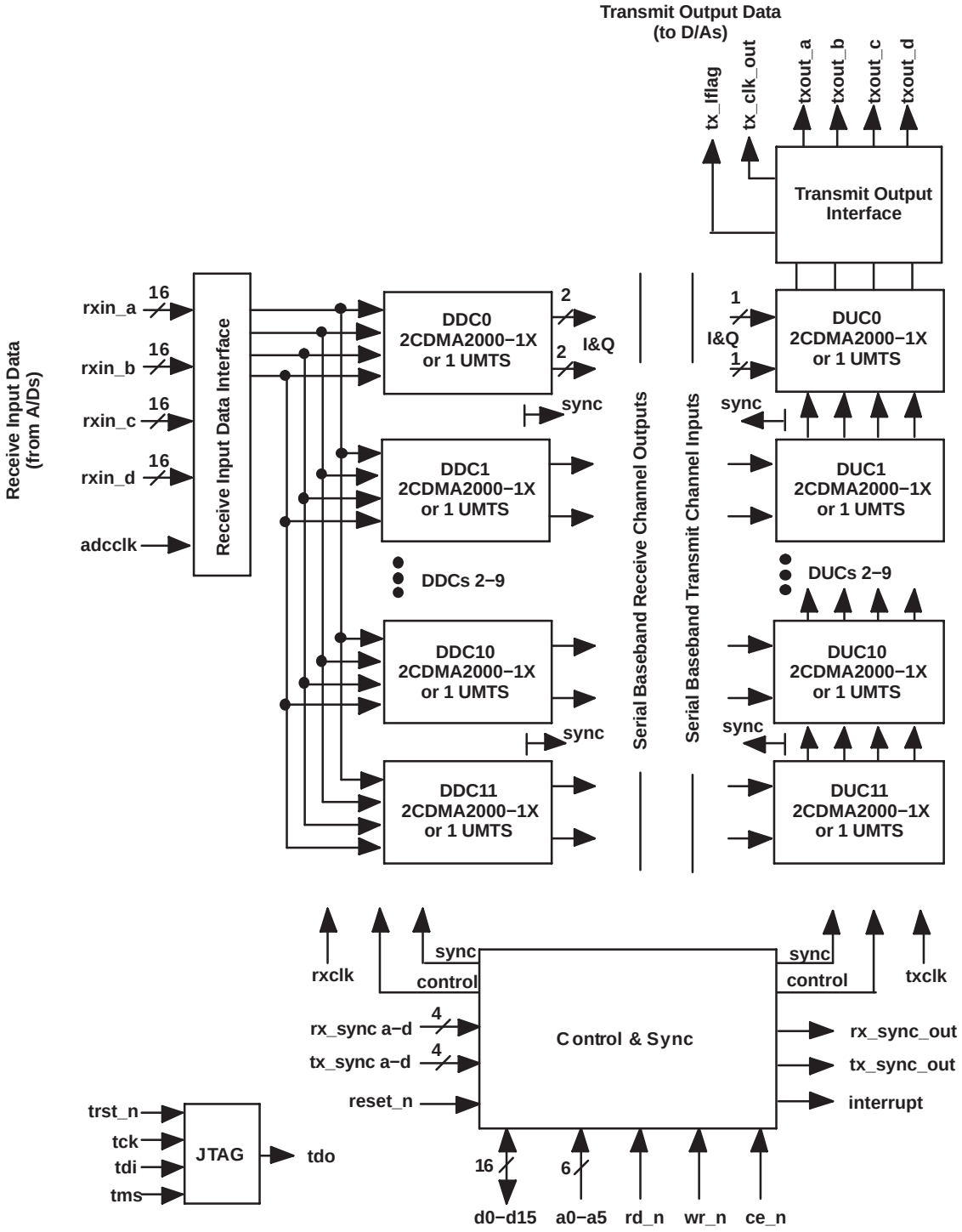
The chip provides up to 24 CDMA digital downconverter (DDC) and digital upconverter (DUC) channels or 12 UMTS DDC and DUC channels. The GC5316 can also support a combination of CDMA and UMTS channels. The DDC and DUC channels are independent and operate simultaneously.

The chip is ideal for cellular base transceiver systems where a large number of digital radio channels are required. Each of the 24 CDMA (or 12 UMTS) channels can operate independently. On the DDC side there are four 16 bit input ports that can accept real or complex input data. The input ports are driven with parallel data, typically from an analog-to-digital converter. Each downconverter channel can be programmed to accept data from any one of the four input ports.

On the DUC side, there are four 18-bit output ports. Each output port can sum any of the DUC channels in a daisy-chain fashion. This permits creating a stack of CDMA or UMTS signals. These ports can output either real or complex data. Real output data would generally drive one or more D/A converters and output the stack of signals at an intermediate frequency (IF). Complex data (at baseband or an IF) is used when a quadrature modulator upconversion scheme is employed. Complex output data can also be used when the output stack is further processed using crest factor reduction or power amplifier predistortion techniques.

Table of Contents

1.1 Functional Block Diagram



1.2 Package/Ordering Information

PRODUCT	PACKAGE LEAD	PACKAGE DESIGNATOR	SPECIFIED TEMPERATURE RANGE	PACKAGE MARKING	ORDERING NUMBER	TRANSPORT MEDIA, QUANTITY
GC5316	Thermally Enhanced Plastic BGA w/Heat Slug − 388	ZED	−40°C to 85°C	GC5316IZED	GC5316IZED	Tray, 40

2 GC5316 Receive

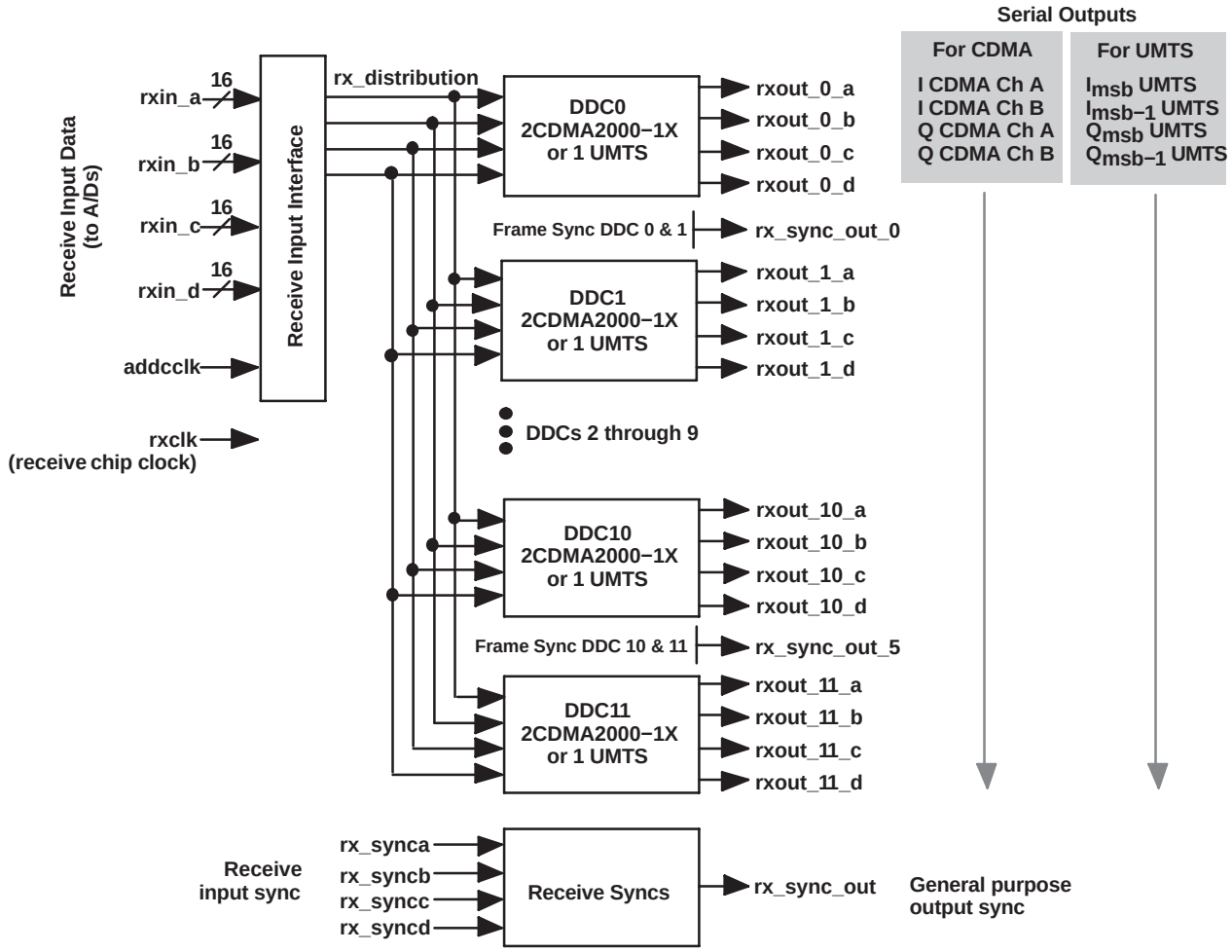


Figure 1. Receive Section

The receive section of the GC5316 consists of the receive input interface, the rx_distribution bus, and 12 digital downconverter blocks.

The purpose of the receive input interface is to accept signal data from four input ports (generally from analog-to-digital converters) and to distribute the data to the DDC blocks. The input interface also has a user-controlled test generator and noise source, as well as a resampling block. The resampler accepts real inputs at 3/4rxclk or rxclk rate, mixes down by Fs/4, low-pass filters, and decimates to rxclk/2. This is useful for handling data at 3/4 rxclk rate (for example, a 92.16-MSPS adcclk rate with a 122.88-MHz rxclk). It is also useful to process more than 12 CDMA signals when sampling at rxclk rate.

The rx_distribution bus distributes the four channels of signal data to each of the 12 DDC blocks.

Each DDC block selects one of the four channels from the rx_distribution bus and then performs downconversion tuning, programmable delay, channel filtering with decimation, power measurement, fixed gain adjust, and automatic gain control. Each DDC block can support one UMTS channel or two CDMA channels. An optional mode permits stacking two DDC blocks to provide double-length channel filtering. Tuned, filtered, and decimated signal data is output in bit serial format.

2.1 Receive Input Interface

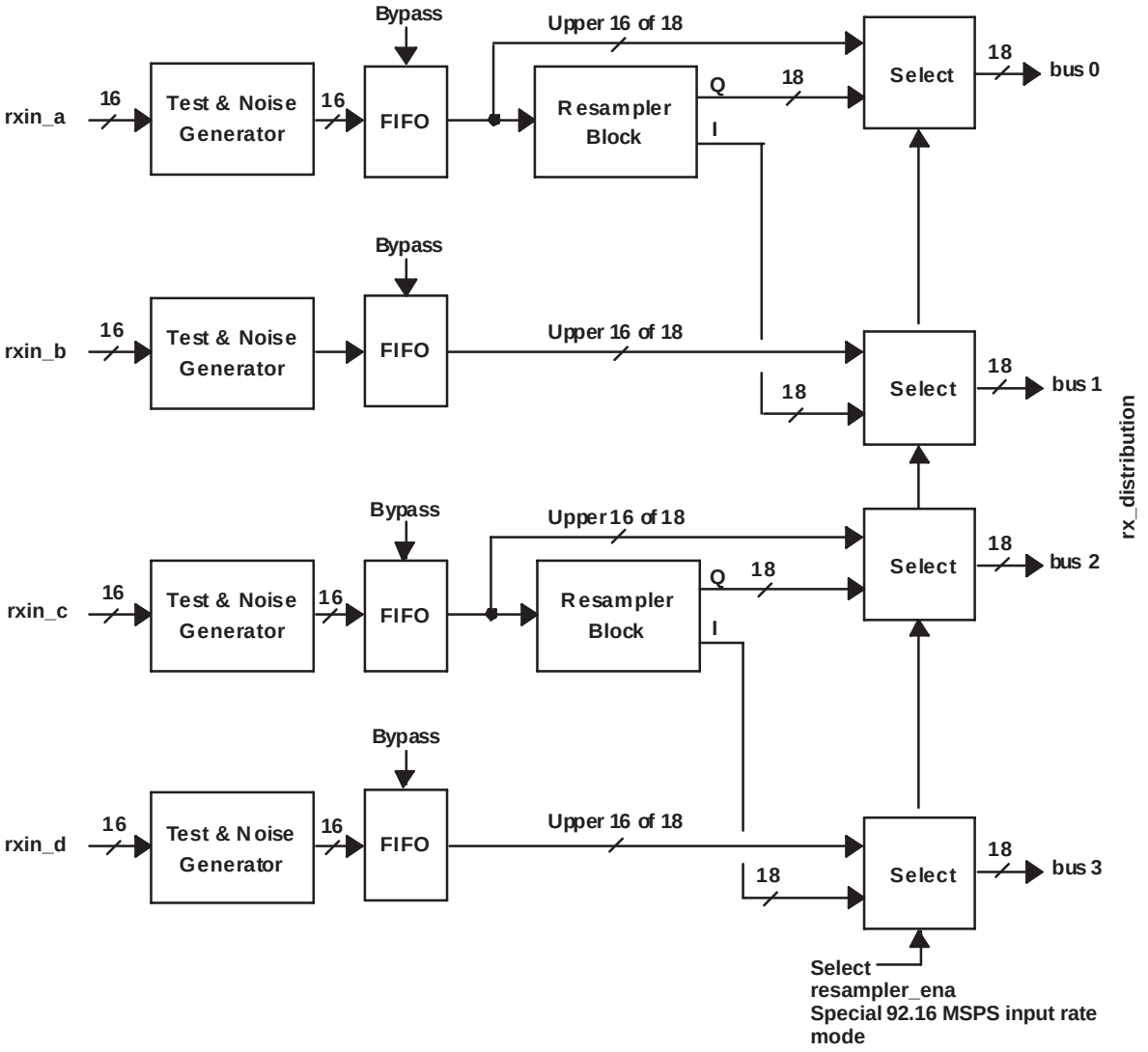


Figure 2. Receive Input Interface

The four ports support four independent real input signals or two complex. Complex signal data is input with I data driving one input port and Q data driving another. This means that there are only two signal data ports available when in complex input mode. The mapping of I and Q data onto the four input ports is programmable.

2.1.1 Test and Noise Generator and FIFO

Incoming data first enters the test and noise generator block. This block can either pass the data through, replace the input data with a pattern generated internally (useful in test), or can add noise to the input at a user programmable level.

Most applications pass data through this block unchanged by clearing slf_tst_ena, rduz_sens_ena, and tst_on.

Test sequences are useful for board bring-up or for power-on self-test. Board bring-up and self-test procedures and configuration files are available on the web. Self-test will be described in greater detail in a later section. Suffice to say here that the receive data input is replaced by pseudo-random patterns at this point in the processing chain.

A few applications that require receiver desensitization which is done by adding digital noise to the input. The same pseudo-random sequence generator is used as a noise source. Nz_pwr_mask is used to select which input data bits get noise added to them. In this way the user has control over the noise power introduced in receiver desensitization.

Next the signal is sent to an 8-stage FIFO. This allows an arbitrary phase relationship between adcclk and rxclk. The frequency relationship is fixed by the configuration. The FIFO can be bypassed, clocking the input data directly on rxclk. Note that if the FIFO is bypassed, the hold times are longer than usual. If the input rate is a fraction (1/2, 1/4, or 1/8) of rxclk then ssel_rxin determines which of the multiple rising clock edges are used to sample the data.

Table 1. Programming

VARIABLE	DESCRIPTION
rduz_sens_ena	When enabled adds noise to the ADC input using nz_pwr_mask.
nz_pwr_mask	Selects the noise bits to be added to the ADC input sample when rduz_sens_ena is one.
adc_fifo_bypass	When asserted bypasses the input FIFO. Data is latched directly using the rxclk input when FIFO is bypassed. Should set this to 0 when input data is to be latched using the adcclk input. Most applications should not bypass the fifo.

2.1.2 Resampler Block

Two of the four 16-bit data input ports, rxin_a, and rxin_c have resampler blocks. The data is clocked from an external clock signal adcclk. The real input signal is downconverted by adcclk/4 and is then low-pass filtered and decimated.

Decimation can be either by 1.5 or by 2. Both these decimation modes support up to 24 CDMA DDC channels, or 12 UMTS DDC channels.

Resampler Decimate by 1.5

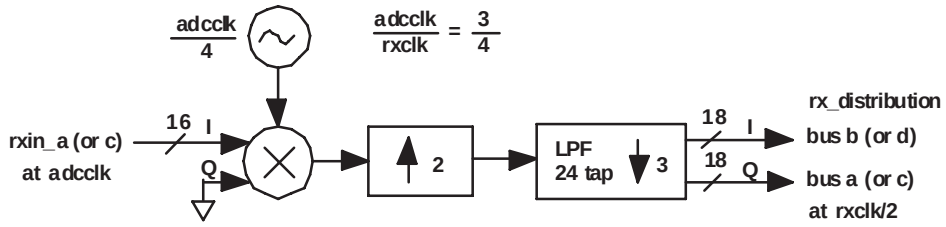


Figure 3. Resampler Decimate by 1.5 Mode

The decimate by 1.5 mode requires the input data rate to be 3/4 of the receive clock rate (adcclk frequency is 3/4 rxclk frequency). A 24-tap low-pass decimation filter with programmable 18-bit coefficients removes alias images that would fold into the passband prior to decimation. The following table shows the performance of filters designed for various bandwidths when the resampler is decimating by 1.5. The table also shows the resulting passband frequencies assuming the input data rate is 92.16 MSPS. Each horizontal row is a unique 24-tap filter which is available on the web.

Table 2. Resampler Filter Performance in the Decimate by 1.5 Mode

GENERAL APPLICATION			EXAMPLE APPLICATION			
			adclk: 92.16 MHz			
PASSBAND	RIPPLE	STOPBAND	BANDWIDTH	F LOWER	F CENTER	F UPPER
of clk	dB	dB	MHz	MHz	MHz	MHz
0.05	0	−101.5	18.4	13.8	23	32.3
0.06	0	−98	22.1	12	23	34.1
0.07	0	−91.7	25.8	10.1	23	35.9
0.08	0.01	−84.1	29.5	8.3	23	37.8
0.09	0.03	−75.6	33.2	6.5	23	39.6
0.1	0.07	−67.1	36.9	4.6	23	41.5
0.11	0.18	−59.5	40.6	2.8	23	43.3
0.05	0	−101.5	18.4	13.8	23	32.3
0.06	0	−98	22.1	12	23	34.1

Resampler Decimate by 2

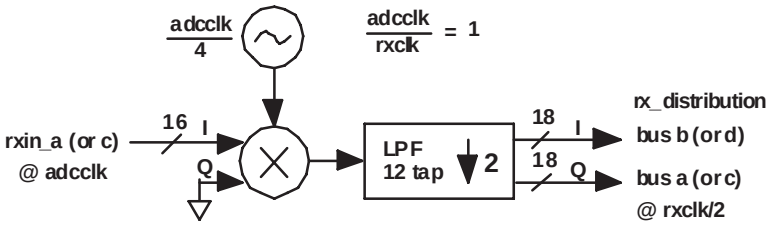


Figure 4. Resampler Decimate by 2 Mode

The decimate by 2 mode permits input data rates up to the rxclk rate (adcclk frequency equals rxclk frequency). This is useful for processing up to two real inputs at the rxclk rate and extracting more than 12 CDMA signals. A 12-tap low-pass decimation filter with programmable 18-bit coefficients removes alias images that would fold into the passband prior to decimation. Table 3 shows the performance of filters designed for various bandwidths when the resampler is decimating by 2. Table 3 also shows the resulting passband frequencies assuming the input data rate is 122.88 MHz. Each horizontal row is a unique 12-tap filter which is available on the web.

Table 3. Resampler Filter Performance in the Decimate by 2 Mode

GENERAL APPLICATION			EXAMPLE APPLICATION			
			adclk: 122.88 MHz			
PASSBAND	RIPPLE	STOPBAND	BANDWIDTH	F LOWER	F CENTER	F UPPER
of clk	dB	dB	MHz	MHz	MHz	MHz
0.06	0	−109.8	14.7	23.3	30.72	38.1
0.07	0	−98.8	17.2	22.1	30.72	39.3
0.08	0	−93.4	19.7	20.9	30.72	40.6
0.09	0.01	−89.2	22.1	19.7	30.72	41.8
0.1	0.02	−81.5	24.6	18.4	30.72	43
0.11	0.05	−76.8	27	17.2	30.72	44.2
0.12	0.09	−71.4	29.5	16	30.72	45.5
0.13	0.16	−66.5	31.9	14.7	30.72	46.7
0.14	0.28	−61.8	34.4	13.5	30.72	47.9

The output of this processing block is complex at rxclk/2 and goes through the selector to drive the rx_distribution bus. I data for channel rxin_a is routed to a selector driving DDC bus 1, the Q data is input to a selector driving DDC bus 0. I data for channel rxin_c is routed to a selector driving DDC bus 3, the Q data is input to a selector driving DDC bus 2.

Table 4. Programming

VARIABLE	DESCRIPTION
resampler_ena	When asserted, turns on the resamplers on input ports rxin_a and rxin_c.
resampler_decim	1 = decimate by 1.5x, 0 = decimate by 2x
rate_sel	This selects the FIFO output rate when adc_fifo_bypass = 0. When using the resampler, this value should be programmed to a 0. When set to 0, the FIFO output is clocked by rxclk (gated if resampler is on and decimating by 1.5). When set to 1, the FIFO output rate is 1/2 of rxclk rate. When set to 2, the FIFO output rate is 1/4 of rxclk rate, and when set to 3, the FIFO output is at 1/8 of rxclk rate. e.g.: With rxclk 122.88MHz, set rate_sel to 0, 1, 2, or 3 respectively for adcclk 122.88, 61.44, 30.72, or 15.36 MHz.
ssel_rxin(2:0)	Synchronizes the rx_distribution bus source and destination and clock generation in each of the DDC blocks.
ssel_resamp(2:0)	Synchronizes the resampler Fs/4 mixer and decimation.
ssel_adc_fifo(2:0)	Synchronizes the FIFO read and write pointers (fifo depth).
remix_only	Set to 0 for complex input data, or to 1 for real data. Set this value to 0 when using the resampler. Note that mixed ral and complex input is not allowed.
resampler coefficients	The resample's 18-bit coefficients are loaded by the software cmd5316. The user must provide a coefficient file with one integer coefficient per line.

2.2 DDC Organization

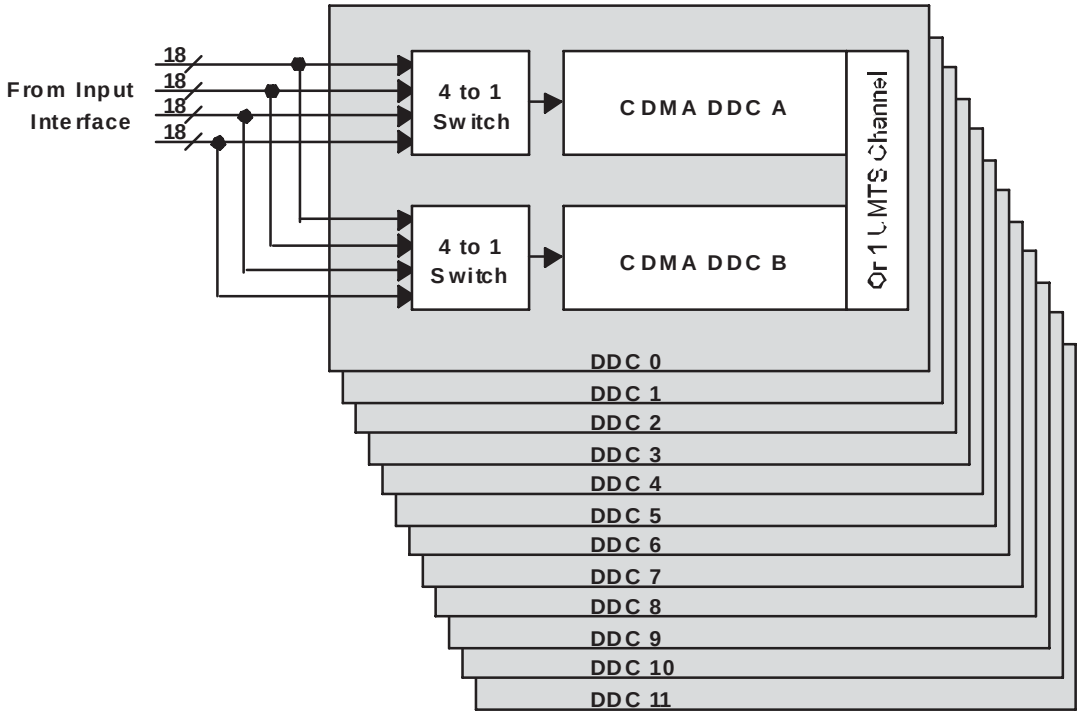


Figure 5. Receive DDC Blocks

The GC5316 provides downconversion for up to 24 CDMA2000 receive channels or 12 UMTS receive channels. Downconversion channels are organized into 12 DDC blocks. Each DDC block provides two CDMA2000 DDC channels, A and B, or one UMTS channel. Each DDC block has its own register set and may be programmed independently (except for parameters specifying the configuration of the rx_distribution bus).

Two DDC blocks (for example, DDC 0 and DDC 1) can be strapped together to form a single UMTS DDC channel with double-length filtering. The GC5316 can therefore provide six UMTS DDC channels with double-length FIR filtering.

Configuration parameters ending in _a apply to the A CDMA channel or to the UMTS channel. Parameters ending in _b apply the B CDMA channel and are unused in UMTS mode. Many parameters are shared between the A and B channels in CDMA mode (such as filter coefficients), while key parameters are independent for the two channels (such as frequency, phase, and gain).

Both CDMA DDC channels in a block can be independently tuned, though they would likely be used as diversity pairs and tuned to the same frequency. Filter coefficients are shared between the two CDMA DDC channels within a block.

Table 5. Programming

VARIABLE	DESCRIPTION
ddc_duc_ena	When set turns on the DDC. When cleared the clock to the DDC is stopped reducing its power consumption to essentially zero.
cdma_mode	When set, puts the DDC block in dual CDMA2000 mode.

2.3 Receive Downconverter Function Blocks

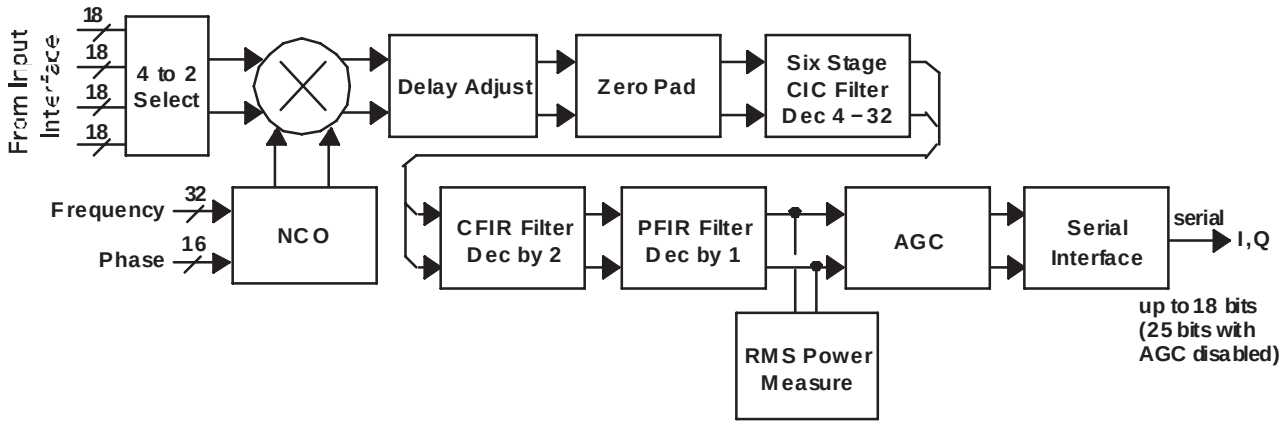


Figure 6. Receive Downconverter Function Blocks

The GC5316 downconversion channel can process two CDMA carriers or a single UMTS carrier. Signal data is selected from one of four ports and can be either real or complex. Data from the selected port is multiplied with a complex, programmable numerically controlled oscillator (NCO) which tunes the signal of interest to baseband. The delay adjust and zero pad blocks permits adjustment of the delay in the end-to-end channel. Zero padding interpolates the signal to the rxclk rate. Filtering consists of a six stage CIC filter which decimates the tuned data by a factor from 4 to 32, a compensating FIR filter (CFIR) which decimates by a factor of two, followed by a programmable FIR filter (PFIR) which does not decimate.

The RMS power meter measures the power within the channel's bandwidth. The AGC automatically drives the gain and keeps the magnitude of the signal at a user-specified level. This allows fewer bits to represent the signal. The serial output interface formats and rounds the output data. Each of the above blocks is described in greater detail in the following sections.

2.3.1 Receive Mixer

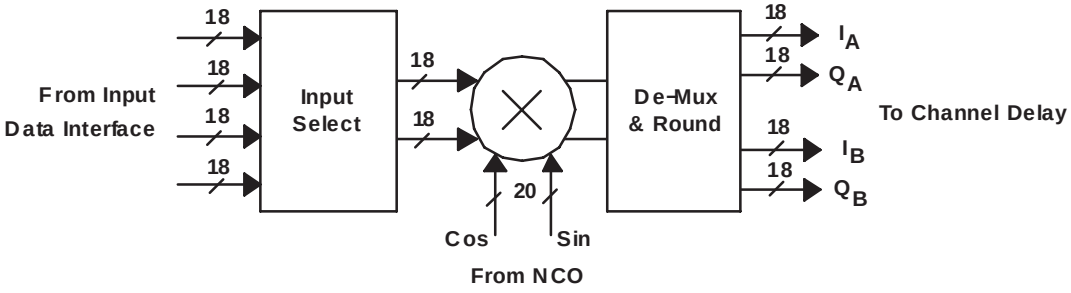


Figure 7. Receiver Mixer

The input select routes one of the four buses in rx_distribution to the I port and a second bus to the Q port of the mixer. In CDMA mode, bus selection is time multiplexed so the two channels may select different data sources if desired. Table 6 shows the bus selection.

Table 6. Bus Selection

SELECT VALUE	DATA TAKEN FROM RX_DISTRIBUTION BUS	
	I Data	Q Data
0	Bus a	Bus a
1	Bus b	Bus b
2	Bus c	Bus c
3	Bus d	Bus d
4	Bus a	Bus b
5	Bus a	Bus c
6	Bus a	Bus d
7	Bus b	Bus a
8	Bus b	Bus c
9	Bus b	Bus d
10	Bus c	Bus a
11	Bus c	Bus b
12	Bus c	Bus d
13	Bus d	Bus a
14	Bus d	Bus b
15	Bus d	Bus c

The receive mixer translates the input from selector to baseband where subsequent filtering is performed to isolate the signal of interest. The mixer is a complex multiplier that accepts 18-bit I and 18-bit Q signal data from the receive input interface and 20-bit sine and cosine sequences from the NCO. The NCO generates a mixing frequency (sometimes referred to as a local oscillator, or LO) specified by the user so that the desired signal of interest is tuned to 0 Hz. The NCO is discussed in detail in the next section.

A DDC channel can support one UMTS signal directly, or two CDMA channels at half the input rate. When in CDMA mode each channel may set the path selection, the mixer tuning and phase independently. The mixer output produces two complex streams; one representing the signal path for the A-side DDC, the other the B-side. Each of these streams drives a channel delay and zero pad block.

The maximum input rate for UMTS is rxclk for either real or complex input data. The maximum rx_distribution rate in CDMA mode (real or complex inputs) is rxclk/2. When adcclk/rxclk > $1/2$, this is normally accomplished using the resampler. For unusual cases where adcclk/rxclk = 1 and the resampler is not used, cdma_mode must be zero so only one signal can be processed in a DDC block, even if it is a CDMA signal.

The mixer gain is:

$$\text{Gain} = 2^{\text{mixer\_gain} - 2}$$

Table 7. Programming

VARIABLE	DESCRIPTION
ddcmux_sel_a(3:0)	Programs the I and Q complex input data routing onto two of the four input ports for stream A of CDMA DDC
ddcmux_sel_b(3:0)	Programs the I and Q complex input data routing onto two of the four input ports for stream B of CDMA DDC
remix_only	Set to 0 for complex input data, or to 1 for real data
ch_rate_sel(1:0)	Informs the DDC of the rx_distribution bus rate (1, 1/2, 1/4, or 1/8 rxclk for settings 0, 1, 2, or 3 respectively. Note this parameter must be the same in all enabled DDC blocks.
mixer_gain	When asserted adds 6 dB of gain in the mixer. This gain is highly recommended.

2.3.2 Receive Number Controlled Oscillator (NCO)

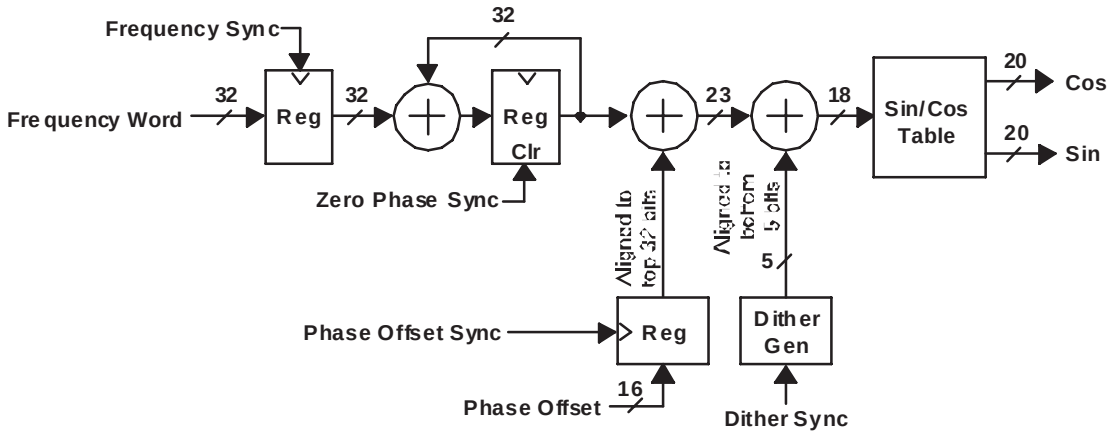


Figure 8. Receive Number Controlled Oscillator

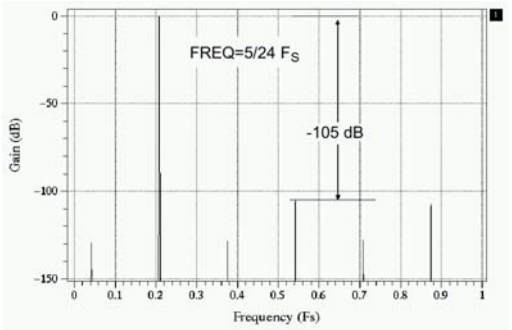
The NCO is a digital complex oscillator that is used to translate (or downconvert) an input signal of interest to baseband. The block produces programmable complex digital sinusoids by accumulating a frequency word which is programmed by the user. The output of the accumulator is a phase argument that indexes into a Sin/Cos ROM table which produces the complex sinusoid. A phase offset can be added prior to indexing if desired for channel calibration purposes. This changes the Sin/Cos phase with respect to other channels' NCOs.

A 5-bit dither generator is provided and generates a small level of digital pseudo-noise that is added to the phase argument below the bottom bit and is useful for reducing NCO spurious outputs.

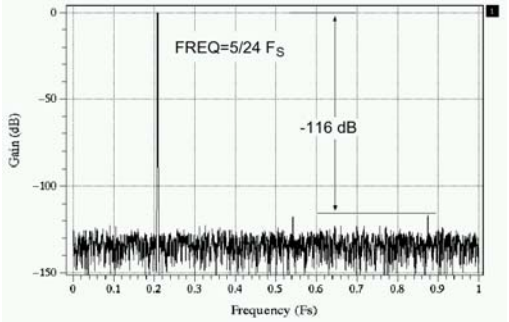
Table 8. Programming

VARIABLE	DESCRIPTION
dither_ena(2)	When set turns dither on. Clearing turns dither off.
test_bits_1(1:0)	Test bits. MUST be cleared for normal operation.

The NCO spurious levels are better than –115 dBC. Added phase dither randomizes the periodic nature of the phase accumulation process and reduces low-level spurious energy. For some frequencies (K×Fs/24), dither is ineffective – in these cases an initial phase of four reduces NCO spurs. Figure 9 and Figure 10 show the spur level performance of the NCO without dither, with dither, and with a phase offset value.

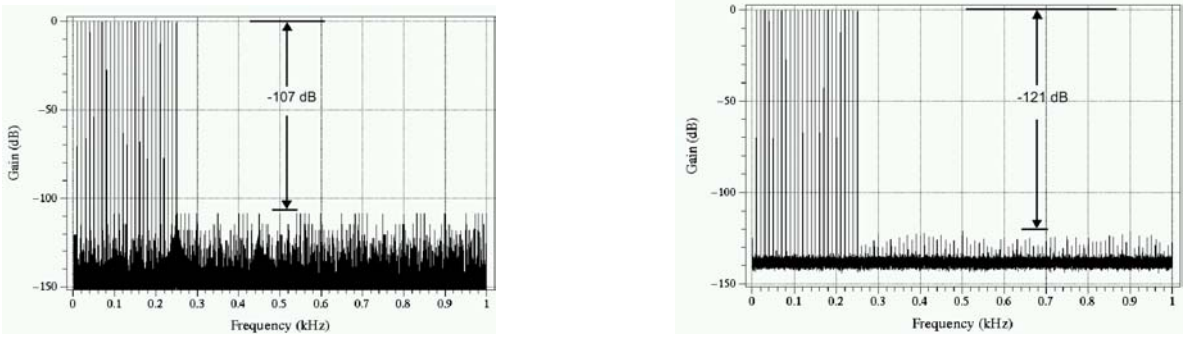


a) Worst Case Spectrum Without Dither



b) Spectrum With Dither (Tuned to Same Frequency)

Figure 9. Example NCO Spurs With and Without Dither



a) Plot Without Dither or Phase Initialization **b) Plot With Dither and Phase Initialization**

Figure 10. NCO Peak Spur Plot

The tuning frequency is specified as a 32-bit frequency word and is programmed as two sequential 16-bit words over the control port. The NCO operates at the same speed as the rx_distribution or rxclk / (tadj_interp_decim + 1). The NCO frequency resolution is simply the $F_{clk}/ 2^{32}$. As an example, at an input clock rate of 61.44 MHz, the frequency step size would be approximately 14 milli-Hertz (mHz). The frequency word is determined by the following formula:

$$\text{Frequency Word (in decimal)} = 2^{32} \times \frac{\text{Tuning Frequency}}{F_{clk}}$$

Note that frequency tuning words can be positive or negative valued. Specifying a positive frequency value translates negative frequencies upwards towards 0 Hz. Specifying a negative tuning frequency translates positive frequencies downwards towards 0 Hz.

Table 9. Programming

VARIABLE	DESCRIPTION
phase_add_a(31:0)	32-bit tuning frequency word for the A-side DDC when in CDMA mode. Also for UMTS mode.
phase_add_b(31:0)	32-bit tuning frequency word for the B-side DDC when in CDMA mode. Not used in UMTS mode.

Each of the 24 CDMA DDC channels can be loaded with unique frequency words.

The phase of the NCO's Sin/Cos output can be adjusted relative to the phase of other channel NCOs by specifying a phase offset. The phase offset is programmed as a 16-bit word, yielding a step size of about 5.5 milliDegrees. The phase offset word is determine by the formula:

$$\text{Phase Offset Word} = 2^{16} \times \frac{\text{Offset in Degrees}}{360} \text{ or}$$

$$\text{Phase Offset Word} = 2^{16} \times \frac{\text{Offset in Radians}}{2\pi}$$

Table 10. Programming

VARIABLE	DESCRIPTION
phase_offset_a(15:0)	16-bit phase offset word for the A-side DDC when in CDMA mode. Also for UMTS mode.
phase_offset_b(15:0	16-bit phase offset word for the B-side DDC when in CDMA mode. Not used in UMTS mode.

Each of the 24 CDMA DDC blocks can be loaded with unique phase offset words.

Various synchronization signals are available which are used to synchronize the NCOs of all channels with respect to each other. Frequency sync (ssel_freq) and phase offset sync (ssel_phase) determine when frequency and phase offset changes occur. For example, generating a frequency sync after programming the two frequency words causes the NCO (or multiple NCOs) to change frequency at that time, rather than after each of the two frequency words are programmed over the control bus. Note that the frequency and phase words are not loaded into the working register until their respective sync's are received. The zero phase sync signal (ssel_nco) is used to force the sine and cosine oscillators to their zero phase state. Note that this is an instantaneous phase jump, so the ssel_nco should only be issued when resetting a channel. Dither sync (ssel_dither) can be used to synchronize the dither generators of multiple NCOs. This is normally only required for applications that are performing bit match testing. The NCOs used in the transmit section are identical to what is described for the receive section. Note that there is one set of sync's provided for each DDC. When one DDC is used to process two CDMA signal the sync's are shared between them.

Table 11. Programming

VARIABLE	DESCRIPTION
ssel_nco(2:0)	Sync source for NCO accumulator reset
ssel_dither(2:0)	Sync source for NCO dither reset
ssel_freq(2:0)	Sync source for NCO frequency register loading
ssel_phase(2:0)	Sync source for NCO phase register loading

2.3.3 Receive Filtering and Decimation

The purpose of the receive filter chain is to isolate the signal of interest (and reject all other others) that has been previously translated to baseband via the mixer and NCO. The overall decimation through the chain also needs to be considered. The goal, generally, is to output the isolated signal at a rate that is twice (2X) the signal's chip rate. For UMTS, this would be 7.68 MSPS. For CDMA the output rate should be 2.4576 MSPS.

Receive filtering and decimation is performed in several stages:

- Zero padding to interpolate the input sample rate if needed up to the rxclk rate
- High rate decimation (4 to 32) using a six stage cascade-integrate comb filter (CIC)
- Decimate by two compensation filtering using the programmable compensating FIR filter (CFIR)
- Decimate by one pulse-shape filtering via the programmable FIR filter (PFIR)

Figure 11. DDC Filter Chain

The following table contains some examples listing the decimation and sample rates at the output of each block for UMTS and CDMA standards at input sample rates of 61.44 MSPS and 15.36 MSPS, assuming the GC5316 is clocked at 122.88 MHz.

Table 12. Example UMTS and CDMA2000 DDC Receive Modes

	INPUT SAMPLE RATE (MSPS)	ZEROS ADDED	ZERO PAD OUTPUT RATE (MSPS)	CIC DECIMATION	CIC OUTPUT RATE (MSPS)	CFIR DECIMATION	CFIR OUTPUT RATE (MSPS)	PFIR DECIMATION	PFIR OUTPUT RATE (MSPS)
UMTS	61.44	1	122.88	8	15.36	2	7.68	1	7.68
UMTS	15.36	7	122.88	8	15.36	2	7.68	1	7.68
CDMA	61.44	1	122.88	25	4.9152	2	2.4576	1	2.4576
CDMA	15.36	7	122.88	25	4.9152	2	2.4576	1	2.4576

2.3.4 Receive Channel Delay Adjust and Zero Insertion

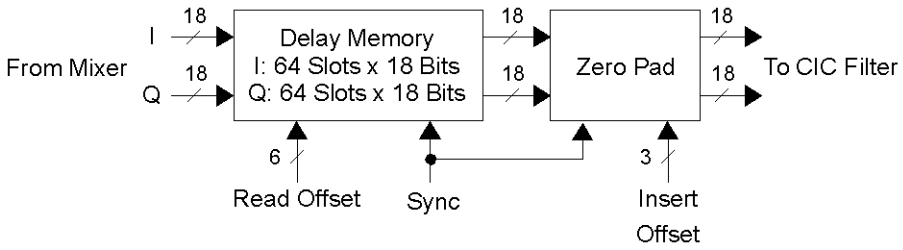


Figure 12. Delay Adjust and Zero Insertion

The receive channel delay adjust function is used to add programmable delays in the channel downconvert path. Adjusting channel delay can be used to compensate for analog elements external to the GC5316 digital downconversion such as cables, splitters, analog downconverters, filters, etc. There are two functions that need to be considered with respect to programming the channel delay; the delay memory and the zero pad blocks. The parameter tadj_interp_decim informs the DDC block the rate at which data is arriving on the rx_distribution bus. The zero pad block interpolates (insert zeros) to bring the signal sample rate up to rxclk rate.

The delay memory provides up to 64 sample delay at the rx_distribution rate. Read offset (tadj_offset_coarse) is a programmable difference between the read and write pointers to the delay memory. This provides a maximum differential delay between channels of 64/rx_distribution_rate. At an rx_distribution rate of 61.44 MSPS the 64 memory slots in the delay memory provide an overall delay window of about 1 µs. The ssel_taj_coarse sync controls the timing for updating the coarse offset.

The zero pad block inserts 0, 1, 3, or 7 zeros between each sample coming from the mixer bringing the sample rate up to rxclk. The tadj_offset_fine parameter specifies when the zeros are inserted relative to the ssel_tadj_fine sync signal. This permits a fine adjustment at the rxclk rate. The 3-bit insert offset parameter allows the zeros to be inserted up to tadj_interp_decim (max 8) high-speed clocks after ssel_tadj_fine sync is asserted. This provides a time adjust resolution of 1/rxclk. For UMTS and assuming a GC5316 clock frequency of 122.88 MHz, the time resolution is 3.84 MCPS / 122.88 MSPS = 1/32 of a chip. For CDMA, the resolution is 1.2288 / 122.88 = 1/100 of a chip.

Table 13. Programming

VARIABLE	DESCRIPTION
tadj_offseta_coarse_a5:0)	Read offset into the 64 element memory for the A channel DDC. Note: When tadj_offset_coarse_a = 62, then the delay is –2. When tadj_offset_coarse_a = 63, then the delay is –1. For all other values, the resulting delay is equal to the value.
tadj_offset_coarse_b(5:0)	Read offset into the 64 element memory for the B channel DDC when in CDMA mode. Note: When tadj_offset_coarse_b = 62, then the delay is –2. When tadj_offset_coarse_b = 63, then the delay is –1. For all other values, the resulting delay is equal to the value.
tadj_offset_fine_a(2:0)	Controls the zero offset (fine adjust) for the A side of the DDC.
tadj_offset_fine_b(2:0)	Controls the zero offset (fine adjust) for the B side of the DDC when in CDMA mode.
tadj_interp_decim(2:0)	The interpolation value minus one. Valid interpolations are (1, 2, 4, or 8). Valid program values for this parameter are (0,1,3,or 7). Same for A and B channels when in CDMA mode.
ssel_tadj_fine(2:0)	Selects the sync source for the fine time adjust
ssel_tadj_coarse(2:0)	Selects the sync source for the coarse time delay adjust

2.3.5 Receive CIC Filter

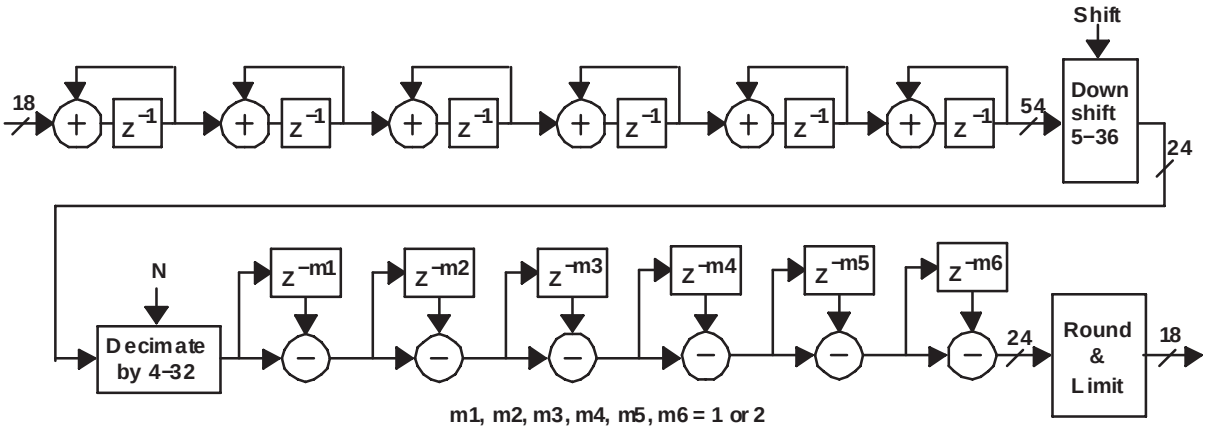


Figure 13. Six Stage CIC Filter

The CIC filter provides the first stage of filtering and large-value decimation. The filter consists of six stages and decimates over a range from 4 to 32.

I data and Q data are handled separately with two CIC filters. In addition, when in CDMA mode (two CDMA channels processed within a single DDC), another pair of CIC filters handles the B-side channel.

The filter response is $(Sin(x)/x)^6$ in character where the key attribute is that the resulting response nulls alias back to dc when the signal is decimated. The aliasing rejection achieved depends on the bandwidth of the signal of interest relative to the CIC output sample rate. A good rule of thumb is the signal of interest should be less than 25% of the CIC output rate. This means that the CIC decimation value should be chosen so that the signal exiting the CIC filter is oversampled by at least a factor of four. (Generally, it is close enough for digital signals that the CIC output rate be at least four times the symbol rate).

The filter is equivalent to six stages of a FIR filter with uniform coefficients (six combined boxcar filter stages). Each filter would be of length Ncic if m=1, or 2×Ncic if m=2.

The filter is made up of six banks of 54-bit accumulator sections followed by six banks of 24-bit subtractor sections. Each of the subtractor sections can be independently programmed with a differential delay of either one or two. A shift block follows the last integration stage and can shift the 54 bit accumulated data down by 36−cic_scale (a programmable factor from 0 to 31 bits).

The CIC filter exhibits a droop across its frequency response. This should be compensated in either the CFIR or PFIR filters that follow. Typically, droop compensation is done in the CFIR but it is also possible to compensate for CIC droop in the PFIR filter.

The gain of the receive CIC filter is: $Ncic^6 \times 2^{(number\ of\ stages\ where\ M=2)} \times 2^{(-36+CIC\_SCALE)}$ where CIC_SCALE is 0 to 31. There is no rollover protection internal to the CIC or at the final round so the user must guarantee no sample exceeds full scale prior to rounding. For practical purposes, this means the CIC gain must be less than or equal to one.

A fixed gain of 12 dB at the output of the CIC can also be programmed.

The post CIC gain is rollover protected. Post CIC gain = $2^{(cic\_gain\_ddc\ x\ 2)}$.

Table 14. Programming

VARIABLE	DESCRIPTION
cic_interp_decim(4:0)	The CIC decimation ratio (4 to 32) Ncic = cic_interp_decim + 1. This ratio applies to both A and B channels of the DDC block.
cic_scale_a(4:0)	The shift value for the A channel.
cic_scale_b(4:0)	The shift value for the B channel.
cic_gain_ddc	When asserted, adds a gain of 12 dB at the CIC output.
cic_m2_ena_a(5:0)	Sets the differential delay value M for each of the CIC subtractor stages for the A channel. Cic_m2_ena_a(0) controls m1 in the figure above.
cic_m2_ena_b (5:0)	Sets the differential delay value M for each of the CIC subtractor stages for the B channel.
cic_bypass	Test feature. Clear for normal operations.
ssel_cic(2:0)	Sets syncing (1 of 8 sources) for the CIC decimation moment.

2.3.6 Receive Compensation FIR Filter

The receive CFIR filter decimates the output of the CIC filter by a fixed factor of two. Filter coefficient size, input data size, and output data size are 18 bits. The CFIR length can be programmed. This permits *turning off* taps and saving power if shorter filters are appropriate. The CFIR power dissipation is proportional to its length. Exploiting symmetry of the coefficients by setting symmetric_cfir saves a small amount of power (but provides no additional available taps).

The maximum CFIR filter length is a function of GC5316 clock rate and output sample rate and is limited by the number of coefficient memory registers. The maximum number of taps is 64 and the minimum number is 14. Lengths between these limits can be specified in increments of 2.

Subject to the above minimum and maximum values, in the general case, the number of taps available is:

$$\text{UMT Mode : } 2 \times \frac{\text{rxclk}}{\text{output sample rate}}$$

CDMA Mode if cic_iterp_decim is even (decimating by an odd number) : $2 \times$ (cic_intrp_decim)

CDMA Mode if cic_iterp_decim is odd (decimating by an even number) : $2 \times$ (cic_intrp_decim + 1)

For CDMA, assuming the GC5316 is clocked at 122.8 MHz and with 2x oversampled output data (2.4576 MSPS), the CFIR filter length can range from 14 to 48 (not 50) in increments of 2.

For UMTS, at a GC5316 clock rate of 122.8 MHz with 2x oversampled output data (7.68 MSPS), the CFIR filter length can range from 14 to 32, in increments of 2.

A single set of programmed tap values are used for both the A-side and B-side DDC channels (two CDMA channels) within a single DDC block when in CDMA mode.

The CFIR filter performs the convolution, gain is applied at full precision, the signal is rounded, and then hard limited.

There is a shifter at the output of the filter the scales the data by either $2e^{-19}$ or $2e^{-18}$. The gain through the filter is therefore:

$$\text{Gain} = \text{Sum(CFIR coefficients)} \times 2^{-19+\text{cfir\_gain}} \text{ where cfir\_gain is 0 or 1}$$

Table 15. Programming

VARIABLE	DESCRIPTION
crastsrttap_cfir(4:0)	Number of DDC CFIR filter taps is 2(crastarttap + 1)
cfir_gain	Only ls-bit is used. CFIR gain. 0 = $2e^{-19}$, 1= $2e^{-18}$
	The CFIR filter's 18-bit coefficients are loaded in two 64 word memories. Zone 2 CFIR RAM holds the lower 2 bits of the 18 bit coefficients,. This address is specified by setting up the page register to write to Zone 2 and upper space bit Zp. The remaining bits are specified by ZZZZZ which are GC5316 address pins. The total address is thus ZpZZZZZ which writes to 64 locations.
	The CFIR filter's 18 bit coefficients are loaded by the software cmd5316. The user must provide a coefficient file with one integer coefficient per line. Note that the CFIR filter coefficients are shared by the A and B channels in CDMA mode.

2.3.7 Receive Programmable FIR Filter

The receive programmable FIR filter (PFIR) pulse shapes the baseband signal data. It does not perform any decimation. Filter coefficient size, input, and output data size is 18 bits. A special strapped mode can be employed for UMTS where two adjacent DDCs (2k and 2k+1, k=0 to 5) can be combined to yield a filter with twice the number of coefficients.

The PFIR length is programmable. This permits turning off taps and saving power if short filters are appropriate. The filter's output data can be shifted over a range of 0 to 7 bits where it is then rounded and hard limited to 18 bits. The shift range results in a gain that ranges from $2e^{-19}$ to $2e^{-12}$.

The gain of the PFIR block is:

$$\text{Gain} = \text{Sum(CFIR coefficients)} \times 2^{-19+\text{pfir\_gain}} \text{ where pfir\_gain ranges 0 7}$$

The maximum PFIR filter length is a function of GC5316 clock rate and output sample rate and is limited by the number of coefficient memory registers. The maximum number of taps is 64 and the minimum number is 28 (UMTS) or 32 (CDMA). Lengths between these limits can be specified in increments of 4.

Subject to the above minimum and maximum values, the number of maximum taps available is:

$$\text{UMTS Mode}: \ 4 \times \frac{\text{rxclk}}{\text{output sample rate}}$$

$$\text{Strapped Mode}: \ 8 \times \frac{\text{rxclk}}{\text{output sample rate}}$$

$$\text{CDMA Mode}: \ 2 \times \frac{\text{rxclk}}{\text{output sample rate}}$$

The strapped mode can be employed for UMTS where two adjacent DDCs (2k and 2k+1, k=0 to 5) can be combined to yield a filter with twice the number of coefficients. This means the GC5316 can support six UMTS DDC channels with double-length filter coefficients. Figure 14 shows the interconnect between the two DDCs when the PFIR filters are strapped. In strapped mode, data out of the last PFIR data delay ram in the main DDC (DDC 2k) is sent to the adjacent secondary DDC (DDC 2k+1) PFIR as input thus forming a 128-tap delay line. Also, data received from the adjacent PFIR summers is added into the main DDC's PFIR sum to form the output. When using strapped mode, set double_tap to 2 for the main (even) DDC and to 1 for the secondary (odd) DDC.

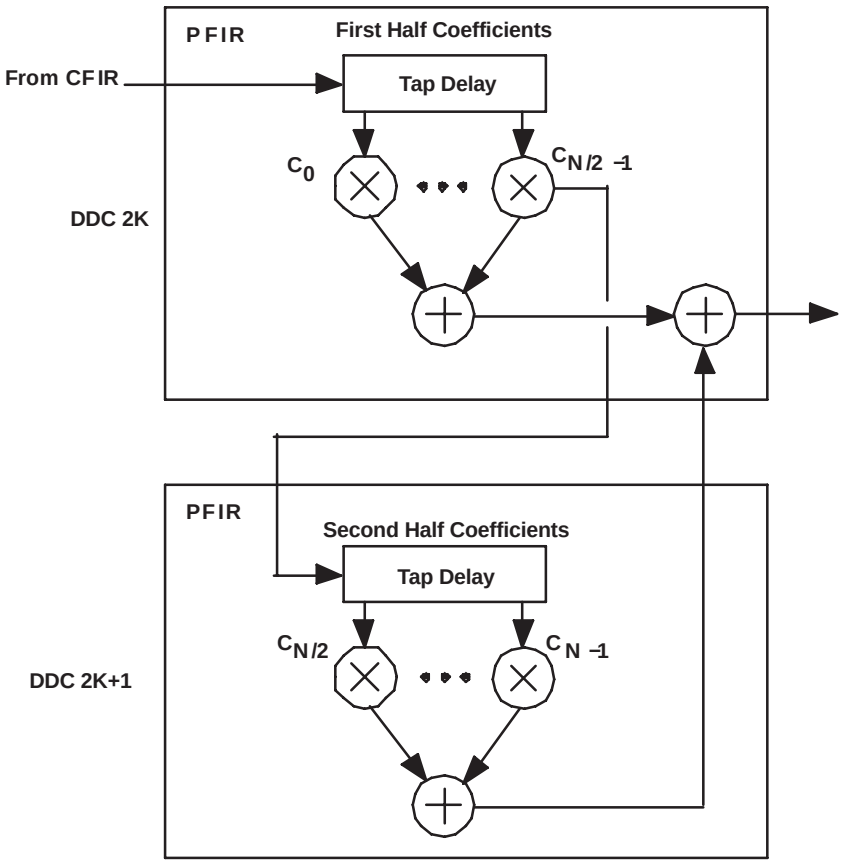


Figure 14. Double Tap Mode Interconnect

When in double-tap mode, the first half of the coefficients should be loaded into the even DDC, the remaining coefficients go into the odd DDC. The even DDC must be turned on (ddc_duc_ena 1), and the odd DDC must be turned off (ddc_duc_ena 0).

For strapped UMTS with double length filters, the range of taps available is 56 to 128 in increments of eight.

PFIR coefficients and gain shift values are shared between both A and B CDMA channels in a DDC block. The number of maximum taps available for double length UMTS mode is:

Double Length UMTS Mode: 8 x (rxclk ÷ output sample rate)

Table 16. Programming

VARIABLE	DESCRIPTION
crastsrttap_pfir(3:0)	Number of DDC PFIR filter taps is 4(crastartap+1) For double-length PFIR the number of taps is 8(crastartap+1)
pfir_gain(2:0)	Sets the gain of the PFIR filter.
double_tap	When set, puts two adjacent DDC (2k and 2k+1, k = 0 to 5) in 127 tap UMTS mode. Set to 0 for normal mode. Set to 2 for the main (even) DDC Set to 1 for the secondary (odd) DDC. When in double tap mode, the first half of the coefficients should be loaded into the even DDC, the remaining coefficients go into the odd DDC. ALSO: In double tap mode, the even DDC must be turned on (ddc_duc_ena 1), and the odd DDC must be turned off (ddc_duc_ena 0).
	The PFIR filter's 18-one integer coefficient per line. Note that the PFIR filter coefficients are shared by the A and B channels in CDMA mode.
	Note: that the above PFIR filter coefficients are shared between both A and B sides of a DDC block.

2.4 Receive RMS Power Meter

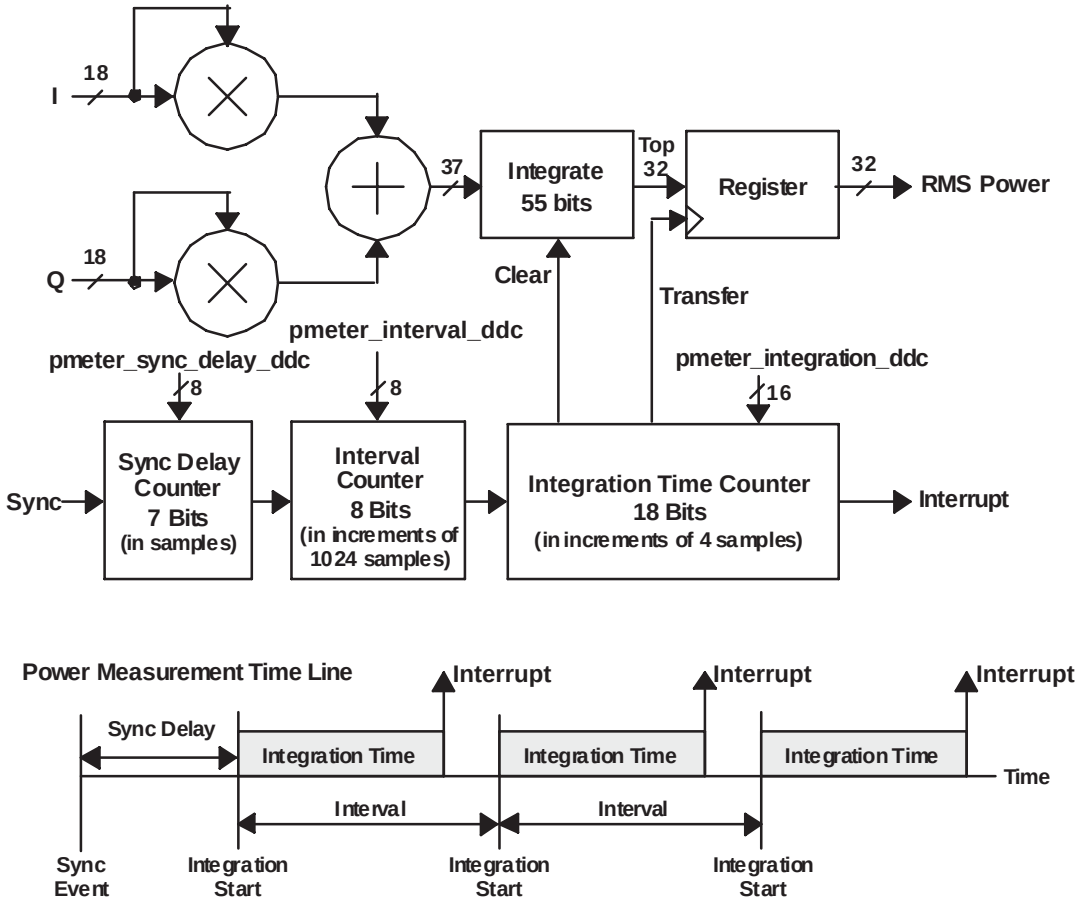


Figure 15. Receive Power Meter and Timing

Each DDC channel includes an RMS power meter which is used to measure the total power within the channel passband.

The power meter samples the I and Q data stream after the PFIR filter. Both 18-bit I and Q data are squared, summed, and then integrated over a time determined by a programmable counter: pmeter_integration_ddc (16 bits). The integration time is a 16-bit word which is programmed into the 18-bit counter. Integration time = 4 x pmeter_integration_ddc + 1 (in units of a sample period or generally chip period/2).

There is a programmable 8-bit interval counter which sets the interval over which power measurements are repeated. The timer counts in increments of 1024 samples. This allows the user to select intervals from 1 x 1024 samples up to 256 x 1024 samples. The interval time = 1024 x pmeter_interval_ddc. The interval time must be greater than (not equal to) the integration time.

The power measurement process starts with a sync event (ssel_pmeter). The integration starts at sync event + 3 chips + sync_delay. The 7-bit delay register permits delays from 3 to 130 samples after sync. The integration continues until the integration count is met. At that point the top 32 bits of the 55-bit accumulator is transferred to the read register and an interrupt is generated indicating the power value is ready to read. The interval counter continues until the programmed interval count is reached. When reached, the integration counter and the interval counter start over again. Each time the integration count is reached the upper 32 bits are again transferred to the read register overwriting the previous value and sending an interrupt signifying the data is ready to be read. Failure to read the data timely results in overwriting the previous interval measurement.

Sync ssel_pmeter starts the process. Whenever a sync is received, all the counters are reset to zero no matter what the status.

For UMTS, I and Q are calculated and the integrated power is read. When in CDMA mode the power is calculated for both the A and B signals, producing two 32-bit results.

For CDMA mode, the integration time is slightly longer. The power read in CDMA mode with a dc input is:

A power: $[I^2(X \times 4 + 1) + Q^2(X \times 4 + 0)]2^{-23}$ Note that one Q sample is missing from the integration.

B power: $[I^2(X \times 4 + 1) + Q^2(X \times 4 + 1)]2^{-23}$

Where X is the integration count.

Table 17. Programming

FIELD	DESCRIPTION
pmeter_result_a_lsb(15:0)	Lower 16 bits of the A DDC channel power measurement result.
pmeter_result_a_msb (31:16)	Upper 16 bits of the A DDC channel power measurement result.
pmeter_result_b_lsb (15:0)	Lower 16 bits of the B DDC channel power measurement result. Only available for CDMA.
pmeter_result_b_msb (31:16)	Upper 16 bits of the B DDC channel power measurement result. Only available for CDMA.
pmeter_integration_ddc(15:0)	Integration time = 4(1+pmeter_count_ddc).
pmeter_sync_delay_ddc(7:0)	Start delay from sync = 3 + pmeter_sync_delay_ddc.
pmeter_interval_ddc(7:0)	Interval time = 1024(pmeter_interval_ddc+1). Interval time must be greater than (not equal) integration time.
ssel_pmeter(2:0)	Sync source options
pmeter_sync_disable	Turns off the sync to the channel power meter. This can be used to individually turn off syncs to a channels power meter, while still having syncs to other power meters on the chip.

2.5 Receive Gain and AGC

The receive AGC can be used as a simple gain or as a flexible AGC.

2.5.1 Receive Simple Gain

The receive AGC can be used as simple gain by freezing the AGC accumulator at its current level, then clearing the current value. This is done by setting the agc_clear and agc_freeze bits. The output can be rounded from 3 to 18 bits using agc_rnd or the full 25 bits can be output by setting agc_rnd_disable.

2.5.2 Receive AGC

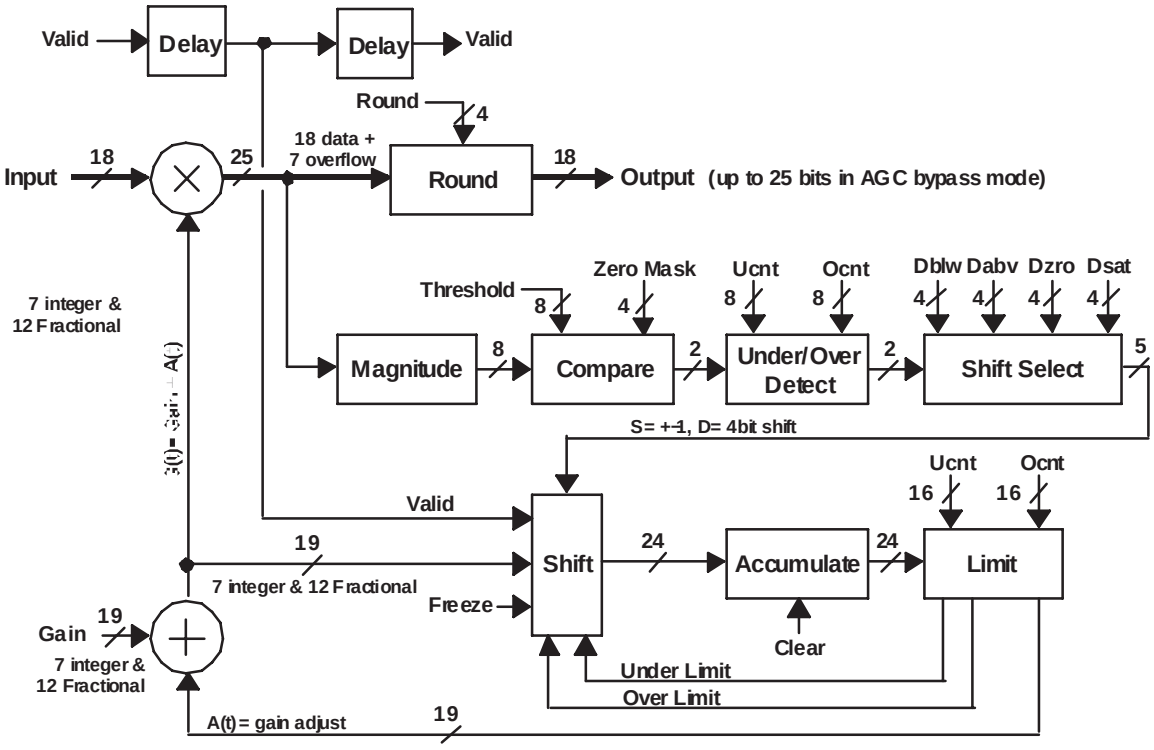


Figure 16. AGC Block Diagram

The GC5316 automatic gain control circuit is shown above. The basic operation of the circuit is to multiply the 18-bit input data from the PFIR by a 19-bit gain word that represents a gain or attenuation in the range of 0 to 128. The gain format is mixed integer and fraction. The 7-bit integer allows the gain to be boosted by up to factor of 128 (42 dB). The 12-bit fractional part allows the gain to be adjusted up or down in steps of one part in 4096, or approximately 0.002 dB. If the integer portion is zero, then the circuit attenuates the signal. The gain adjusted output data is saturated to full scale and then rounded to between 3 and 18 bits in steps of one bit or the full 25 bits may be output by setting agc_rnd_disable.

The AGC portion of the circuit is used to automatically adjust the gain so that the *median* magnitude of the output data matches a target value, which is performed by comparing the magnitude of the output data with a target threshold. If the magnitude is greater than the threshold, then the gain is decreased, otherwise it is increased. The gain is adjusted as: G(t) = G + A(t), where G is the default, user supplied gain value, and A(t) is the time varying adjustment. A(t) is updated as $A(t) = A(t) + G(t) \times S \times 2^{-D}$, where S=1 if the magnitude is less than the threshold and is −1 if the magnitude exceeds the threshold, and where D sets the adjustment step size. Note that the adjustment is a fraction of the current gain. This is designed to set the AGC noise level to a known and acceptable level while keeping the AGC convergence and tracking rate constant, independent of the gain level. The signal to AGC noise ratio will be equal to 6×D dB, so for noise purposes, D should be set to 5 or more to preserve an SNR > 30 dB, while typical CDMA or UMTS applications set D considerably higher (longer AGC time constant). The time constant is how long it takes the AGC to converge to within 63% of a required gain change. (It takes four time constants to converge to within 98% of the change.)

If one assumes the data is random with a Gaussian distribution, which is valid for UMTS if more than 12 users with different codes have been overlaid, then the relationship between the RMS level and the median is MEDIAN = 0.6745×RMS, hence the threshold should be set to 0.6745 times the desired RMS level.

The gain step size can be set using four different values of *D*, each of which is a 4-bit integer. D can range from 3 to 18. The user can specify values of D for different situations, i.e., when the signal magnitude is below the user-specified threshold (Dblw), is above the threshold (Dabv), is consistently equal to zero (Dzro) or is consistently equal to maximum (Dsat). It is important to note that D represents a gain step size. Smaller values of D represent larger gain steps. The definition of *equal to zero* is any number when masked by zero_mask is considered to be zero. This permits consistently very small amplitude signals to have there gain be increased rapidly.

The different D values allows the user to set different attack and decay time constants when the signal is in a useful working range. When the output signal so weak or so strong that no useful information remains there is no concern about preserving signal quality and the desire is to move the signal rapidly into a useful working range. The magnitude is considered to be uselessly weak by using a 4-bit counter that counts up every time the masked 8-bit magnitude value is zero, and counts down otherwise. If the counter's value exceeds a user specified threshold, then *Dzro* is used. Similarly the magnitude is considered uselessly strong by using a counter that counts up when the magnitude is maximum, and counts down otherwise. If this counter exceeds another user specified threshold, then *Dsat* is used.

As an example using a dc-signal input, if the AGC's current gain at a particular moment in time is 5.123, and the magnitude of the output signal is greater than zero, but less than the user-programmed threshold. Step size Dblw will be used to increase the gain for the next sample. This represents the AGC attack profile. If Dblw is set to a value of 5, then the gain for the next sample will be $5.123 + 5.123 \times 2^{-5} = 5.283$. If the output signal's magnitude is still less than the user-programmed threshold, then the gain for the next sample will be $5.283 + 5.283 \times 2^{-5} = 5.448$. This continues until the output signal's magnitude exceeds the user-programmed threshold. When the magnitude exceeds threshold (but is not saturated), then step size Dabv is automatically employed as a size rather than Dblw.

The AGC converges linearly in dB with a step size of $40\log(1+2^{-D})$ when the error is greater than 12 dB (i.e., the gain is off by 12 dB or more). Within 6 dB, the behavior is approximately an exponential decay with a time constant of $2^{(D+0.5)}$ samples.

The suggested value of D is 5 or 6, when the error is greater than 12 dB (i.e., in the fast range detected by consistently zero or saturated data). This gives a step size of 0.5 dB or 0.25 dB per sample.

The suggested value when the gain is off by less than 12 dB is D=10, giving a exponential time constant for delay of around 1722 samples (63% decay every 1722 samples).

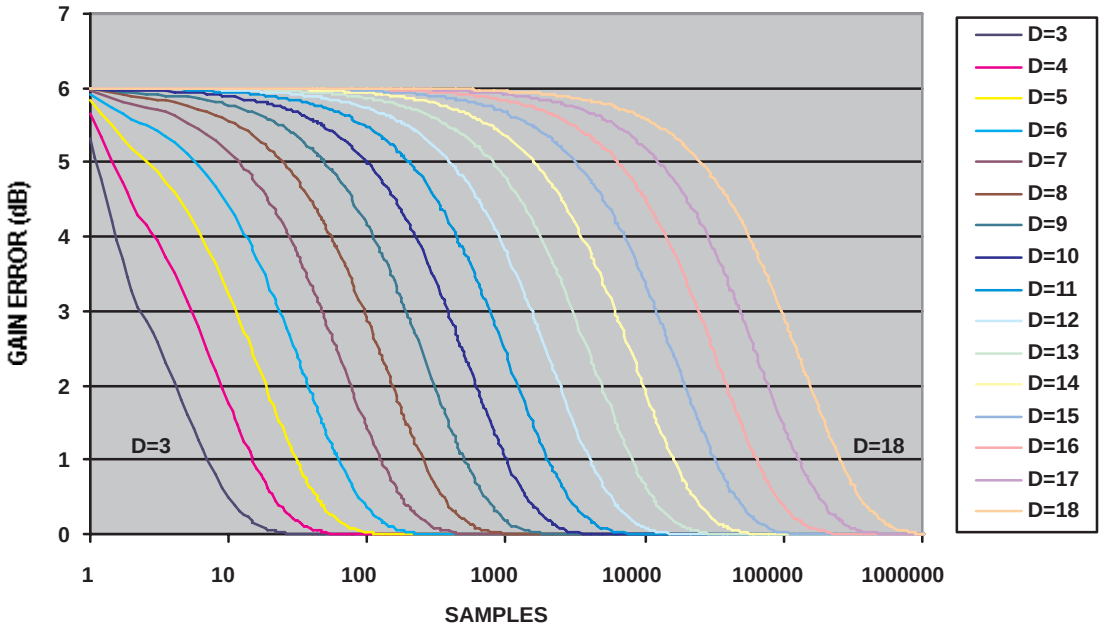


Figure 17. AGC Gain Error vs Samples

The AGC noise once the AGC has converged is a random error of amplitude $\pm 2^{-D}$ relative to the RMS signal level. This means that the error level is −6×D dB below the signal RMS level. At D=10 (−60 dB) the error is negligible. The plot below shows the AGC response for vales of D ranging from 3 to 18. *Error dB* represents the distance the signal level is from the desired target threshold.

The AGC is also subject to user specified upper and lower adjustment limits. The AGC stops incrementing the gain if the adjustment exceeds Amax. It stops decrementing the gain if the adjustment is less than Amin.

The input data is received with a valid flag that is high when a valid sample is received. For complex data, the I and Q samples are on the same data input line and are not treated independently. An adjustment is made for the magnitude of the I sample, and then another adjustment is made for the Q sample.

The AGC operates on UMTS and CDMA data. When in UMTS mode the I and Q data are each used to produce the AGC level. There is no separate I path gain and Q path gain. When in CDMA mode there are separate gain levels for the signal and diversity I and Q data. The I and Q for A (or the Signal) pair is calculated and then the I and Q for the B (or diversity) pair is calculated.

There is a freeze mode for holding the accumulator at its current level. This puts the AGC in a hold mode using the user-programmed gain along with the current gain_adjust value. To only use the user programmed gain value as the gain, set the freeze bit and then clear the accumulator. When using the freeze bit, the full 25-bit output is sent out of the AGC block to support transferring up to 25 bits when the AGC is disabled.

The current AGC gain and state can also be optionally output with the DDCs I and Q output data by setting the gain_mon variable. When in this mode, the top 14 bits of the current AGC gain word are integrated in with the AGC-modified I and Q output data.

Table 18. Output Data Format With Embedded AGC Gain Data

Output	Bits(17:10)	Bits(9:4)	Bits(3:2)	Bits(1:0)
I	I output data	Gain(18:11)		00
Q	Q output data	Gain(10:5)	AGC State(1:0)	00

Table 19. Programming

VARIABLE	DESCRIPTION
agc_dbelow(3:0)	Sets the value of gain step size Dblw (data × current gain below threshold). Dblw = 3 + agc_dbelow. agc_dbelow ranges from 0 to 15.
agc_dabove(3:0)	Sets the value of gain step size Dabv (data × current gain above threshold). Dabv = 3 + agc_ dabove. agc_ dabove ranges from 0 to 15.
agc_dzero(3:0)	Sets the value of gain step size Dzro (data × current gain consistently zero). Dzro = 3 + agc_ dzero. agc_ dzero ranges from 0 to 15.
agc_dsat(3:0)	Sets the value of gain step size Dsat (data × current gain consistently saturated). Dsat = 3 + agc_ dsat. agc_ dsat ranges from 0 to 15.
agc_zero_msk(3:0)	Masks the lower 4 bits of signal data so as to be considered zeros.
agc_thres(7:0)	AGC threshold. Compared with magnitude of 8 bits of input × gain.
agc_gaina_lsb(15:0)	Lower 16 bits of 19-bit gain word for DDC A. Requires a sync (ssel_gain) to load.
agc_gaina_msb(18:16)	Upper 3 bits of 19-bit gain word for DDC A. Requires a sync (ssel_gain) to load.
agc_gainb_lsb(15:0)	Lower 16 bits of 19-bit gain word for DDC B (in CDMA mode). Requires a sync (ssel_gain) to load.
agc_gainb_msb(18:16)	Upper 3 bits of 19-bit gain word for DDC B (in CDMA mode). Requires a sync (ssel_gain) to load.
ssel_gain(2:0)	Sync to update agc_gain settings. Note that both A and B are updated.
agc_zero_cnt	When the AGC output (input × gain) masked magnitude is zero value this number of times, the shift value is changed to agc_dzero.
agc_max_cnt	When the AGC output (input × gain) is zero value this number of times, the shift value is changed to agc_dsat.
agc_md(3:0)	AGC rounding. Number of output bits = 18 – agc_rnd.
agc_rnd_disable	AGC rounding is disabled when this bit is set.
agc_freeze	Freezes the adaptive portion of the gain to current value.
agc_clear	Clears the adaptive portion of the gain.
agc_amax(15:0)	The maximum value that gain can be adjusted up to. Top 7 bits are integer, bottom 9 bits are fractional.
agc_amin(15:0)	The minimum value that gain can be adjusted down to. Top 7 bits are integer, bottom 9 bits are fractional.
gain_mon	When set, combines current AGC gain with I and Q data. The 18-bit output format thus becomes: I Portion: 8 bits of AGC'd I data − Gain(18:11) − 00 Q Portion: 8 bits of AGC'd Q data − Gain(10:5) − Status(1:0) − 00. Note: Bit 0 of status, when set, indicates the data is saturated. Bit 1 of status, when set, indicates the data is zero.

2.6 Receive Output Interface

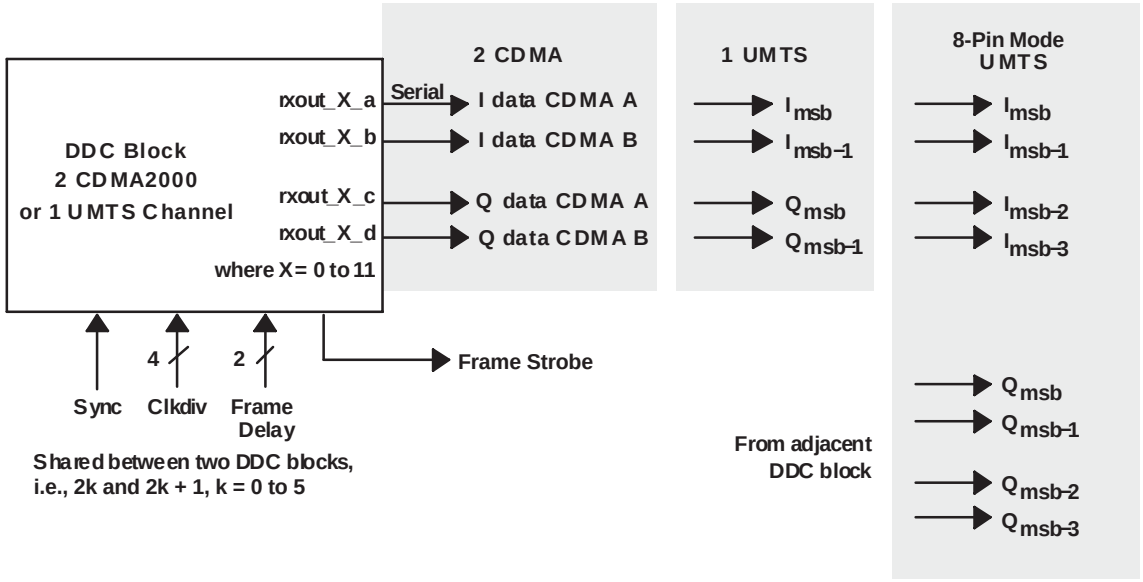


Figure 18. Receive Output Interface

Each DDC block has four serial output data pins. These pins are used to transfer downconverted I/Q baseband data out of the GC5316 for subsequent processing. The usage of these pins changes depending on how the DDC block is configured.

When the block is configured for two CDMA channels, a pair of serial data pins provides separate I and Q data output for the two DDC channels. Word size is selectable from 4 to 25 bits with the most significant bit first. **Note, carefully the signal to pin assignment, for example that Ia is assigned to rxout_X_a and Qa is assigned to rxout_X_c.**

When the DDC block is configured for a single UMTS channel, even and odd I and Q data drive the four serial pins separately, most significant bit first.

Four serial pins each for I and Q data can be optionally employed (instead of two for I and two for Q) at half the output rate. This would most likely be used when two DDC channels (2k and 2k + 1, k= 0 to 5) are combined to support double−length PFIR filtering (a channel is sacrificed). Formatting for I data is then: Imsb, Imsb−1, Imsb−2, Imsb−3. Q data formatting is: Qmsb, Qmsb−1, Qmsb−2, Qmsb−3.

Two DDC blocks share a frame strobe output pin. The frame strobe is driven high when the channel outputs another frame of data. The frame strobe can be programmed to arrive from 0 to 3 bit clocks early via a 2-bit control parameter. Frame interval can be programmed from 1 to 63 bits**.** A programmable 4-bit clock divider circuit is can be used to specify the serial bit rate. The clock divider circuit is synchronized using a sync block discussed later in this document.

Programming the serial port clock divider requires some thought and depends upon the channel's overall decimation ratio, frame sync interval, number of output bits, and CDMA−UMTS mode.

In general:

The serial clock divide ratio × the frame sync interval = the total receive decimation

The relationship between the number of serial bits output, clock divide ratio, and overall decimation ratio is:

$$\text{CDMA}: \frac{[\text{overall decimation} \times (\text{pser\_rec\_8pin} + 1)]}{(\text{pser\_recv\_clkdiv} + 1)} > \text{pser\_recv\_bits} + 1$$

$$\text{UMTS}: 2 \times \frac{[\text{overall decimation} \times (\text{pser\_rec\_8pin} + 1)]}{(\text{pser\_recv\_clkdiv} + 1)} > \text{pser\_recv\_bits} + 1$$

Figure 19 shows the DDC serial output timing.

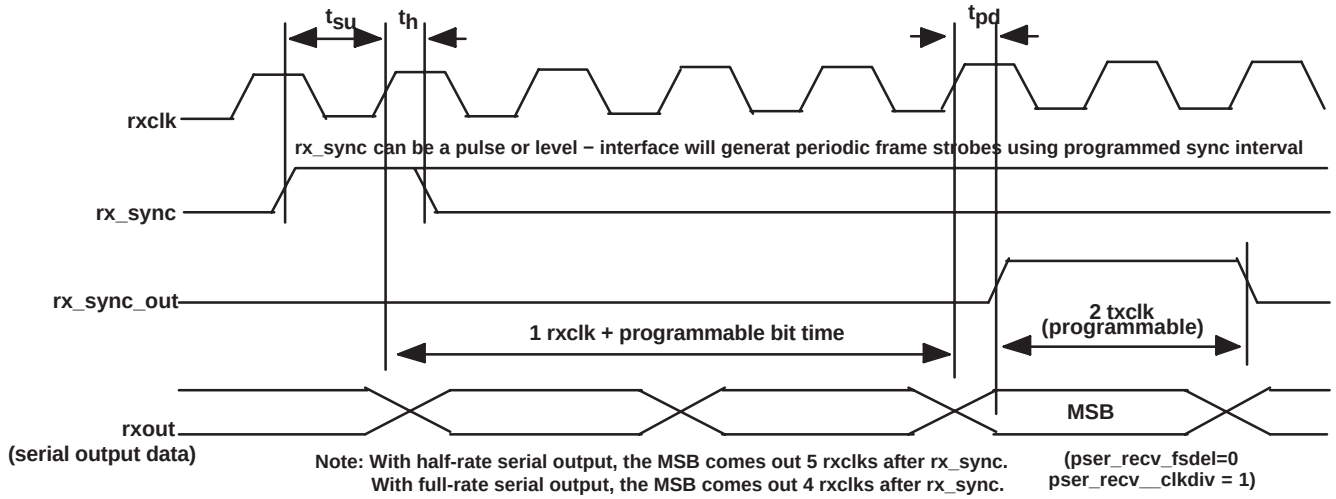


Figure 19. DDC Serial Output Timing

Table 20. Programming

VARIABLE	DESCRIPTION
pser_recv_fsinvl(6:0)	Frame sync interval in bits
pser_recv_bits(4:0)	Number of data output bits − 1. i.e.: 10001 = 18 bits
pser_recv_clkdiv(3:0)	Receive serial interface clock divider rate − 1. 0= rxclk, 15= rxclk/16
pser_recv_8pin	When set, configures the serial out pins for 4I and 4Q in UMTS mode. When clear, the mode is 2I and 2Q. Used in conjunction with pser_recv_alt.
pser_recv_alt	When set, outputs Q data from adjacent DDC channel.
pser_recv_fsdel(1:0)	Number of bit clocks the frame sync is output early with respect to serial data.
ssel_serial(2:0)	Sync source (1 of 8).

3 GC5316 Transmit

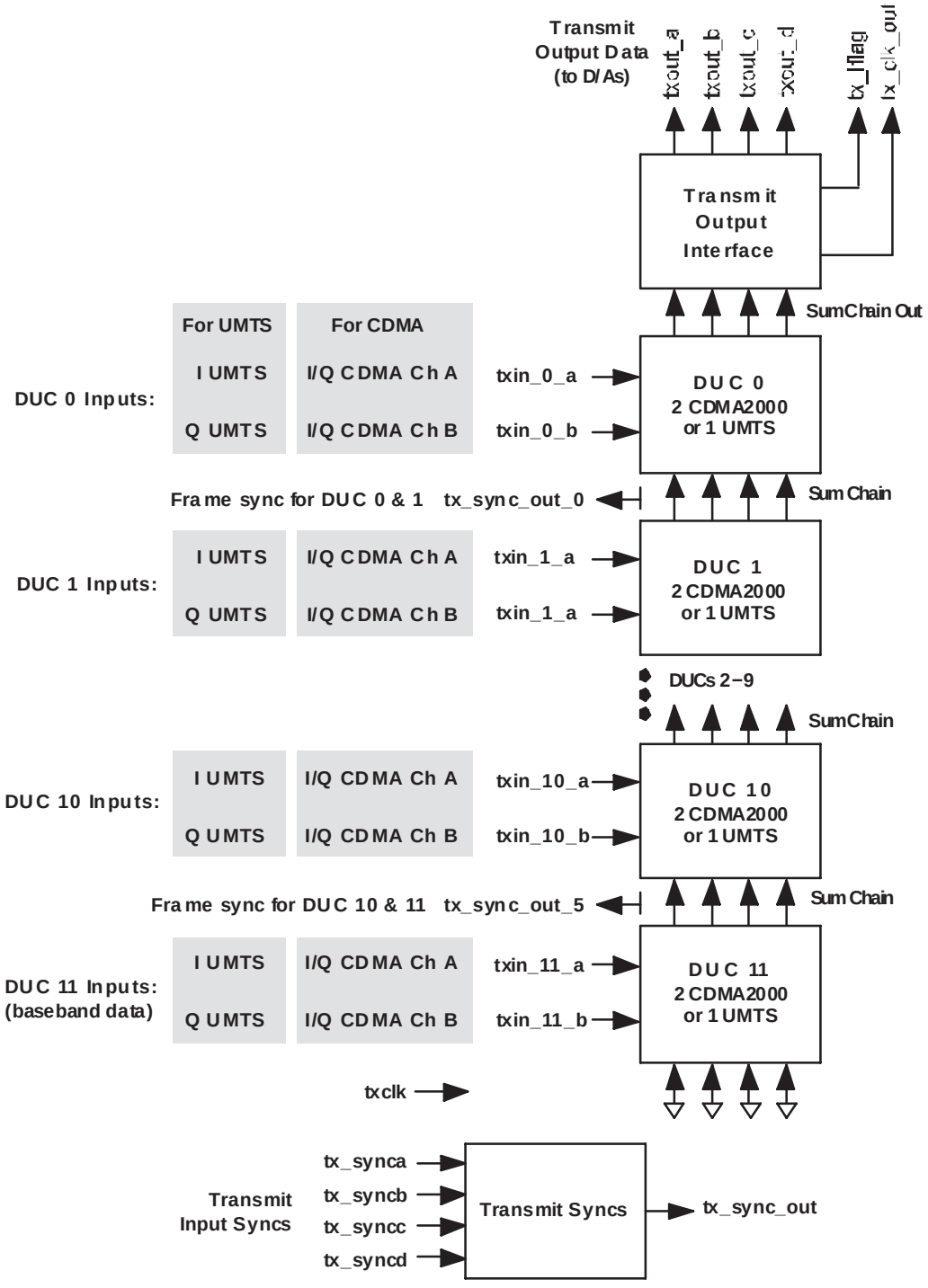


Figure 20. Transmit Section

The GC5316 transmit section provides up to 24 CDMA2000 or 12 UMTS digital upconversion (DUC) channels. There are 12 DUC blocks, DUC 0 through DUC 11. Each block can be configured as a single UMTS channel or two CDMA channels.

The outputs of all DUCs drive four independent complex sum chains. Any DUC can contribute (or not) to any or all of the four sum chains. The output of a DUC block's sum chain drives the sum chain input of the next block. The first DUC to output data is DUC0, while the last is DUC11. The four outputs of a DUC are the sum of all the contributing channels of all the higher numbered DUC blocks and itself. The sum chain inputs of DUC 11 are grounded. Within the chain, all DUC blocks from 0 up to the highest numbered DUC in use must be turned on otherwise the sum chain is broken.

The transmit output interface takes the four summed chains of DUC output data from the output of DUC 0 and then scales and rounds to a user-programmed number of bits. Composite power meters with programmable integration periods and intervals compute the power in each of the four output streams. The data is then formatted for output over the four tx_data_out outputs.

3.1 Digital Upconvert Block (DUC)

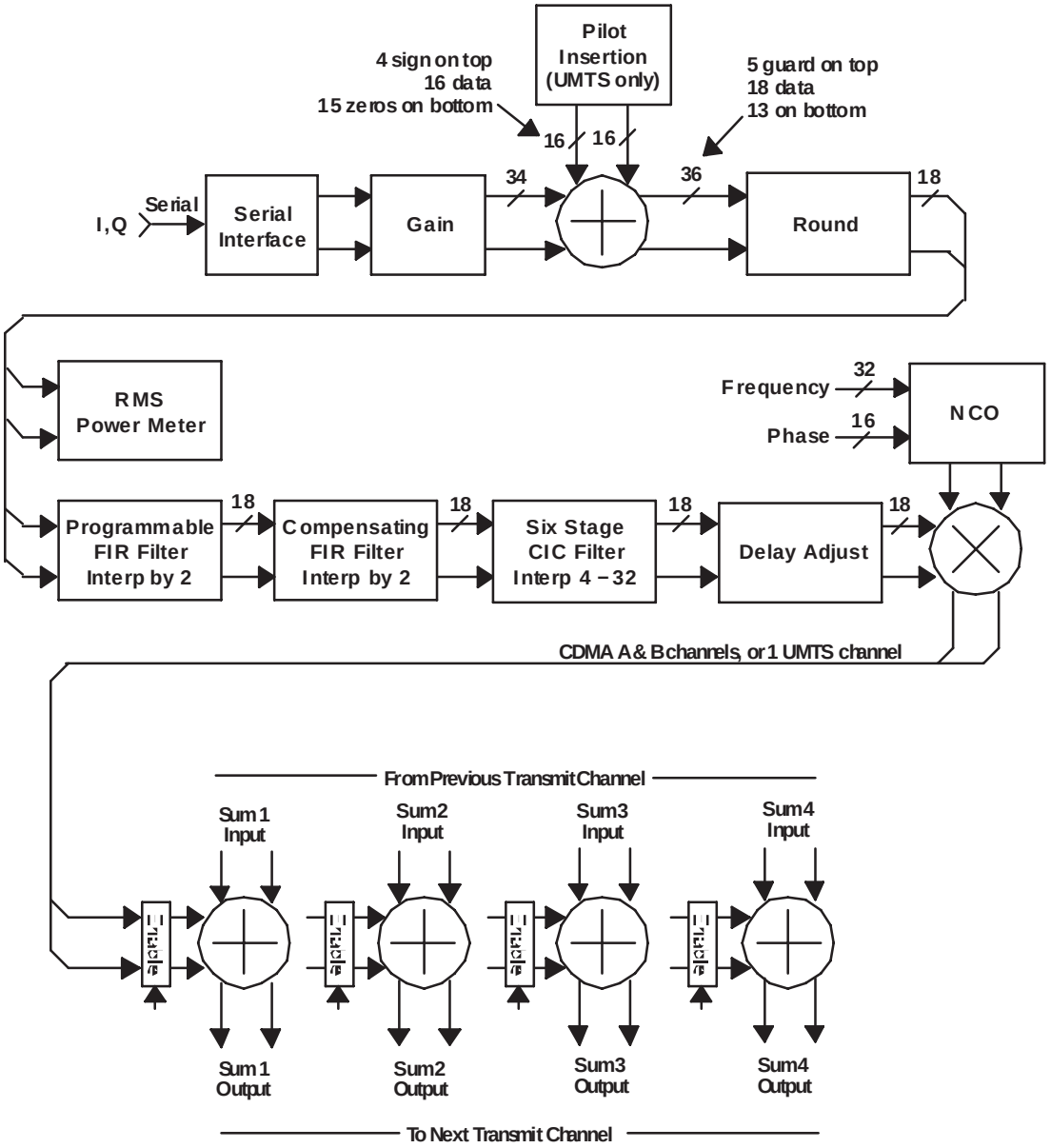


Figure 21. Digital Upconvert Block

This section describes the functions available in each of the 12 DUC blocks. Each DUC block has its own register set and may be programmed individually. The final output rates must match since they are added together.

The diagram above shows the different signal processing blocks and general signal processing flow of an individual transmit channel. Within a DUC block a single set of hardware performs these functions for one UMTS signal or two CDMA signals. When processing two CDMA signals the gain, round, power meter, PFIR and CFIR blocks are time shared to process both signals with one set of hardware. Each DUC can support one UMTS channel or two CDMA channels.

Each DUC block accepts baseband serial data. At this point the gain can be adjusted and a pilot sequence can be summed with the data. Power can be measured, and then the data is pulse-shape filtered and interpolated to a higher rate. The programmable FIR filter (PFIR) is used to pulse shape the data and interpolates by a factor of two. The compensating CIC filter (CFIR) compensates for the roll-off of the following CIC filter and also interpolates by a factor of two. The CIC filter performs additional interpolation which is programmable. The delay adjust block permits the channel's delay to be adjusted relative to all other DUC channels.

The interpolated, filtered, and delayed data is then tuned to a user-programmed frequency with a digital mixer and oscillator. The DUC's output data then drives four independent sum chain paths, where output data from each DUC can be summed into four composite streams.

Each function block is described in greater detail in subsequent sections.

Table 1. Programming

VARIABLE	DESCRIPTION
ddc_duc_ena	When set this turns on the DUC. When unset, the block is turned off.
cdma_mode	When set, the DUC block is in CDMA2000 mode.

3.1.1 Transmit Serial Input Interface

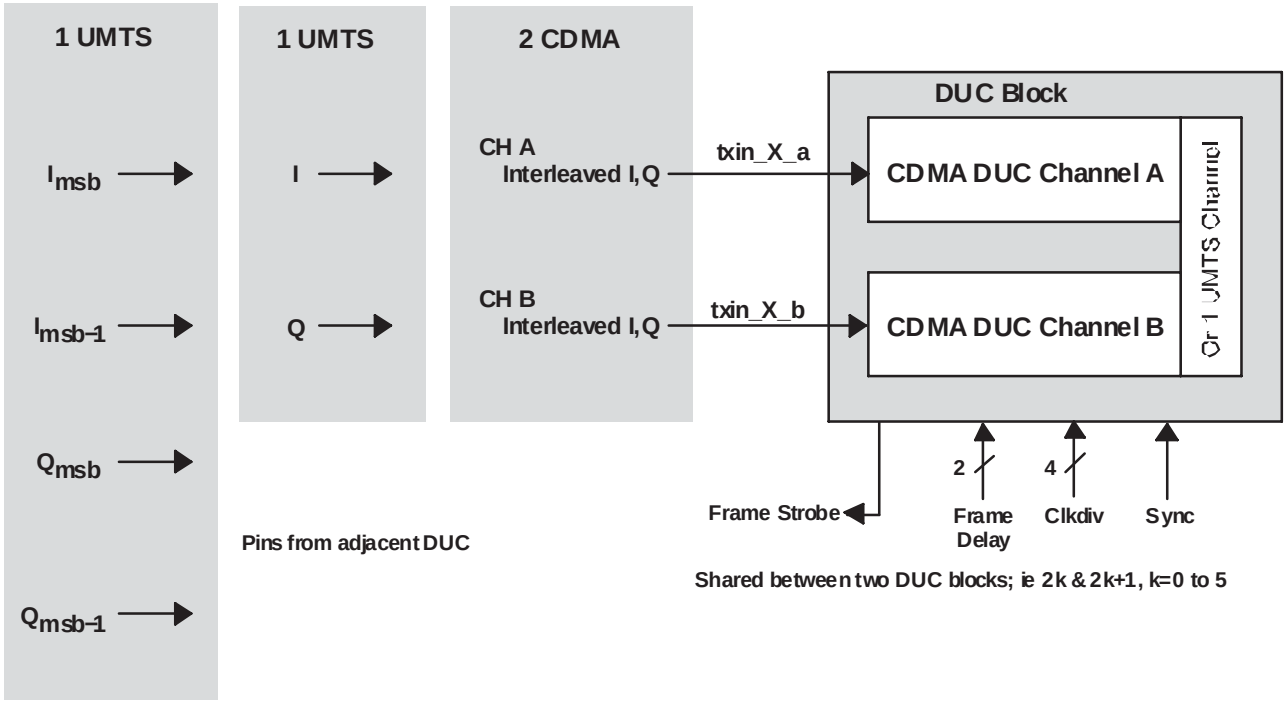


Figure 22. Transmit Serial Input Interface

Each DUC block has two serial input data pins. These pins are used to transfer I/Q baseband data into the DUC channel for interpolation, filtering, and tuning to a carrier frequency. How these pins are used depends on the channel configuration of the DUC block.

When the block is configured for two CDMA channels, one pin (txin_X_a) accepts serial data for signal A, the other pin (txin_X_b) for signal B. Input I and Q data, programmable up to 18 bits, is multiplexed over the serial input pin starting with the most significant I bit. The maximum input bit rate is txclk. The interface can be programmed to accept up to 32 bits, but only the upper 18 bits will be used as input signal data.

When the block is configured for a single UMTS channel, the txin_a is for I data, and txin_b carries the Q data. The most significant bit is sent first.

The four pin mode is a less common mode. It employs another two pins from the adjacent (2k+1) DUC, sacrificing the use of that DUC in order to allow reduced datarate on the serial pins. The I data (Imsb, Imsb−1) are carried on txin _(2k) _a and txin _(2k)_b, while the Q data (Qmsb Qmsb−1) is carried on txin _(2k+1) _a and txin _(2k+1) _b.

Each pair of DUC blocks 2k and 2k+1 share the clock division, frame delay, sync generation, and a frame strobe output pin.

A programmable clock divider circuit can be used to specify the serial bit rate with respect to txclk. The divider is programmed as txclk / (1+serp_trans_clkdiv). The clock divider circuit is synchronized using a general sync block discussed in another section of this document.

The frame sync interval can be programmed from 1 to 127 bits (which are divided clocks).

The number of bits in a word is set as (serp_tran_bits+1).

The frame strobe is an output from the gc5316 that indicates when the msb is expected. The frame strobe can be programmed to arrive from 0 to 3-bit clocks ahead of when the msb is expected via the serp_tran_fsdel parameter. The source must transmit all of its data before the next frame strobe is generated. Use of the frame strobe is optional in that when the msb is expected is determined by the sync (ssel_serial).

The parameter chosen must satisfy the following constraints:

- serp_tran_fsinv x (serp_tran_clkdiv+1) = 4 x (cic_interp_decim+1)
- serp_tran_fsinv >= (serp_tran_bits+1)2 for CDMA mode
- serp_tran_fsinv >= (serp_tran_bits+1) for UMTS mode
- serp_tran_fsinv >= (serp_tran_bits+1)0.5 for four−pin mode

NOTE: For half-rate data (when serp_tran_clkdiv= 1), the MSB of the input data stream is captured on the 4th rising edge of txclk, after txsync occurs. For full-rate data (when serp_tran_clkdiv= 0), the MSB of the input data stream is captured on the 3rd rising edge of txclk, after txsync occurs.

Figure 23 shows the transmit serial input timing.

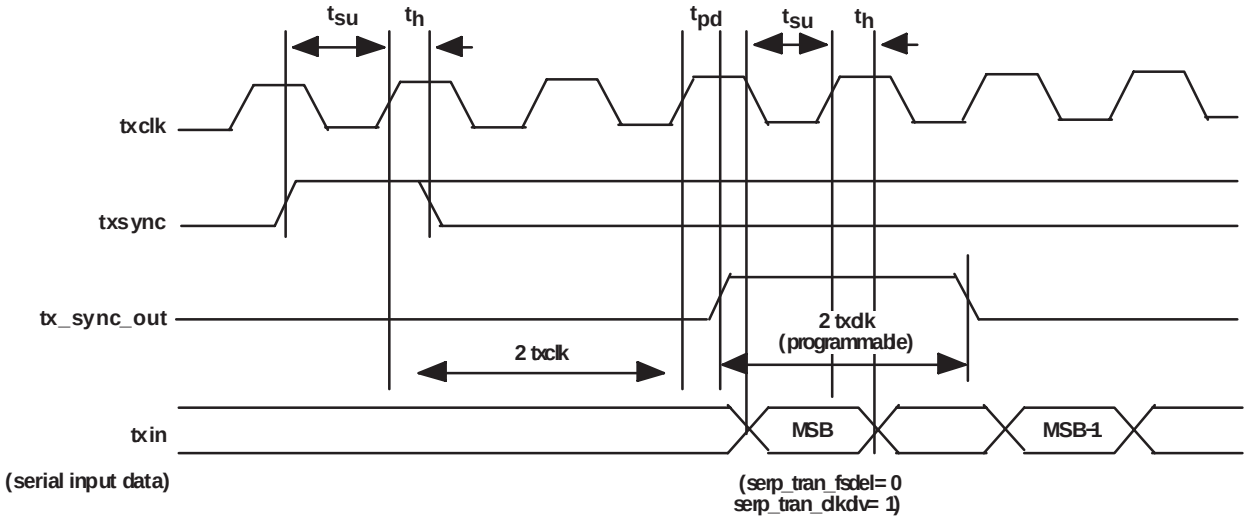


Figure 23. Transmit Serial Input Timing

Table 2. Programming

VARIABLE	DESCRIPTION
serp_tran_bits(4:0)	Number of serial input bits in a word – 1. i.e., 10001 = 18 bits
serp_tran_fsinvl(6:0)	Frame sync interval in bits
serp_tran_fsdel(1:0)	The number of serial bits after frame strobe that the data MSB is expected.
serp_tran_4pin	0= 2 pin input mode. Applies to UMTS mode for separate I and q data bits, as well as CDMA mode where one pin is for interleaved I/Q data for the CDMA A channel and another pin for interleaved I/Q data for the CDMA B channel. 1= 4 pin mode. Applies to UMTS mode where the channel has two bits for I data (I_{msb} and I_{msb-1}) and two bits for Q data (Q_{msb} and Q_{msb-1})
serp_tran_clkdiv(3:0)	Serial input data bit clock divider factor − 1
ssel_serial(2:0)	Sync source

The parameters are set for a pair of DUC blocks; i.e., for 2k and 2k+1 DUCs, where k= 0 to 5.

3.1.2 Transmit Gain

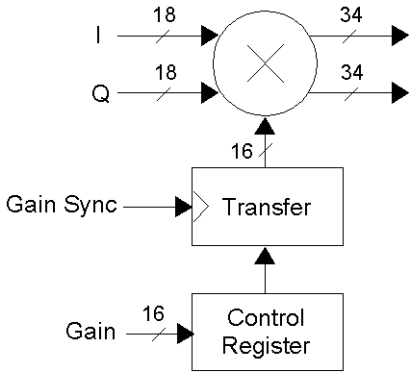


Figure 24. Transmit Gain Block

The transmit gain block is a multiplier that increases or decreases the level of the input data. The unsigned 16-bit gain word is interpreted with the binary point three bits down from the MSB. It multiplies the input data by (gain word/8192). The maximum gain is therefore 65535/8192. There are different gain registers for the A and B signals in CDMA mode.

A transfer register in combination with a sync (ssel_gain) is used to synchronize gain changes across multiple channels.

Table 3. *Programming*

VARIABLE	DESCRIPTION
gainfora(15:0)	Gain for the A-side DUC. Interpreted as gainfora/8192 and is unsigned.
gainforb(15:0)	Gain for the B-side DUC. Interpreted as gainforb/8192 and is unsigned.
ssel_gain(2:0)	Sync source

3.1.3 Transmit UMTS Pilot Code Insertion

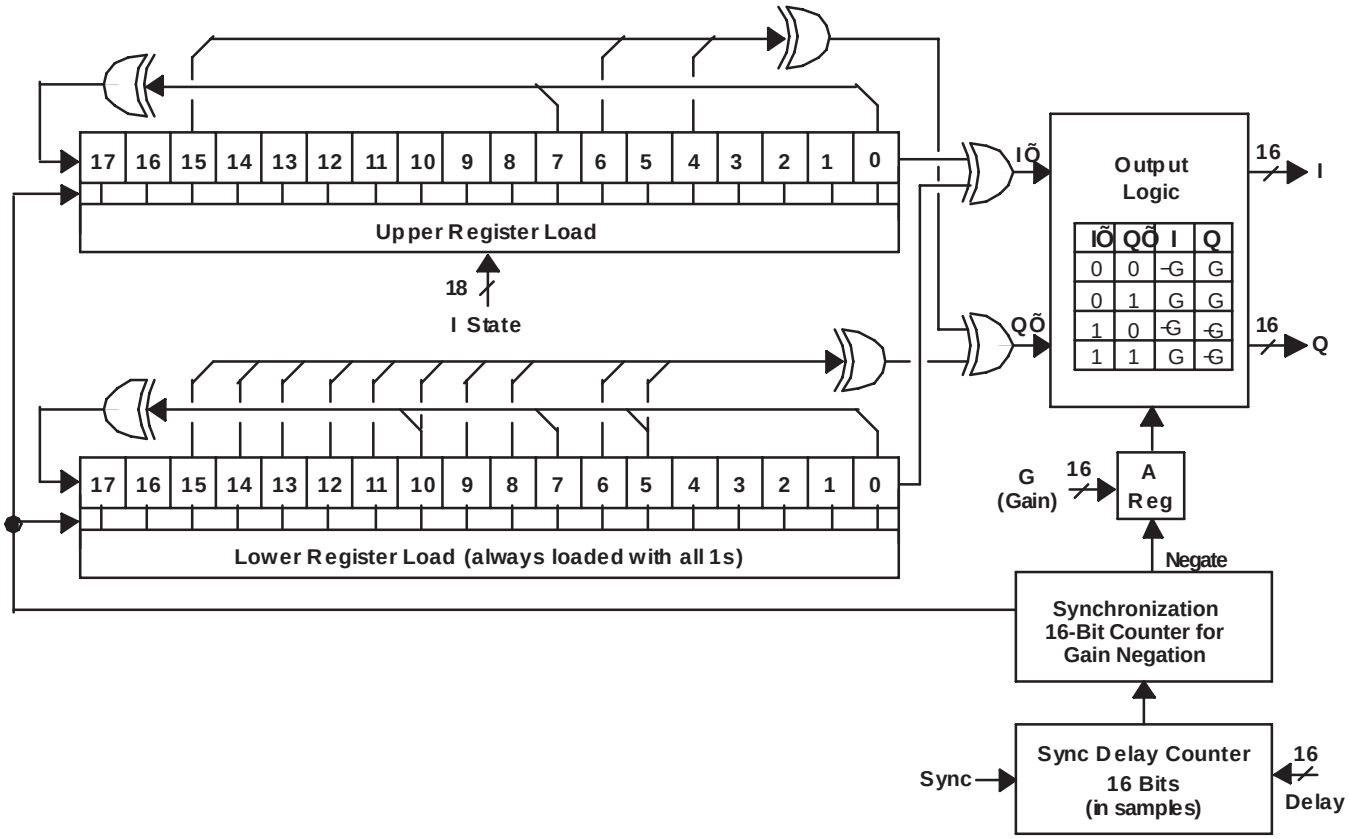


Figure 25. Pilot Code Insertion Logic

The pilot code insertion block is used to generate UMTS pilot scrambling sequences and does not apply when transmitting CDMA. The pilot sequence is summed with the UMTS input baseband data prior to PFIR filtering.

The sequence is complex and generated from two 18-bit shift registers, each with a unique set of feedback taps. Specific taps are exclusive/or combined to form the I and Q streams. The streams are then modified by a user-programmed complex gain value. The gain word G is a signed 16-bit value. The output sequence is +− G. Setting gain to zero turns off pilot insertion.

Note: Gain MUST be set to zero for CDMA operation.

The upper 18-bit shift register is programmed with a starting sequence based on the desired primary scrambling code (PSC). There are 512 start sequences for all of the BTS codes. The lower register is always started with a string of all 1s.

When diversity channels are employed, a counter in the synchronization block toggles the sign of the gain value in a prescribed fashion. The UMTS frame starts with positive gain for 256 chips, then toggles to negative gain for 512 chips, then toggles again to positive gain for 512 chips, etc. until the end of the frame. The last 256 chips of the frame will be negative gain. This sequence repeats for subsequent frames.

Table 4. Programming

VARIABLE	DESCRIPTION
pilot_psc(15:0)	Lower 16 bits of the 18-bit pilot LFSR initial sequence.
pilot_psc(17:16)	Upper 2 bits of the 18-bit pilot LFSR initial sequence.
pilot_diversity	Sets main or diversity pilot generation. 0 = main, 1 = diversity
pilot_delay(15:0)	Unsigned delay value (in chips) from sync event. 0 to 38399 chips.
pilot_gain_0(15:0)	Gain value. pilot_gain_0 and pilot_gain_1 must be set to the same value for proper operation. Must be set to 0 for CDMA operation.
pilot_gain_1(15:0)	Gain value. pilot_gain_0 and pilot_gain_1 must be set to the same value for proper operation. Must be set to 0 for CDMA operation.
ssel_pilot(2:0)	Sync source

3.1.4 Transmit Channel RMS Power Meter

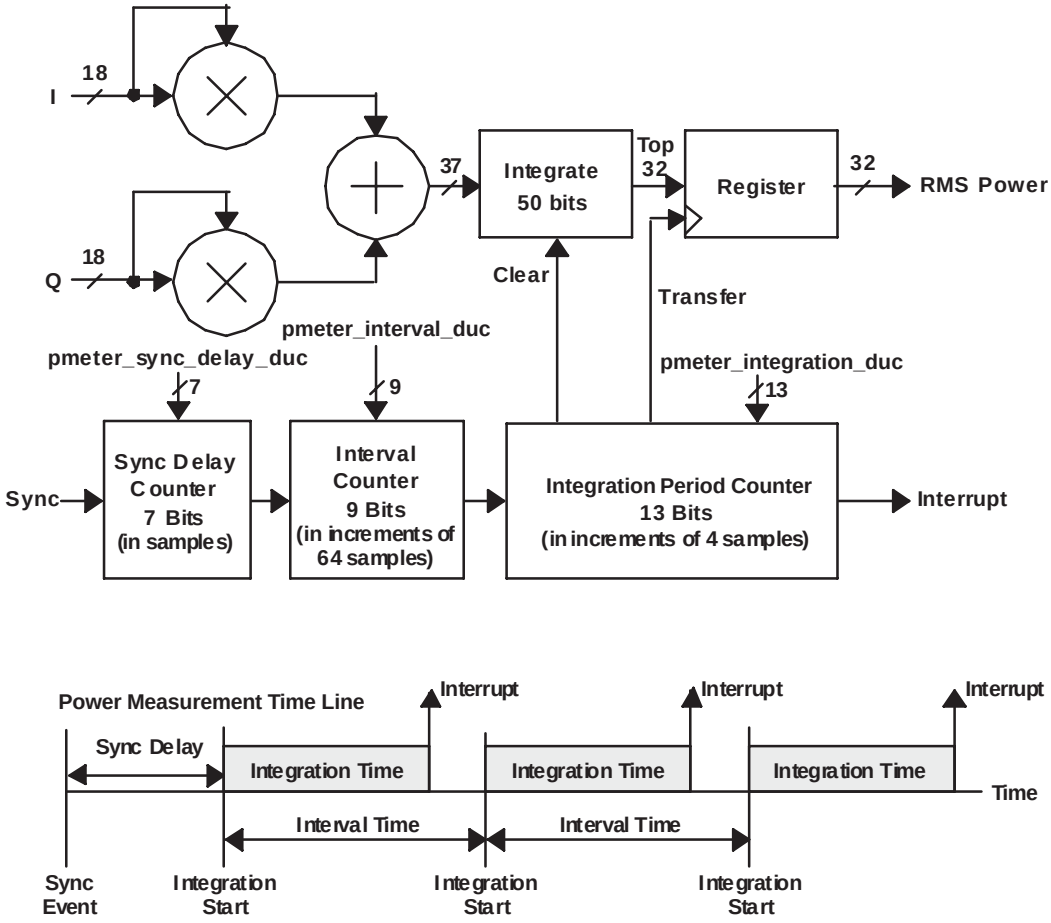


Figure 26. Transmit Channel RMS Power Meter

Each transmit channel includes an RMS power meter used to measure the RMS power within the channel. Functionally, the power meter block is identical to the RMS power meter blocks used in the receive chain.

The power meter samples the I and Q data stream before the PFIR filter. Both 18-bit I and Q data are squared, summed, and then integrated over a time determined by pmeter_integration_duc (13 bits). Integration time = 4 x pmeter_integration_duc + 1 (in units of a sample period or generally a chip period).

There is a programmable 9-bit interval counter which sets the interval over which power measurements are repeated. The timer counts in increments of 64 samples. The interval time = 64(pmeter_interval_duc+1) The interval time must be greater than (not equal to) the integration time.

The power measurement process starts with a sync event (ssel_pmeter). The integration starts at sync event + 3 chips + sync_delay. The 7-bit delay register permits delays from 3 to 130 samples after sync. The integration continues until the integration count is met. At that point the top 32 bits of the 50-bit accumulator is transferred to the read register and an interrupt is generated indicating the power value is ready to read. The interval counter continues until the programmed interval count is reached. When reached, the integration counter and the interval counter start over again. Each time the integration count is reached the upper 32 bits are again transferred to the read register overwriting the previous value sending an interrupt signifying the data is ready to be read. Failure to read the data timely results in overwriting the previous interval measurement.

Sync ssel_pmeter starts the process. Whenever a sync is received, all the counters are reset to zero no matter what the status.

For UMTS, I and Q are calculated and the integrated power is read. When in CDMA mode the power is calculated for both the A and B signals, producing two 32-bit results.

For CDMA mode, the integration time is slightly longer. The power read in CDMA mode with a dc input is:

- A power: $[I^2 \times (X \times 4 + 1) + Q^2 \times (X \times 4 + 0)] \times 2^{-18}$. Note, one Q sample is missing from the integration.
- B power: $[I^2 \times (X \times 4 + 1) + Q^2 \times (X \times 4 + 1)] \times 2^{-18}$

Where X is the integration count.

Table 5. Programming

VARIABLE	DESCRIPTION
pmeter_result_a_lsb(15:0)	Lower 16 bits of the A channel power measurement.
pmeter_result_a_msb (31:16)	Upper 16 bits of the A channel power measurement.
pmeter_result_b_lsb (15:0)	Lower 16 bits of the B channel power measurement result. Only available in CDMA mode.
pmeter_integration_duc(12:0)	Integration time = 4 x pmeter_integration_duc+1.
pmeter_sync_delay_duc(6:0)	Sync delay count in samples.
pmeter_interval_duc(9:0)	Interval time = 64(pmeter_interval_duc+1). Interval time must be greater than (not equal) integration time.
ssel_pmeter(2:0)	Sync source options.
pmeter_sync_disable	Turns off sync to the channel's power meter

3.1.5 Transmit Filter Chain

GC5316 transmit filtering is performed in three stages:

- Interpolate by two pulse-shape filtering using the programmable FIR filter (PFIR)
- Interpolate by two compensation filtering using the programmable compensating FIR filter (CFIR)
- High-rate interpolation (4 to 32) using the six stage cascade-integrate comb filter (CIC)

Figure 27. DUC Filter Chain

The purpose of the transmit filter chain is to interpolate the input signal data up to the mixer clock rate, nominally 122.88 MHz. The following table provides two examples of how the interpolation can be allocated among the three different filters for both CDMA and UMTS.

Table 6. Example UMTS and CDMA2000 DUC Transmit Modes

	INPUT RATE RATE	PFIR INTERPOLATION	CFIR INTERPOLATION	CIC INTERPOLATION	OVERALL INTERPOLATION
CDMA	1.2288 MSPS	2	2	25	100
UMTS	3.84 MSPS	2	2	8	32

3.1.5.1 Transmit Programmable FIR Filter

The transmit programmable FIR filter (PFIR) pulse shapes the baseband signal data and interpolates by a fixed factor of two.

The PFIR length is programmable. This permits turning off taps and saving power if short filters are appropriate. The maximum PFIR filter length is a function of GC5316 clock rate and input sample rate and is limited by the number of coefficient memory registers. The number of taps available ranges in CDMA mode ranges from 31 to 63, in UMTS mode it ranges from 15 to 63. Both in increments of four taps.

Subject to the above range, the maximum number of taps available is:

- UMTS Mode: 2 x (txclk ÷ input sample rate)
- CDMA Mode: txclk ÷ input sample rate

Assuming a txclk of 122.88 MHz, both UMTS (3.84 MSPS) and CDMA (1.2288 MSPS) modes provide 63 taps. The same PFIR coefficients are used for both the A and B signals in CDMA mode.

The PFIR filter consists of 32 forward and reverse data RAM cells each 36 bits in width. The coefficient memory provides storage for up to 63 unique 18-bit taps. A 19 x 18 multiplier and full-precision accumulator form the filter convolution. An optional (pfir_gain) up-shift of one follows. Finally, the output is hard-limited. The PFIR gain is:

Gain = sum(coefficients) x $2^{pfir\_gain - 18}$

Table 7. Programming

VARIABLE	DESCRIPTION
crastsrttap_pfir(4:0)	Number of DUC PFIR filter taps is (2 x crastarttap) +1
cdma_mode	When set, puts the CFIR and PFIR blocks in CDMA2000 mode.
symmetric_pfir	Set to 1 if filter is symmetric. This saves a modest amount of power.
	The PFIR filter's 18-bit coefficients are loaded by the software cmd5316. The user must provide a coefficient file with one integer coefficient per line. Note that the PFIR filter coefficients are shared by the A and B signals in CDMA mode.

3.1.5.2 Transmit Compensating FIR filter

The transmit CFIR filter interpolates by a fixed factor of two and is usually programmed to compensate for the CIC filter's roll-off.

The CFIR filter length is programmable. This permits turning off taps and saving power if short filters are appropriate. The maximum CFIR filter length is a function of GC5316 clock rate and input sample rate and is limited by the number of coefficient memory registers. The number of CFIR taps in CDMA mode is 31 to 47, while in UMTS mode it is 15 to 31. The number of taps may be increased in increments of four taps.

Subject to the above minimum, maximum, and increment values, the maximum number of taps available is:

- UMTS Mode: txclk ÷ input sample rate
- CDMA Mode: 0.5 x (txclk ÷ input sample rate)

Assuming a txclk of 122.88 MHz, UMTS (at 3.84 MSPS) mode would provide 31 taps and CDMA (1.2288 MSPS) mode provides 47 taps.

The CFIR coefficients are shared by the A and B signals in CDMA mode. CFIR gain is:

Gain = sum(coefficients) x $2^{cfir\_gain-19}$

Table 8. Programming

VARIABLE	DESCRIPTION
crastsrttap_cfir(4:0)	These bits define the number of taps that CFIR uses for the filtering. DUC CFIR: (2 x crastarttap_cfir) +1, Note: crastarttap_cfir must be odd.
symmetric_cfir	Set to 1 if filter is symmetric. Saves a bit of power.
cfir_gain	CFIR gain adjustment.
	The PFIR filter's 18-bit coefficients are loaded by the software cmd5316. The user must provide a coefficient file with one integer coefficient per line. Note that the PFIR filter coefficients are shared by the A and B signals in CDMA mode.

3.1.5.3 Transmit CIC Filter

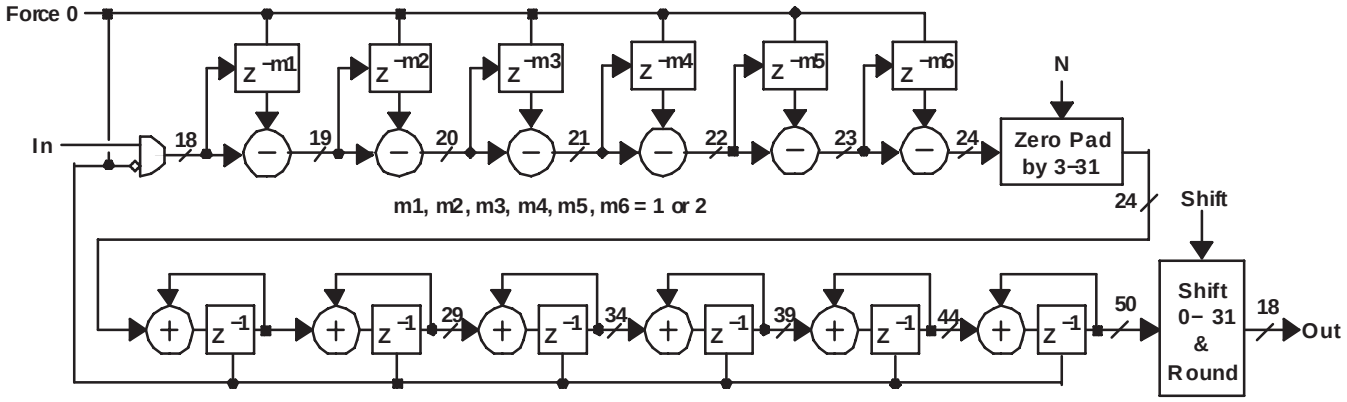


Figure 28. Transmit CIC Filter

The transmit six stage CIC filter interpolates over a programmable range from 4 to 32. The filter is made up of six banks of 24-bit subtractor sections followed by six banks of integrator sections. Each of the six subtractor sections can be independently programmed with a differential delay of one or two. A shift block follows the last integration stage and can shift the 50-bit accumulated data down by 31−TCIC_SHIFT bits yielding 18-bit output data.

The CIC filter exhibits a droop across its frequency response. Usually the preceding CFIR filter precompensates for the CIC droop with a gradually rising frequency response. However, it is also possible to provide the precompensation in the PFIR filter.

CIC interpolation filters can become unstable if an external event (such as a cosmic particle) disturbs a storage node in the CIC integrator section. This can add a bias which subsequently integrates out of control. The GC5316 transmit CIC employs a patented method to detect and then automatically flush and reset the filter. Register bits are available to disable and to test this auto-flush feature. A maskable interrupt becomes active if a CIC error occurred.

The gain of the CIC filter is: $Ncic^5 \times 2^{(\text{number of stages where M=2})} \times 2^{(CIC\_SCALE - 31)}$ where CIC_SCALE is 0 to 31. Ncic is the interpolation ratio and is programmed as cic_interp_decim + 1.

Since the CIC output is full rate for both UMTS and CDMA, a complete hardware path is required for each of the signals A and B from this point on in the transmit signal path. For CDMA, there are four independent CIC filters (I/Q for signal A and I/Q for signal B). For UMTS, the two signal B CIC filters are disabled.

Table 9. Programming

VARIABLE	DESCRIPTION
cic_interp_decim(4:0)	The CIC interpolation is Ncic = cic_interp_decim + 1. This ratio applies to both A and B channels of the DUC block in CDMA mode. Legal values for cic_interp_decim are 3 to 31.
cic_scale_a(4:0)	The shift value for the A channel. A value of 0 is no shift, each increment in value increases the amplitude of the shifter output by a factor of 2.
cic_scale_b(4:0)	The shift value for the B channel. A value of 0 is no shift, each increment in value increases the amplitude of the shifter output by a factor of 2.
cic_m2_ena_a(5:0)	Sets the differential delay value M for each of the CIC subtractor stages for the A channel. Cic_m2_en_a(0) controls m1, cic_m2_en_a(5) controls the m5. A set bit programs the differential delay M to 2, if cleared M is programmed to 1.
cic_m2_ena_b(5:0)	Sets the differential delay value M for each of the CIC subtractor stages for the B channel.
ssel_cic(2:0)	Sync source
cic_auto_flush_dis(3:0)	When set disables the CIC auto−flush. Bits {0, 1, 2, 3} correspond to CICs for {CDMA−A I data, CDMA−A Q data, CDMA−B I data, CDMA−B Q data} sections.
cic_auto_flush_test(3:0)	On rising forces a CIC overflow error. Program to 0 then to 1 for edge to occur. Bits {0, 1, 2, 3} correspond to CICs for {CDMA−A I data, CDMA−A Q data, CDMA−B I data, CDMA−B Q data} sections.
cic_auto_flush_clear(3:0)	On rising clears a CIC overflow condition. Program to 0 then to 1 for edge to occur. Bits {0, 1, 2, 3} correspond to CICs for {CDMA−A I data, CDMA−A Q data, CDMA−B I data, CDMA−B Q data} sections.

3.1.6 Transmit Adjustable Channel Delay

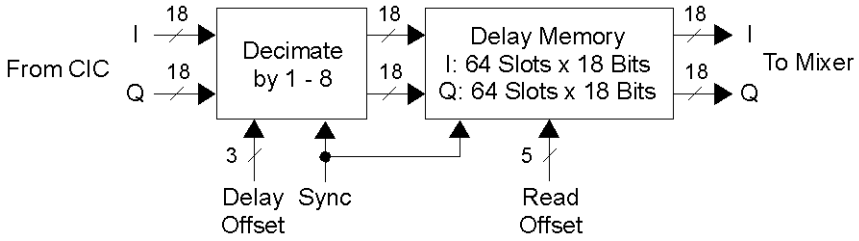


Figure 29. Transmit Delay Adjustment

The transmit channel delay adjust function permits the user to add a programmable time delay in each of the upconverter paths. This is used to calibrate multiple transmit channels in the overall base transceiver system. The adjustable delay compensates for analog elements external to the digital upconversion such as cables, splitters, analog upconverters, filters, etc., and to compensate for differential delay between channels within the GC5316. There is an additional delay of two output sample times for each pair of DUC blocks to allow for pipelining of the sumchain (specifically, DUC0 and 1 have the same delay, DUCs 2 and 3 are the same but are two output sample times larger than DUC0 and 1, etc.).

There are two elements that need to be considered with respect to programming the delay: the decimation and delay memory blocks.

The decimation function reduces the sample rate from txclk to the desired output rate. The decimation amount is set by parameter (tadj_interp_decim+1). Phasing of the decimation operation permits finer delay resolution. The 3-bit delay offset parameter permits finer delay resolution in steps of the reciprocal of the GC5316's tx clock rate. At 122.88 MHz, this would equate to a time delay resolution of 8.1 ns (1/32 chip for UMTS, 1/100 of a chip for CDMA). The offset may be set from 0 to tadj_interp_decim.

The coarse delay adjustment is done using a delay memory of 64 memory locations by 36 bits (18 for I and 18 for Q). Read and write pointers in the memory are separated by tadj_offset_coarse. Data written into a location is read out tadj_offset_coarse output sample times later. 24 locations are needed to equalize the time delay within the GC5316 for various channels. The remaining 40 locations provide a total delay of up to about 1.3 μs when the DUC output data rate is 30.72 MSPS**.**

A sync signal permits the decimation operation to be synchronized over multiple channels.

Table 10. Programming

VARIABLE	DESCRIPTION
tadj_offset_coarse_a(5:0)	Read offset into the 64 element memory for the A channel DUC. When tadj_offset_coarse_a = 62, then the delay is –2. When tadj_offset_coarse_a = 63, then the delay is –1. For all other values, the resulting delay is equal to the value.
tadj_offset_coarse_b(5:0)	Read offset into the 64 element memory for the B channel DUC when in CDMA mode. Note: When tadj_offset_coarse_b = 62, then the delay is –2. When tadj_offset_coarse_b = 63, then the delay is –1. For all other values, the resulting delay is equal to the value.
tadj_offset_fine_a(2:0)	Controls the zero offset (fine adjust) for the A side of the DUC. See note below for mapping.
tadj_offset_fine_b(2:0)	Controls the zero offset (fine adjust) for the B side of the DUC when in CDMA mode. See the note below for mapping.
tadj_interp_decim(2:0)	The decimation value (1, 2, 4, or 8) for the DUC. Same for A and B channels when in CDMA mode.
ssel_tadj_fine(2:0)	Selects the sync source for the fine time adjust
ssel_tadj_coarse(2:0)	Selects the sync source for the coarse time delay adjust

NOTE: The fine adjust is mapped differently.

For a decimation value of 1 the only legal setting is 0 and there is no fine adjustment since there is no decimation moment.

tadj_offset_fine = 0 ⇒ fine delay by 0

For a decimation value of 2:

tadj_offset_fine = 0 ⇒ fine delay by 0

tadj_offset_fine = 1 ⇒ fine delay by 1

For a decimation value of 4:

tadj_offset_fine = 0 ⇒ fine delay by 0

tadj_offset_fine = 1 ⇒ fine delay by −1

tadj_offset_fine = 2 ⇒ fine delay by −2

tadj_offset_fine = 3 ⇒ fine delay by 1

For a decimation value of 8:

tadj_offset_fine = 0 ⇒ fine delay by 0

tadj_offset_fine = 1 ⇒ fine delay by −1

tadj_offset_fine = 2 ⇒ fine delay by −2

tadj_offset_fine = 3 ⇒ fine delay by −3

tadj_offset_fine = 4 ⇒ fine delay by −4

tadj_offset_fine = 5 ⇒ fine delay by −5

tadj_offset_fine = 6 ⇒ fine delay by −6

tadj_offset_fine = 7 ⇒ fine delay by 1

3.1.7 Transmit Mixer

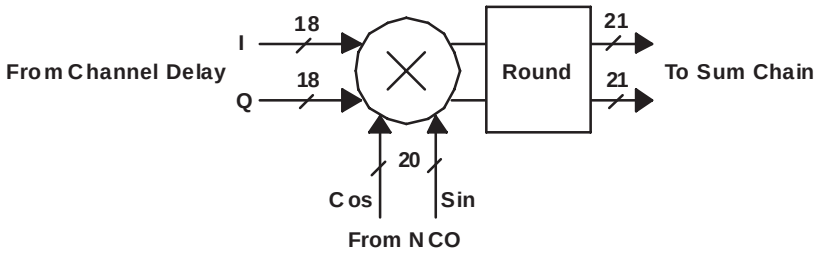


Figure 30. Transmit Mixer

The transmit mixer, like the receive mixer, is a complex multiplier which takes the baseband I and Q data that has been previously pulse-shaped and interpolated and translates to a carrier frequency programmed into the NCO. The mixer data size is 18 bits for the signal path and 20 bits for the NCO path.

The gain through the mixer is –12 dB. It can be increased by 6 dB through a control bit. It is recommended that this extra gain always be used. The output is then rounded to 21 bits.

Mixer gain = $2^{mixer\_gain - 2}$

The mixer output of each channel is combined daisy-chain fashion in four sum chain adder blocks that are described in a subsequent section.

For CDMA, the maximum output rate is txclk/2. The maximum output rate for UMTS is txclk.

Table 11. Programming

VARIABLE	DESCRIPTION
mixer_gain	When asserted adds 6 dB of gain in the mixer. Should always be set.

3.1.8 Transmit NCO

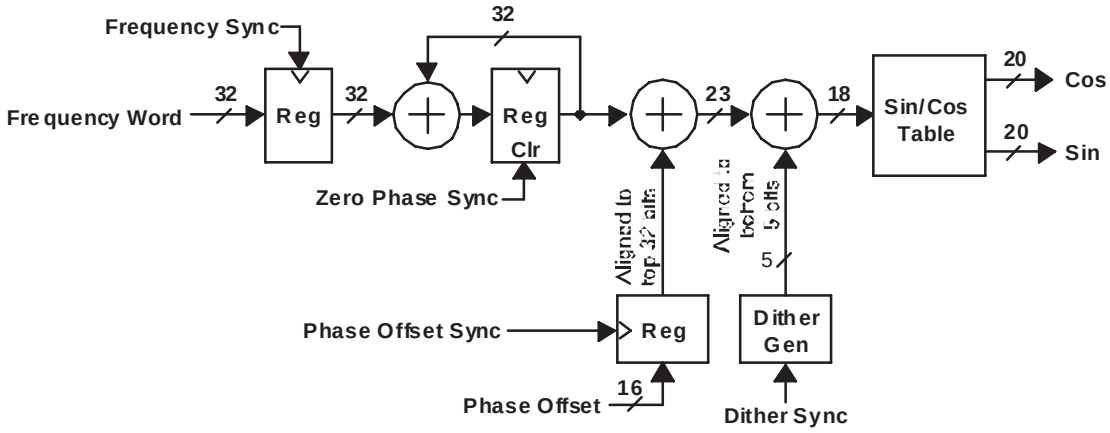


Figure 31. Transmit NCO

The NCO is a digital complex oscillator that is used to translate (or upconvert) interpolated and filtered baseband signals to a programmable carrier frequency.

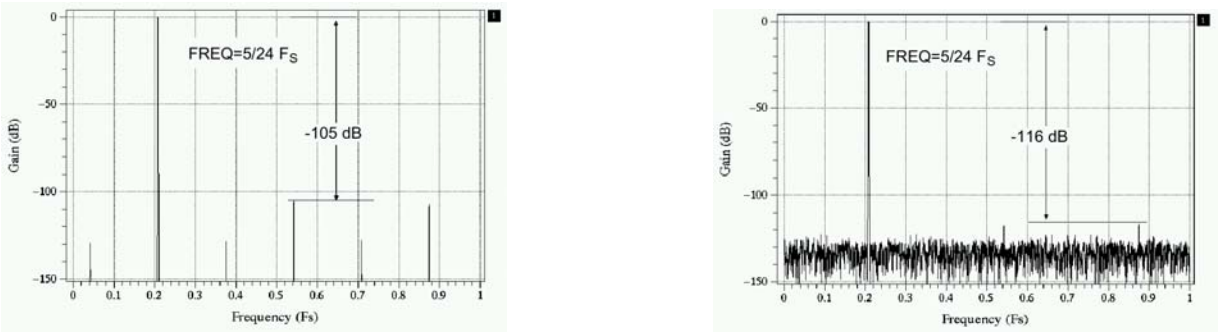
The block produces programmable complex digital sinusoids by accumulating a frequency word which is programmed by the user. The output of the accumulator is a phase argument that indexes into a Sin/Cos ROM table which produces the complex sinusoid. A phase offset can be added prior to indexing if desired for channel calibration purposes. This changes the Sin/Cos phase with respect to other channels' NCOs.

A 5-bit dither generator is provided and generates a small level of digital pseudo-noise that is added to the phase argument below the bottom bit and is useful for reducing NCO spurious outputs.

Table 12. Programming

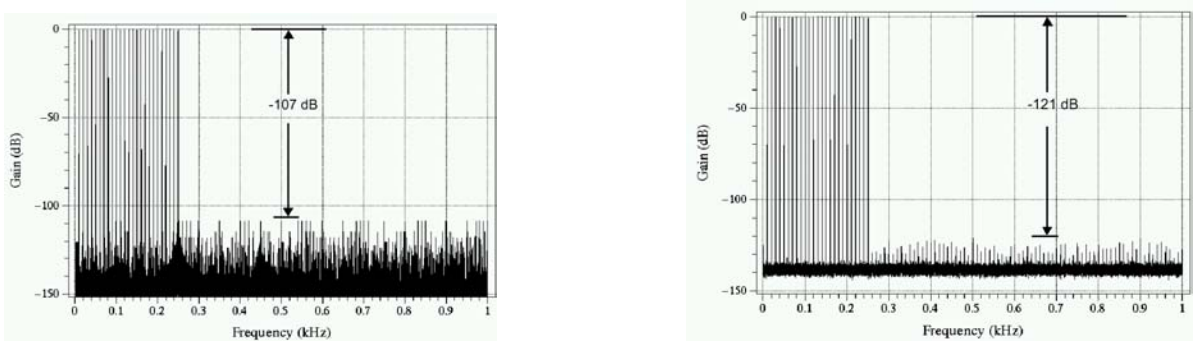
VARIABLE	DESCRIPTION
dither_ena	When set turns dither on. Clearing turns dither off.

The NCO spurious levels are better than –115 dBC. Added phase dither randomizes the ROM lookup slightly, hence the ROM lookup error – spreading the spurious energy around rather than concentrating it in a few frequencies. The phase dither is added below the lsb of the ROM lookup. If the tuning frequency has no high bits more than 17 bits below the msb, the phase dither has no effect. If the tuning frequency is a multiple of Fs/96 then an initial phase offset of four often reduces NCO spurs. Figure 32 and Figure 33 show the spur level performance of the NCO without dither, with dither, and with a phase offset value.



a) Worst Case Spectrum Without Dither **b) Spectrum With Dither (Tuned to Same Frequency)**

Figure 32. Example NCO Spurs With and Without Dithering



a) Plot Without Dither or Phase Initialization **b) Plot With Dither and Phase Initialization**

Figure 33. NCO Peak Spur Plot

The tuning frequency is specified as a 32-bit frequency word and is programmed as three sequential 16-bit words over the control port. The NCO operates at the same speed as the txclk / (tadj_interp_decim + 1). The frequency resolution is simply the $F_{clk}/ 2^{32}$. The NCO frequency resolution is simply the $F_{clk}/ 2^{32}$. As an example, at an input clock rate of 61.44 MHz, the frequency step size would be approximately 14 MHz. The frequency word is determined by the formula:

Frequency word (in decimal) = 2^{32} x Tuning Frequency / F_{clk}

NOTE: Frequency tuning words can be positive or negative valued. Specifying a positive frequency value translates baseband frequencies upward. Specifying a negative tuning frequency translates baseband frequencies downwards.

Table 13. Programming

VARIABLE	DESCRIPTION
phase_add_a(31:0)	32-bit tuning frequency word for the A signal when in CDMA mode. Also for UMTS mode.
phase_add_b(31:0)	32-bit tuning frequency word for the B signal when in CDMA mode. Not used in UMTS mode.

The phase of the NCO's Sin/Cos output can be adjusted relative to the phase of other channel NCOs by specifying a phase offset. The phase offset is programmed as a 16-bit word, yielding a step size of about 5.5 milliDegrees. The phase offset word is determine by the following formula:

Phase Offset Word= 2^{16} x Offset_in_Degrees / 360 or,

Phase Offset Word= 2^{16} x Offset_in_Radians / 2π

Table 14. Programming

VARIABLE	DESCRIPTION
phase_offset_a(15:0)	16-bit phase offset word for the A signal when in CDMA mode. Also for UMTS mode.
phase_offset_b(15:0)	16-bit phase offset word for the B signal when in CDMA mode. Not used in UMTS mode.

Various synchronization signals are available, which are used to synchronize the NCOs of all channels with respect to each other. Frequency sync and phase offset sync determine when frequency and phase offset changes occur. For example, generating a frequency sync after programming the two frequency words causes the NCO (or multiple NCOs) to change frequency at that time, rather than after each of the three frequency words is programmed over the control bus. The zero phase sync signal is used to force the sine and cosine oscillators to their zero phase state. Dither sync can be used to synchronize the dither generators of multiple NCOs. The NCOs used in the transmit section are identical to what is described for the receive section.

Table 15. Programming

VARIABLE	DESCRIPTION
ssel_nco(2:0)	Sync source for NCO accumulator reset
ssel_dither(2:0)	Sync source for NCO dither reset
ssel_freq(2:0)	Sync source for NCO frequency register loading
ssel_phase(2:0)	Sync source for NCO phase register loading

3.1.9 Transmit Sum Chain

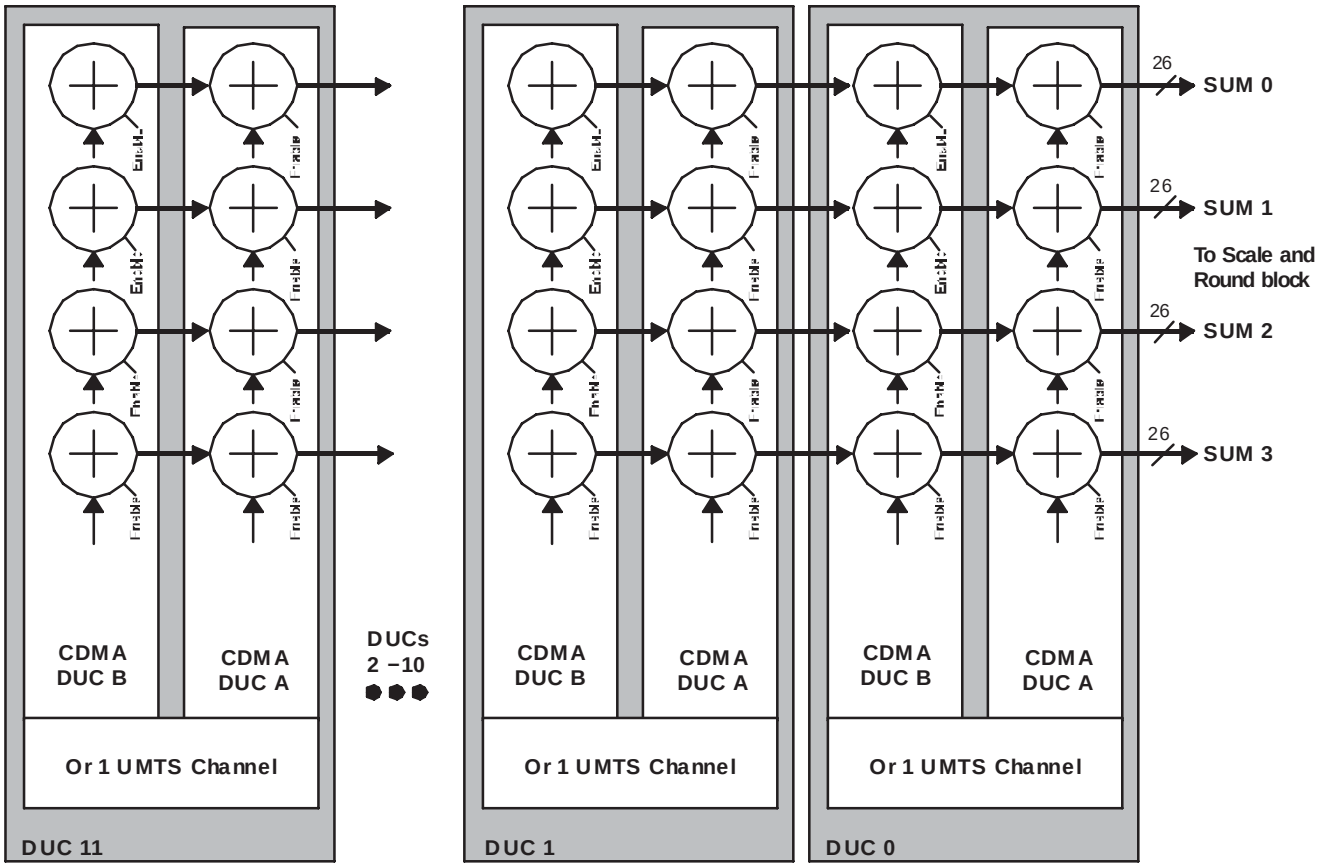


Figure 34. Transmit Sum Chain

The transmit sum chain is a daisy-chain of all DUC outputs into four independent composite output streams. Each DUC output drives four complex adders, each summing the DUC's contribution into the sum chain.

The DUC output data driving the adders is 21 bits. The sum chain partial sum outputs are 26 bits to allow for word growth. Each DUC output can contribute to any of the four sum chains, or not, via programmable enable lines. The output of the last daisy-chained sum is then the composite of all of the 24 CDMA or 12 UMTS channels. One should always ensure that within a sum chain, there are no DUCs powered down with lower numbers than those that are active with higher numbers, thus breaking the chain. In other words, if DUCs 3−5 are used and active, DUCs 0−2 must not be powered down.

As shown in the diagram above, each DUC contains a portion of the sumchain. Within the programming for each DUC one can enable adding results from that DUC signal path(s) to each of the four sum-chains using the parameters below.

Table 16. Programming

VARIABLE	DESCRIPTION
sumchn_sel_a(3:0)	Enable bits signal A contribution to the four sum chain outputs. Signal A Out 0000 Signal A added to no busses 0001 Signal A I/Q to bus0 0010 Signal A I/Q to bus1 0100 Signal A I/Q to bus2 1000 Signal A I/Q to bus3 Note: Signal A output can contribute to any combination of the four sumchain outputs. The above 4-bit code can range from 0 to 15.
sumchn_sel_b(3:0)	Enable bits for signal B contribution to the four sum chains (only when in CDMA mode). Signal B Out 0000 Signal B added to no busses 0001 Signal B I/Q to bus0 0010 Signal B I/Q to bus1 0100 Signal B I/Q to bus2 1000 Signal B I/Q to bus3 Note: Signal B output can contribute to any combination of the four sumchain outputs. The above 4-bit code can range from 0 to 15.

3.2 Transmit Sum Chain Shifting and Rounding

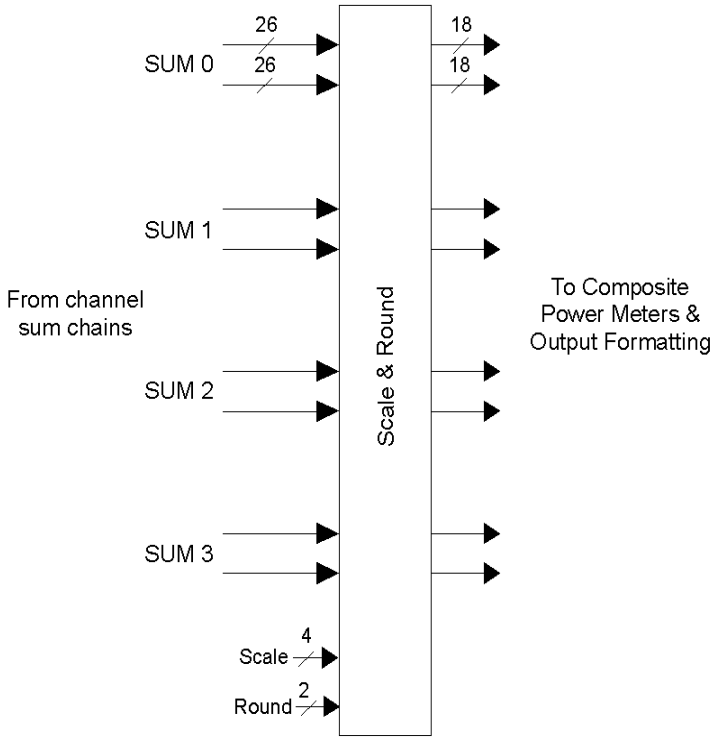


Figure 35. Final Sum Chain Scale and Round Block

Summed data is scaled from 26 bits down to 18 bits. The desired 18 bits can be taken anywhere over the 26 bit sum chain output window via a programmable register. These 18 bits can range from sumchain(25:8) on the top end, to sumchain (17:0) on the bottom end of the 26 bit output.

The scaled data is then hard-limited and rounded to 18, 16, 14, or 12 bits. Rounded data is MSB justified with the bottom bits zeroed. For example, 12-bit rounding would force the output data to the top 12 bits of the 18-bit word and the bottom 6 bits would be zeroed.

Gain = $2^{interf\_scale-5}$

Table 17. Programming

VARIABLE	DESCRIPTION
interf_scale_0(3:0) interf_scale_1(3:0) interf_scale_2(3:0) interf_scale_3(3:0)	Selects the sum chain scale value for each of the four sum chains. The 18-bit output can be slide anywhere across the 26-bit window. 0000 = sumchain(25:8) 0001 = sumchain(24:7) … 0111 = sumchain(18:1) 1000 = sumchain(17:0)
interf_round(1:0)	Specifies the rounding of all four sum chains. 00 = 18 bits 01 = 16 bits 10 = 14 bits 11 = 12 bits

3.2.1 Transmit Power Meters

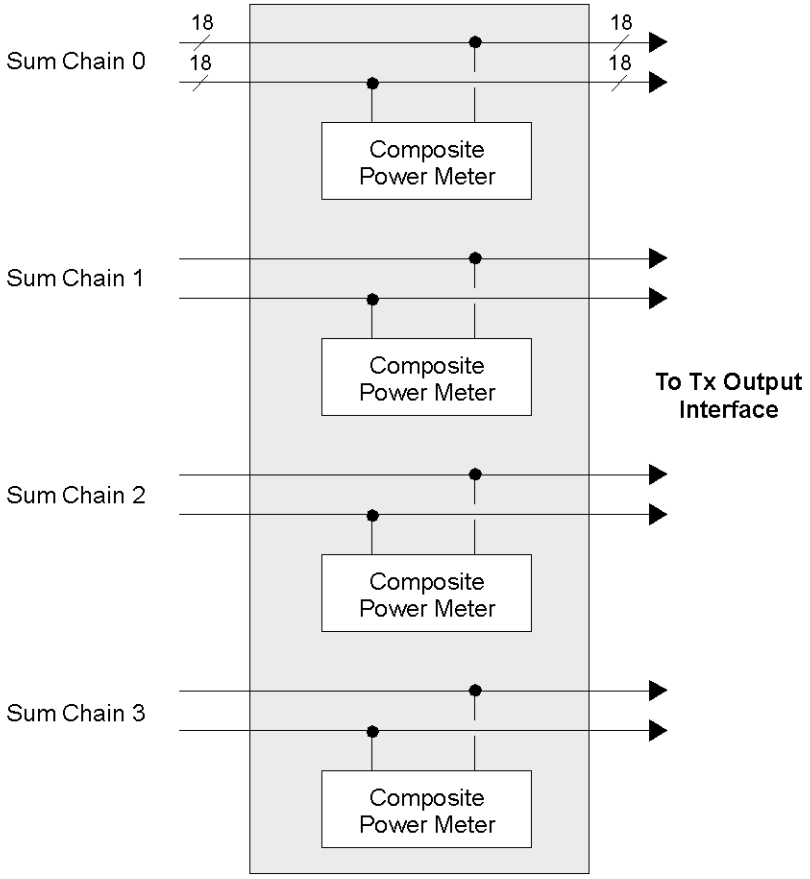


Figure 36. Transmit Power Meter Block

The composite power in each of the four transmit sum chains can be measured using power meters similar to those used in the individual DDC and DUC blocks.

There are four composite RMS power meters, one for each of the four sum chains. Each of the above power meters are independently programmable with respect to the measurement period, interval, and delay from sync. The following two sections describe the sum chain power meters in more detail.

3.2.1.1 Transmit Composite RMS Power Meter

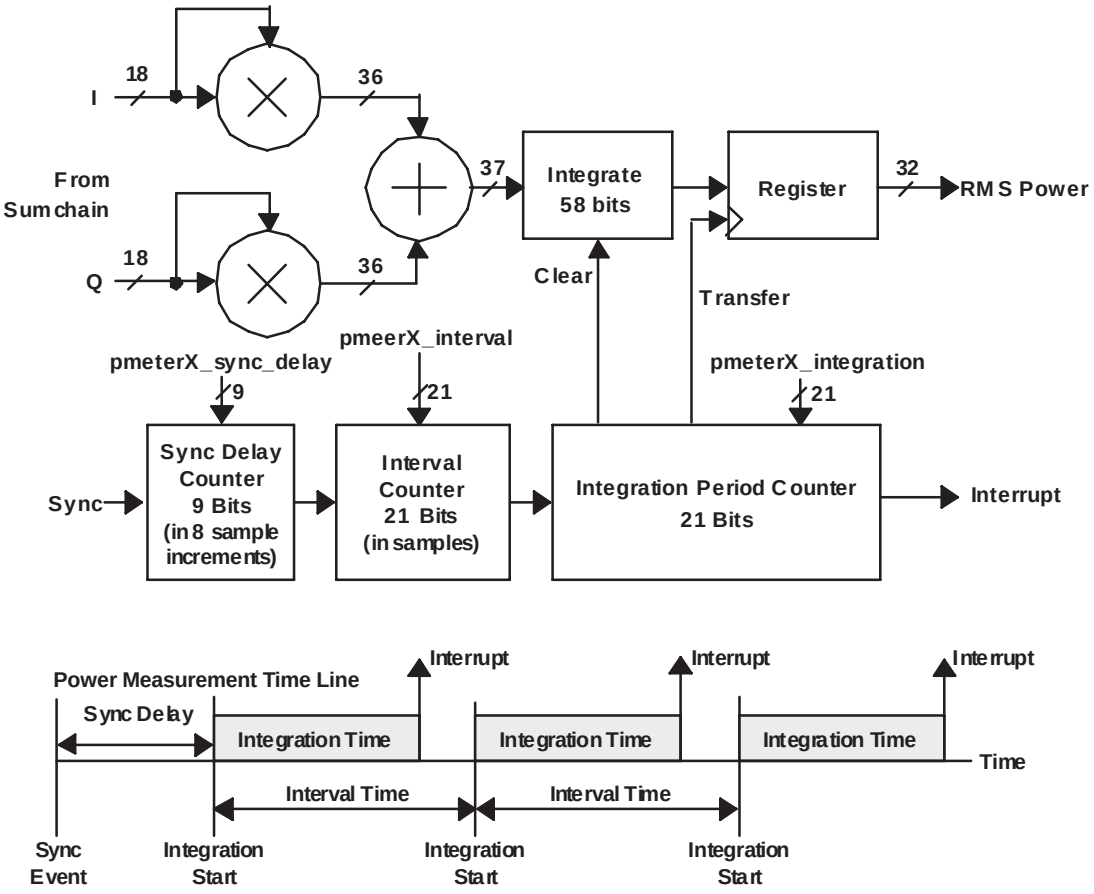


Figure 37. Transmit RMS Power Meter

The GC5316 provides four independent transmit output ports, each of which is the sum of a number of individual transmit carriers (a sum chain). Four composite RMS power meters measures the RMS power of the combined carriers in each of the four sum chains. These power meters are similar to those used to measure the RMS power of each individual channel, but have different counter lengths.

The input to the power meter is the scaled and rounded output of a sum chain. Power is calculated by squaring each 18-bit I and Q sample, summing, and then integrating the summed-squared results into a 58-bit accumulator. The integration time is pmeter_integration (21 bits) output sample periods.

There is a programmable 21-bit interval counter which sets the interval over which power measurements are repeated. The interval time = pmeter_interval+1. The interval time must be greater than (not equal to) the integration time. A measurement integration period is started at the beginning of each interval time.

The process begins with a sync event starting the 9-bit delay counter. After the delay count + 2 samples, the integration interval is started. The power is calculated for each I and Q sample and added to the 58-bit accumulator. The integration continues until the integration count is met at which point the upper 32 bits of the 58-bit integrator are transferred to the read register and an interrupt is generated. A new measurement period starts at the end of the interval period.

NOTE: Each of the four composite RMS power meter blocks has its own delay sync, interval, and integration period counters, as well as separate sync source registers.

Table 18. Programming

VARIABLE	DESCRIPTION
comp_pmeterX_result_lsb(15:0)	Lower 16 bits of the composite power measurement for sum chain X. X= 0, 1, 2, or 3
comp_pmeterX_result_msb(31:16)	Upper 16 bits of the composite power measurement for sum chain X. X= 0, 1, 2, or 3
comp_pmeterX_integration_lsb(15:0)	Lower 16 bits of the 21-bit integration period. X = 0, 1, 2, or 3
comp_pmeterX_integration_msb(20:16)	Upper 5 bits of the 21-bit integration period. X = 0, 1, 2, or 3
comp_pmeterX_sync_delay(8:0)	Power meter delay sync period. X = 0, 1, 2, or 3
comp_pmeterX_interval_lsb(15:0)	Lower 16 bits of the 21-bit measurement interval. X = 0, 1, 2, or 3
comp_pmeterX_interval_msb(20:16)	Upper 5 bits of the 21-bit measurement interval. The Interval time must be greater than the integration time for each of the four composite power meters. X = 0, 1, 2, or 3
ssel_comp_pmeter_X(2:0)	Sync source. X = 0, 1, 2, or 3

3.3 GC5316 Transmit Output Interface

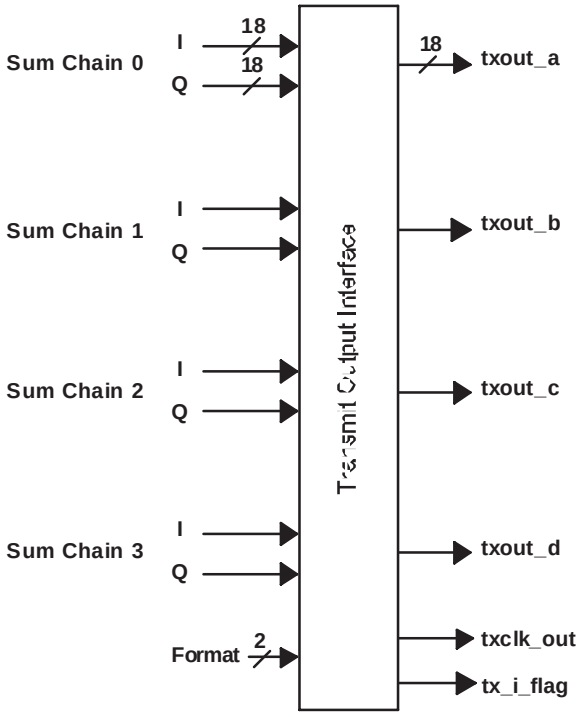


Figure 38. Transmit Output Interface

The GC5316 provides four transmit output signal data ports. Each port can be enabled or disabled. Disabled ports are held low and can also be tri-stated.

Each 18-bit port outputs the sum of the carriers contributing to the composite signal stack. Output data can be real or complex valued. Complex I/Q data can be output either interleaved over a single output port, or, over two ports separately.

Real output data would generally be selected driving a single D/A converter to an IF frequency. Complex output data would be selected when subsequent post-processing such as power amplifier predistortion is employed. Complex outputs can also be used to drive a pair of D/A converters (one for I, the other for Q) for direct I/Q upconversion using a quadrature modulator device.

I and Q complex output data can be interleaved over a single 18-bit port, or, simultaneously over two separate output ports at half the rate. Signal tx_I_flag is active when I data is being output when in complex output interleaved mode. When complex output data is noninterleaved, I data is output on port 0 and Q data is output on port 1 for sum chain 0. For sum chain 1, I data is output on port 2 and Q data is output on port 3.

Real output data is output over a single 18-bit port. For CDMA mode, the maximum real output rate is txclk/2. The maximum real output rate for UMTS mode is txclk.

The maximum complex output rate with I and Q data on separate outputs is txclk/2 for CDMA mode and txclk for UMTS mode. If the complex output data is interleaved on a single bus, the maximum rate is txclk/2 for both UMTS and CDMA and the toggle rate between I and Q samples is txclk.

NOTE: The tx_clk_out signal can not be at full rate (txclk rate) if the mixer is not at full rate (for CDMA mode, the mixer can not be at full rate). If a full-rate clock output signal is desired, the tst_clk signal can be used, with the tst_rate parameter programmed to 0.

Table 19. Programming

VARIABLE	DESCRIPTION
interf_ena(3:0)	When bits are set, enables the corresponding outputs. When cleared, outputs are disabled and held low.
interf_real	When set, outputs are real. When cleared, outputs are complex.
interf_interl	When set, complex data is output interleaved.
tristate(3)	When set, turns on tx_data_out3 outputs.
tristate(2)	When set, turns on tx_data_out2 outputs as well as sync_tst, aflag_tst, and clk_tst.
tristate(1)	When set, turns on tx_data_out1 outputs.
tristate(0)	When set, turns on tx_data_out0 outputs as well as tx_iflag and tx_clk_out.
trt_rate	The value here controls the output clock rate on the clk_tst pin. A value of 0 gives a full rate output clock (txclk rate), a 1 gives half rate output clock, a 3 gives 1/4th rate output clock, and so on. The number of txclk cycles for which the clk_tst signal is high + low = 1 + tst_rate.

4 GC5316 General Control

The GC5316 is configured over a bidirectional 16-bit parallel data microprocessor control port. The control port permits access to the control registers which configure the chip. The control registers are organized using a paged-access scheme using six address lines. Half of the 64 addresses (address 32 through address 63) represent global registers. The other 32 (address 0 through address 31) are paged resisters. This arrangement permits accessing a large number of control registers using relatively few address lines.

Global address 33 is the page register. Writing a 16-bit value to this register sets the page to which future write or read operations performed. These paged-registers contain the actual parameters that configure the chip and are accessed by writing/reading address 0 through address 31.

Global registers (address 32 through address 63) are used to read/write GC5316 parameters that are global in nature and can benefit from single read/write operations. Examples include chip status, reset, sync options, checksum ramp parameters, and the page register.

4.1 Control Data, Address, and Strobes

The control bus consists of 16 bidirectional control data lines C[0:15], 6 address lines A[0:5], a read enable line $\overline{RD}$, a write enable line $\overline{WR}$, and a chip enable line $\overline{CE}$. These lines usually interface to a microprocessor or DSP chip and is intended to look like a block of memory.

Data is written by: 1) Setting up the desired address A[0:5], 2) Setting $\overline{CE}$ low, 3) Setting the desired data on C[0:15], and then 4) Pulsing $\overline{WR}$ low. Data is written when $\overline{WR}$ returns high.

4.2 MPU Timing Diagrams

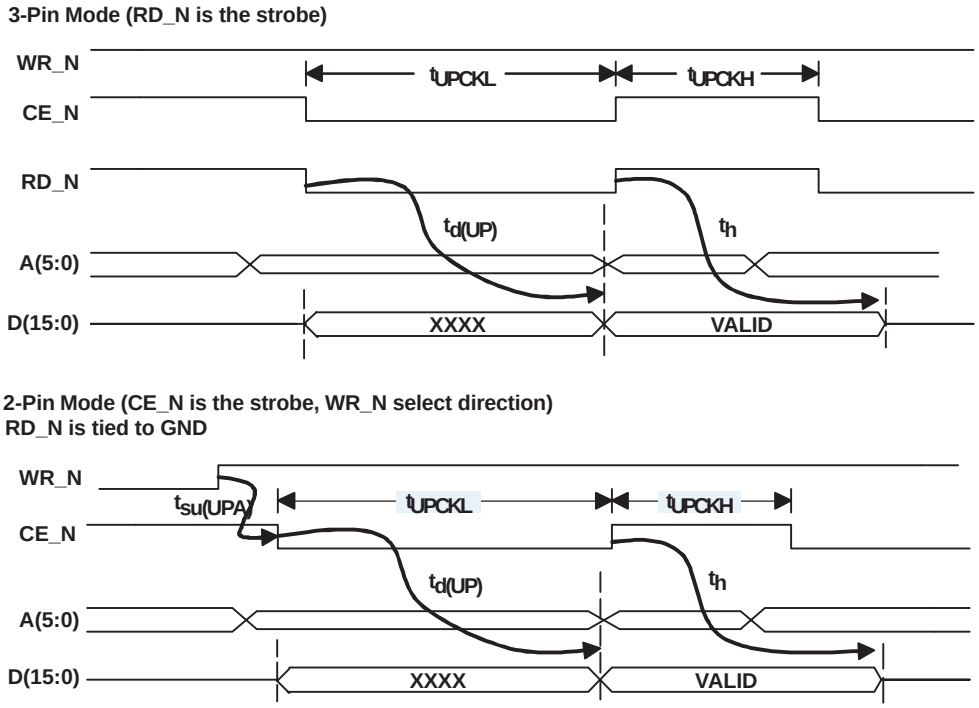


Figure 39. Read Diagrams

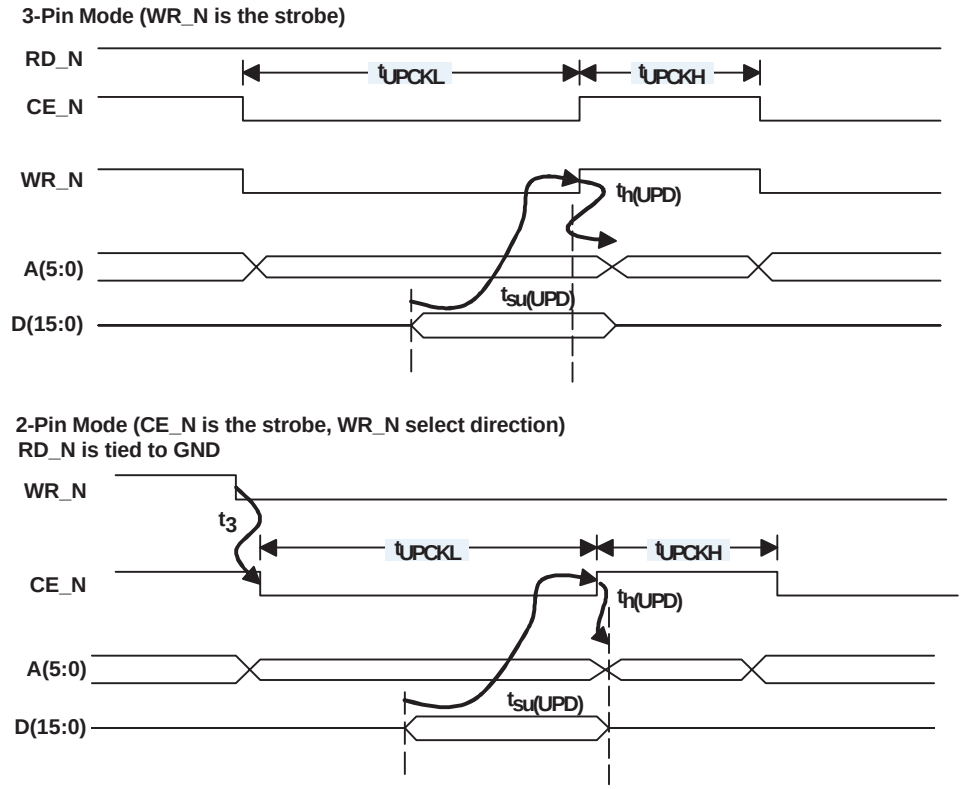


Figure 40. Write Diagrams

4.3 Interrupt Handling

When a GC5316 block sets an interrupt, the interrupt pin goes active if the interrupt source is not masked. The microprocessor should then read the three interrupt flag registers to determine the source of the interrupt. The microprocessor then has to read the interrupt source circuits register(s) to clear the interrupt pin and interrupt flag register bit. The three interrupt registers are listed in the global registers part of the control registers section.

4.4 Sync Signals

Various function blocks within the GC5316 need to be synchronized in order to realize predictable results. The GC5316 provides a flexible system where each function block that requires synchronization can be independently synchronized from either device pins or from a software one-shot. The one-shot option is setup and triggered through control registers. The receive and transmit sections of the chip each have four hardware sync input pins available. These sync pins are qualified on the chip's rising clock edge.

Table 1 shows the different sync modes available for both receive and transmit sections.

Table 1. Different Sync Modes Available for Both Receive and Transmit Sections

MODE	RECEIVE SYNC SOURCE	TRANSMIT SYNC SOURCE
0	RxSyncA	TxSyncA
1	RxSyncB	TxSyncB
2	RxSyncC	TxSyncC
3	RxSyncD	TxSyncD
4	DDC sync counter TC	DUC sync counter TC
5	DDC sync triggered by one-shot	DUC sync triggered by one-shot
6	0 (always off)	0 (always off)
7	1 (always on)	1 (always on)

Table 2 through Table 5 summarize the blocks which have functions that can be synchronized using the above eight sync options:

Table 2. Transmit Common Syncs

SYNC NAME	PURPOSE
ssel_comp_pmeter	Initializes the xmit composite power meter
ssel_duc_counter	Initializes the xmit common sync counter
ssel_duc_serp	Initializes the xmit serial interface
ssel_duc_gain	Updates the gain register
ssel_duc_pilot	Initializes the xmit pilot generator and updates the pilot gain
ssel_duc_tadj	Updates the delay adjust register
ssel_duc_pmeter	Initializes the xmit channel power meter

Table 3. Transmit Channel Syncs

SYNC NAME	PURPOSE
ssel_duc_nco	Resets the NCO accumulator
ssel_duc_freq	Updates the NCO freq registers
ssel_duc_phase	Updates the NCO phase register
ssel_duc_dither	Resets the NCO dither

Table 4. Receive Common Syncs

SYNC NAME	PURPOSE
ssel_ddc_counter	Initializes the receive sync counter
ssel_ddc	Initializes the receive ADC samples interface and clock gen circuits (including CIC decimation)
ssel_ddc_tadj	Updates the delay adjust register
ssel_ddc_pmeter	Initializes the receive power meter
ssel_ddc_agc	Updates the AGC registers
ssel_ddc_pser	Initializes the receive serial interface

Table 5. Receive Channel Syncs

SYNC NAME	PURPOSE
ssel_ddc_nco	Resets the NCO accumulator
ssel_ddc_freq	Updates the NCO freq registers
ssel_ddc_phase	Updates the NCO phase register

4.5 Initialization

Chip initialization procedures are available from Texas Instruments.

4.6 GC5316 Board Diagnostics

The GC5316 contains built-in test features that can be used to confirm that the chip is operating correctly and to help users debug their boards and systems that conatin the GC5316.

The diagnositc and board test procedures can be download from the web at www.ti.com as the GC5316 Diagnostics Designer's Kit.

5 GC5316 Programming

The cmd5016 program, its user's guide, and example configuration files can be downloaded from the web at www.ti.com as the GC5316 Configuration Designer's Kit.

The GC5316 contains over 7,000 control and coefficient registers that must be written to in order to fully configure the chip. Rather than program each of these registers individually, Texas Instruments supplies a configuration program called cmd5316 which accepts top-level configuration information for the chip and then generates the full register map and control writes required to program the chip.

The configuration controls have been defined in the functional description of each section of the chip. The following tables summarize these controls and identify which register and which bits within the registers they occupy.

Each control register table has a column identifying whether the variable must be specified by the user in cmd5316 (U), is typically left at the default value and does not need to be specified (D), is computed by cmd5316 and should not be set (C), or is for expert use only (E).

Tables are also included that show the top level page mapping for the chip controls, the status and measurement result registers, and additional controls that are used with the GC101/GC5316 DIMM evaluation platform.

5.1 cmd5316 Keywords

These keywords are used by the cmd5316 program to set general configuration parameters.

NAME	ARGUMENT	USE	DESCRIPTION
print	config	Global	Tells cmd5316 to generate a configuration output file for general use.
print	gc101	Global	Tells cmd5316 to generate a configuration output file for GC101 use.
print	analysis	Global	Tells cmd5316 to generate a analysis output file
print	table	Global	Tells cmd5316 to generate a table output file
print	power	Global	Tells cmd5316 to generate an approximate power consumption output file.
rxclk	clock frequency in MHz	Global	Used to calculate receive tuning frequencies
txclk	clock frequency in MHz	Global	Used to calculate transmit tuning frequencies
adc_resampler	filename for resampler taps	General Receive	Specifies the filename containing the resampler taps
ddc	channel number	DDC Channels	All controls after this keyword apply to this DDC channel
copy_ddcchan	channel number	DDC Channels	Copy the DDC channel commands from the specified channel to the current channel
duc	channel_number	DUC Channels	All controls after this keyword apply to this DUC channel
copy_ducchan	channel_number	DUC Channels	Copy the DUC channel commands from the specified channel to the current channel
freqa	tuning frequency in MHz	DDCs and DUCs	Sets the NCO tuning frequency for the UMTS channel or for the a-path in the current channel if in CDMA mode.
freqb	tuning frequency in MHz	DDCs and DUCs	Sets the NCO tuning frequency for the b-path in the current channel if in CDMA mode.
pfir_coeff	filename for pfir taps	DDCs and DUCs	Specifies the filename containing the pfir taps
cfir_coeff	filename for cfir taps	DDCs and DUCs	Specifies the filename containing the cfir taps
overall_gaina	overall channel gain	DDCs and DUCs	Optional − Specifies the overall gain for the UMTS channel or the a-path in the current channel if in CDMA mode.
overall_gainb	overall channel gain	DDCs and DUCs	Optional − Specifies the overall gain for the b-path in the current channel if in CDMA mode.

5.1.1 GC5316 DIMM Keywords

These keywords are used to control how the GC5316 DIMM operates in the GC101 evaluation board.

NAME	ARGUMENT	TYPE	DEFAULT	USE	DESCRIPTION
loopback	0 or 1	G	0	GC5316 DIMM	When asserted, txout_a output connected to rxin_a input, else rxin_a input from GC101.
spin0	0 or 1	G	0	GC5316 DIMM	When asserted, txin_[0:5]_[a:b] ports are active, else data from GC101 assumed to go to rxin_a .
spin1	0 or 1	G	0	GC5316 DIMM	When asserted, txin_[6:11]_[a:b] ports are active, else data from GC101 assumed to go to rxin_b .
sigout0	0−3	G	3	GC5316 DIMM	00 – txout_a enabled, 01 – txout_c enabled, 10 – rxout_[0:3]_[a:d] enabled, 11 − none enabled.
sigout1	0−3	G	3	GC5316 DIMM	00 – txout_b enabled, 01 – txout_d enabled, 10 – rxout_[4:7]_[a:d] enabled, 11 − none enabled.
txout_lsb	0 or 1	G	0	GC5316 DIMM	When asserted, the 2 lsb's each of active txout are output to GC101, else the various strobes/syncouts are output.
sel_syncout	0−3	G	3	GC5316 DIMM	(if txout_lsb=0) selects which signals are output. 00−tx_sync_out + tx_i_flag, 01−sync_tst + aflag_tst, 10−tx_sync_out0 + interrupt, 11−rx_sync_out + test6 .
res_op_en	0−3	G	3	GC5316 DIMM	(if txout_lsb=0) when 00, test 9 + test11 is output , test7 + test8 data is output if 01, and rx_sync_out0 when 10.
sel_clkout	0−3	G	0	GC5316 DIMM	Selects the output clock source. 00 − txclk_out, 01−clk_tst, 10−clkout+.
adcclk_set	0 or 1	G	0	GC5316 DIMM	When asserted, gated adcclk. When 0, NOR gate bypassed (i.e. adcclk will be the same as rxclk).

5.1.2 Page Map

This page map describes which pages and what registers within the pages are used. All other pages are unused. This table is provided for reference only, the registers and the bits within the registers are described in the following control register tables.

PAGES	ADDRESS	DESCRIPTION
BASE+000 and BASE+020	00−1F	PFIR Coefficients, 2 LSBs
BASE+040 and BASE+060	00−1F	PFIR Coefficients, 16 MSBs
BASE+080 and BASE+0A0	00−1F	CFIR Coefficients, 2 LSBs
BASE+040 and BASE+0E0	00−1F	CFIR Coefficients, 16 MSBs
BASE+100	00−1F	Channel Control Registers
BASE+120	00−1D	Channel Control Registers
Where BASE is (DUCn x 0200) for DUCn from 0 to 11, and is (DDCn x 0200 + 2000) for DDCn from 0 to 11		
1800	00−09	General Receive Control Registers
1800	0A−1F	adc_resampler coefficients
1820	00−04	adc_resampler coefficients
1820	06	General Receive Control Registers
1C00	00−11 and 1A−1E	General Transmit Control Registers
1C20	00−07 and 0C	General Transmit Control Registers

5.1.3 Status and Read-Only Registers

These registers can be accessed by the user to read status or read measurement results from the chip. These register names are not used in cmd5316.

NAME	PAGE	ADDRESS	LSB POSITION	BIT WIDTH	DESCRIPTION
Version	Global	20	0	5	A 5-bit read only register indicating the current GC5316 revision status
inter_pmeter	Global	25	12	4	Indicates which transmit composite power meter generated the interrupt
inter_tx_pmeter	Global	26	4	12	Indicates which transmit power meter generated the interrupt
inter_rx_pmeter_msb	Global	26	0	4	Indicates which receive power meter generated the interrupt− 4 MSBs
inter_rx_pmeter_lsb	Global	27	8	8	Indicates which receive power meter generated the interrupt− 8 LSBs
inter_tx_cic	Global	28	0	12	Indicates which transmit cic overflow detect generated the interrupt
comp_pmeter0_lsb	1C20	00	0	16	16 LSBs of composit power meter 0
comp_pmeter0_msb	1C20	01	0	16	16 MSBs of composit power meter 0
comp_pmeter1_lsb	1C20	02	0	16	16 LSBs of composit power meter 1
comp_pmeter1_msb	1C20	03	0	16	16 MSBs of composit power meter 1
comp_pmeter2_lsb	1C20	04	0	16	16 LSBs of composit power meter 2
comp_pmeter2_msb	1C20	05	0	16	16 MSBs of composit power meter 2
comp_pmeter3_lsb	1C20	06	0	16	16 LSBs of composit power meter 3
comp_pmeter3_msb	1C20	07	0	16	16 MSBs of composit power meter 3
tx_chk_sum	1C20	0C	0	16	Transmit checksum result
pmeter_a_lsb	BASE+0120	07	0	16	DDCa power meter 16 LSBs
pmeter_a_msb	BASE+0120	08	0	16	DDCa power meter 16 MSBs
pmeter_b_lsb	BASE+0120	09	0	16	DDCb power meter 16 LSBs
pmeter_b_msb	BASE+0120	0A	0	16	DDCb power meter 16 MSBs
ddc_chk_sum	BASE+0120	13	0	16	DDC checksum
	BASE = (DDCn x 0200 + 2000) for DDC channels, where DDCn equals 0 to 11				

5.1.4 Global Control Variables

These registers contain global controls for the GC5316. These registers are not paged and are accessed directly using addresses 32−63 (20−3f hex)

VARIABLE NAME	TYPE	ADDRESS	LSB POSITION	BIT WIDTH	DEFAULT	DESCRIPTION
page	E	21	0	16	0	The Page register selects which page addresses 00−1F (0−31) will access.
slf_tst_ena	D	22	15	1	0	(TESTING PURPOSES) Turns on the checksum LFSR for receive and transmit.
rduz_sens_ena	D	22	14	1	0	When enabled, adds noise to the LSB's to the ADC inputs.
tst_sel_chan	E	22	1	2	0	(TESTING PURPOSES) In each slice, these bits control which tst_out is sent to the transmit block. (which duc/ddc in the slice)
tst_on	E	22	0	1	0	(TESTING PURPOSES) When asserted the testbus is active, txout_c (17:0), and txout_d (17:0) form the 36-bit test word output.
						The following tristates are active low, 0 turns the output on, 1 tristates it.
tristate_10	E	23	10	1	1	Reserved outputs for test, must be set to 1 (tristate)
tristate_9	C	23	9	1	1	This bit turns on the slice5 tx_sync, rx_sync, and rx serial data outputs.
tristate_8	C	23	8	1	1	This bit turns on the slice4 tx_sync, rx_sync, and rx serial data outputs.
tristate_7	C	23	7	1	1	This bit turns on the slice3 tx_sync, rx_sync, and rx serial data outputs.
tristate_6	C	23	6	1	1	This bit turns on the slice2 tx_sync, rx_sync, and rx serial data outputs.
tristate_5	C	23	5	1	1	This bit turns on the slice1 tx_sync, rx_sync, and rx serial data outputs.
tristate_4	C	23	4	1	1	This bit turns on the slice0 tx_sync, rx_sync, rx serial data, tx_sync_out, and rx_sync_out outputs.
tristate_3	C	23	3	1	1	This turns on the txout_d outputs.
tristate_2	C	23	2	1	1	This turns on the txout_c CLK_TST, IFLAG_TST, and SYNC_TST outputs.
tristate_1	C	23	1	1	1	This turns on the txout_b outputs.
tristate_0	C	23	0	1	1	This turns on the txout_a, TX_IFLAG, and TXCLK_OUT outputs.
tx_oneshot	D	24	15	1	0	When set a one shot pulse is sent to the transmit blocks for syncing. This only works if the blocks are programmed to see the oneshot. To use the oneshot again, it must be programmed back to a '0' and then back to a '1'.
rx_oneshot	D	24	7	1	0	When set a one shot pulse is sent to the receive blocks for syncing. This only works if the blocks are programmed to see the oneshot. To use the oneshot again, it must be programmed back to a '0' and then back to a '1'.
imask_comp_pmeter	D	29	12	4	0	Interrupt mask bits for the transmit composite power meter
imask_tx_pmeter	D	2A	4	12	0	Interrupt mask bits for the composite power meter
imask_rx_pmeter_msb	D	2A	0	4	0	Interrupt mask bits for the receive composite power meter− 4 MSBs

imask_rx_pmeter_lsb	D	2B	8	8	0	Interrupt mask bits for the receive composite power meter− 8 LSBs
imask_tx_cic	D	2C	0	12	0	Interrupt mask bits for overflow detection in the transmit cics

5.1.5 General Receive Controls

These registers control the receive interface to the DDC channels

VARIABLE NAME	TYPE	PAGE	ADDRESS	LSB POSITION	BIT WIDTH	DEFAULT	DESCRIPTION
ddc_counter_lsb	D	1800	5	0	16	65535	32-bit interval timer common to all DDC sync inputs. This timer may be programmed to any interval count, and each DDC synchronization input can select this counter as a source. This counter increments on each RX clock rising edge. 16 LSBs
ddc_counter_msb	D	1800	6	0	16	65535	32-bit interval timer common to all DDC sync inputs.16 MSBs
ssel_ddc_counter	U	1800	7	8	3	0	Selects the sync source for the DDC sync counter.
ddc_counter_width	D	1800	7	0	8	0	Sets the width of the counter generated sync pulse in RX clock cycles, from 1 to 256. The width of the the ddc_counter pulse should be set wide enough to be asserted for an entire clock period of the slowest block to use this sync
ssel_adc_fifo	U	1800	8	12	3	6	Selects the sync source for the adc FIFO block. Sync reinitializes the read and write pointers of the FIFO.
ssel_resamp	U	1800	8	8	3	0	Selects the sync source for the ADC_RESAMPLER block.
ssel_rxsync_out	U	1800	8	4	3	0	Selects the sync source for the RXSYNC_OUT pin.
ssel_rxin	U	1800	8	0	3	0	Synchronizes the rx_distribution bus source and destination and clock generation in each of the DDC blocks.
rate_sel	U	1800	9	14	2	0	This selects the FIFO output rate when adc_fifo_bypass = 0. When using the resampler, this value should be programmed to a 0. When set to 0, the FIFO output is clocked by rxclk (gated if resampler is on and decimating by 1.5). When set to 1, the FIFO output rate is 1/2 of rxclk rate. When set to 2, the FIFO output rate is 1/4 of rxclk rate, and when set to 3, the FIFO output is at 1/8 of rxclk rate. E.g.: With rxclk 122.88 MHz, set rate_sel to 0, 1, 2 or 3 respectively for adcclk 122.88, 61.44, 30.72 or 15.36 MHz.
resampler_ena	U	1800	9	13	1	0	When asserted turns on the ADC_RESAMPLER block.
adc_fifo_bypass	D	1800	9	10	1	0	When asserted, the adc_fifo is bypassed. Input data is then clocked in directly using the RXCLK input. The ssel_rxin selection value will control the location of the internally generated sample clock when this bit is asserted.
resampler_decim	D	1800	9	9	1	1	This tells the ADC_RESAMPLER block the decimation factor (1=1.5X, 0=2X)
nz_pwr_mask	D	1820	6	0	16	0	Used along with rduz_sens_ena, it selects the noise bits to be added to the ADC input sample when asserted.

5.1.6 General Transmit Controls

These registers control the transmit output interface from the DUC channels.

VARIABLE NAME	TYPE	PAGE	ADDRESS	LSB POSITION	BIT WIDTH	DEFAULT	DESCRIPTION
tst_sel_slice	E	1C00	00	13	3	0	(TESTING PURPOSES) This selects the slice block that is generating the tst_out data. (which DUC/DDC)
tst_rate	E	1C00	05	11	5	0	The value here controls the output clock rate on the clk_tst pin. A value of 0 gives a full-rate output clock (txclk rate), a 1 gives half-rate output clock, a 3 gives 1/4th rate output clock, and so on. The number of txclk cycles for which the clk_tst signal is high + low = 1 + tst_rate.
interf_round	D	1C00	00	8	2	0	Controls round point on the transmit output data; {00 = 18b, 01=16b, 10=14b, 11=12b}. Rounded output data is MSB justified. For example, a 12b round point causes the output data to be presented on the output pins (17:6), and the output pins (5:0) to be held low.
interf_ena	D	1C00	00	4	4	15	Enables the individual transmit output busses 3 through 0. Disabled busses are always held low.
interf_interl	U	1C00	00	1	1	0	Enables interleaved I/Q data when asserted.
interf_real	U	1C00	00	0	1	1	Enables real only outputs when asserted. Complex data is output when cleared.
interf_scale_3	U	1C00	01	12	4	0	Selects the scaling between the sumchain output signals and the transmit output pins and transmit composite power meters. Appropriate limiting and rounding is performed as required by the programmed round point. Gain = $2^{(interf\_scale)}$. For sumchain 3.
interf_scale_2	U	1C00	01	8	4	0	Selects the scaling between the sumchain output signals and the transmit output pins and transmit composite power meters. Appropriate limiting and rounding is performed as required by the programmed round point. Gain = $2^{(interf\_scale)}$. For sumchain 2
interf_scale_1	U	1C00	01	4	4	0	Selects the scaling between the sumchain output signals and the transmit output pins and transmit composite power meters. Appropriate limiting and rounding is performed as required by the programmed round point. Gain = $2^{(interf\_scale)}$. For sumchain 1
interf_scale_0	U	1C00	01	0	4	0	Selects the scaling between the sumchain output signals and the transmit output pins and transmit composite power meters. Appropriate limiting and rounding is performed as required by the programmed round point. Gain = $2^{(interf\_scale)}$. For sumchain 0

comp_pmeter0_count_lsb	D	1C00	02	0	16	0	This is the number of sample sets to accumulate for a power measurement. Ia and Qa (signal) are each squared and accumulated. Each pair of I and Q are equal to one integration count. The accumulation interval is initiated when the sync is asserted and the programmed sync_delay has expired or when the interval start time is reached. When the integration count is reached, the accumulated powers are made available for MPU access and an interrupt is generated. Bits 0−15
comp_pmeter0_count_msb	D	1C00	03	0	5	0	Bits 16−20 of the number of sample sets to accumulate for a power measurement. (Used in conjunction with the previous variable.)
comp_pmeter0_sync_delay	D	1C00	03	7	9	0	Programmable start delay from sync, in eight output sample units.
comp_pmeter0_interval_lsb	D	1C00	04	0	16	0	This is the interval over which the integration is restarted and must be greater than the integration count. The interval start counter and RMS power accumulation is started at the sync pulse after the programmed delay and every time the interval counter reaches its limit. Bits 0−15
comp_pmeter0_interval_msb	D	1C00	05	0	5	0	Bits 16−20 of the interval over which the integration is restarted. (Used in conjunction with the previous variable.)
comp_pmeter1_count_lsb	D	1C00	06	0	16	0	See description for pmeter0
comp_pmeter1_count_msb	D	1C00	07	0	5	0	See description for pmeter0
comp_pmeter1_sync_delay	D	1C00	07	7	9	0	See description for pmeter0
comp_pmeter1_interval_lsb	D	1C00	08	0	16	0	See description for pmeter0
comp_pmeter1_interval_msb	D	1C00	09	0	5	0	See description for pmeter0
comp_pmeter2_count_lsb	D	1C00	0A	0	16	0	See description for pmeter0
comp_pmeter2_count_msb	D	1C00	0B	0	5	0	See description for pmeter0
comp_pmeter2_sync_delay	D	1C00	0B	7	9	0	See description for pmeter0
comp_pmeter2_interval_lsb	D	1C00	0C	0	16	0	See description for pmeter0
comp_pmeter2_interval_msb	D	1C00	0D	0	5	0	See description for pmeter0
comp_pmeter3_count_lsb	D	1C00	0E	0	16	0	See description for pmeter0
comp_pmeter3_count_msb	D	1C00	0F	0	5	0	See description for pmeter0
comp_pmeter3_sync_delay	D	1C00	0F	7	9	0	See description for pmeter0
comp_pmeter3_interval_lsb	D	1C00	10	0	16	0	See description for pmeter0
comp_pmeter3_interval_msb	D	1C00	11	0	5	0	See description for pmeter0
duc_counter_lsb	D	1C00	1A	0	16	65535	32-bit interval timer common to all DUC sync inputs. This timer may be programmed to any interval count, and each DUC synchronization input can select this counter as a source. This counter increments on every TXCLK rising edge. Bits 0−15
duc_counter_msb	D	1C00	1B	0	16	65535	Bits 16−31 of the above mentioned 32-bit interval timer.
ssel_duc_counter	U	1C00	1C	8	3	0	Selects the sync source for the DUC sync counter.

duc_counter_width	D	1C00	1C	0	8	0	Sets the width of the counter generated sync pulse in TX clock cycles, from 1 to 256. The width of this pulse must be long enough to be captured by the slowest block to use the DUC counter sync.
ssel_comp_pmeter_0	U	1C00	1D	12	3	0	Selects the sync source for composite power meter 0.
ssel_comp_pmeter_1	U	1C00	1D	8	3	0	Selects the sync source for composite power meter 1.
ssel_comp_pmeter_2	U	1C00	1D	4	3	0	Selects the sync source for composite power meter 2.
ssel_comp_pmeter_3	U	1C00	1D	0	3	0	Selects the sync source for composite power meter 3.
ssel_txsync_out	U	1C00	1E	0	3	0	Selects the sync source for the TXSYNC_OUT pin.

5.1.7 DDC or DUC Channel Controls

These controls are used by both the DDC or DUC channels. These follow either the *ddc <channel_number>* or the *duc <channel_number>* keywords in the cmd5316 configuration file.

VARIABLE NAME	TYPE	PAGE	ADDRESS	LSB POSITION	BIT WIDTH	DEFAULT	DESCRIPTION
cdma_mode	U	0100	0	15	1	1	When asserted the block is in the dual channel CDMA2000 mode.
crastarttap_pfir	C	0100	0	8	5	0	These bits define the number of taps that PFIR uses for the filtering. Another way of looking at these bits is that this value is the location in the RAM of the center tap. DUC PFIR: (2 x crastarttap_pfir) +1, DDC PFIR: 4(crastarttap_pfir+1), DDC PFIR long mode: 8(crastarttap_pfir+1). Note: crastarttap_pfir must be odd for a DUC
crastarttap_cfir	C	0100	0	3	5	0	These bits define the number of taps that CFIR uses for the filtering. DUC CFIR: (2 x crastarttap_cfir) +1, DDC CFIR: 2(crastarttap_cfir+1). Note: crastarttap_cfir must be odd for a DUC
pfir_gain	U	0100	1	13	3	0	This is the gain for the PFIR. The range is from 2e−19 to 2e−12 for the receive PFIR. (“000” = 2e−19 and “111” = 2e−12) For the transmit PFIR however, only the LSB of the word is used and it selects either 2e−18 when ‘0’ or 2e−17 when ‘1’.
cfir_gain	U	0100	1	5	1	0	This is the gain for the CFIR. 0= 2e−19, 1= 2e−18.
cic_scale_a	U	0100	0E	11	5	0	This sets the gain shift at the output of the CDMA A channel (or UMTS channel) CIC. 0x00 is no shift, each increment by 1 increases the signal amplitude by 2X.
cic_scale_b	U	0100	0E	6	5	0	This sets the gain shift at the output of the CDMA B channel CIC. 0x00 is no shift, each increment by 1 increases the signal amplitude by 2X.
cic_interp_decim	U	0100	0E	0	5	24	Sets the CIC interpolation, where interpolation is cic_interp_decim + 1 in the digital up converters. Sets the CIC decimation, where decimation is cic_interp_decim + 1 in the digital down converters.

cic_m2_ena_a	D	0100	0F	10	6	0	Programs the CDMA A channel (or UMTS channel) CIC fir sections M value to 2 when set, 1 when cleared. cic_m2_ena_a(0) controls the M value for the first comb section and cic_m2_ena_a(5) controls the M value for the last comb section.
cic_m2_ena_b	D	0100	0F	4	6	0	Programs the CDMA B channel CIC fir sections M value to 2 when set, 1 when cleared. cic_m2_ena_b(0) controls the M value for the first comb section and cic_m2_ena_b(5) controls the M value for the last comb section.
tadj_offset_coarse_a	D	0100	11	10	6	0	This is part of the time delay adjust. This is the coarse offset and is really an offset from the write address in the delay ram. This value affects the A channel if CDMA mode is being used, or the UMTS channel. Each LSB is one more offset between input to the course delay block and the output of the course block.
tadj_offset_coarse_b	D	0100	11	4	6	0	This is part of the time delay adjust. This is the coarse offset and is really an offset from the write address in the delay ram. This value affects the B channel if CDMA mode is being used. Each LSB is one more offset between input to the course delay block and the output of the course block.
tadj_offset_fine_a	D	0100	12	13	3	0	This is part of the time delay adjust. This is the fine adjust value. It adjusts the time delay at the clock rate. This value affects the A channel if CDMA mode is being used, or the UMTS channel.
tadj_offset_fine_b	D	0100	12	10	3	0	This is part of the time delay adjust. This is the fine adjust value. It adjusts the time delay at the clock rate. This value affects the B channel if CDMA mode is being used.
tadj_interp_decim	U	0100	12	7	3	1	This is the decimation or interpolation value for the fine time adjust block. Decimation or interpolation can be from 1 to 8. This value affects both the A and B channels if CDMA mode is being used, or the UMTS channel.
phase_add_a_lsb	C	0100	13	0	16	0	This 32 bit word is used to control the frequency of the NCO. Derived from the keyword freqa by cmd5316. (for CDMA channel A or UMTS channel). Lower 16 bits.
phase_add_a_msb	C	0100	14	0	16	0	Upper 16 bits of the above 32-bit word.
phase_add_b_lsb	C	0100	15	0	16	0	This 32-bit word is used to control the frequency of the NCO. Derived from the keyword freqb by cmd5316. (for CDMA channel B). Lower 16 bits.
phase_add_b_msb	C	0100	16	0	16	0	Upper 16 bits of the above 32-bit word.
phase_offset_a	D	0100	17	0	16	0	This is the fixed phase offset added to the output of the frequency accumulator for sinusoid generation in the NCO. (UMTS mode and A channel in CDMA mode)
phase_offset_b	D	0100	18	0	16	0	This is the fixed phase offset added to the output of the frequency accumulator for sinusoid generation in the NCO for CDMA B channel.
dither_ena	D	0100	19	15	1	0	This bit controls whether or not dither is turned on(1) or off(0).

test_bits_1	E	0100	19	13	2	0	TEST BITS. Set to ’0’ for normal operation.
pmeter_sync_disable	D	0100	19	12	1	0	Turns off the sync to the channel power meter. This can be used to individually turn off syncs to a channels power meter, while still having syncs to other power meters on the chip.
ddc_duc_ena	U	0100	19	11	1	0	When set this turns on the DUC or DDC. When unset, the clocks to this block are turned off.
mixer_gain	U	0100	19	9	1	0	Adds a fixed −6 dB of gain to the mixer output(before round and limiting) when asserted. Else adds −12-dB gain when deasserted.
mpu_ram_read	E	0100	19	8	1	0	(TESTING PURPOSES) Allows the coefficient RAMs in the PFIR/CFIR to be read out the mpu data bus. This cannot be done during normal operation and must be done when the state of the output data is not important. THIS BIT MUST BE SET ONLY DURING THE READ OPERATION.
sumchn_sel_b	U	0100	19	4	4	2	This word controls the second set of additions for the CDMA B signal in the sumchn output. The selection bits are not mutually exclusive.
sumchn_sel_a	U	0100	19	0	4	1	This word controls the first set of additions for the CDMA A signal (or UMTS signal) in the sumchn output. The selection bits are not mutually exclusive.
tst_sel_block	E	0100	1A	0	6	0	(TESTING PURPOSES) This is the selection of which signal comes out the test bus. When a constant ‘0’ is selected this also reduces power by preventing the data at the input of the test block from changing. It does not stop the clock however.
ssel_pmeter	U	0120	0B	8	3	0	Selects the sync source for the channel power meter.
ssel_serial	U	0120	0B	0	3	0	Selects the sync source for the DUC and DDC serial interface state machines.
ssel_tadj_fine	U	0120	0C	12	3	0	Selects the sync source for the fine time adjust decimation(DUC) or zero stuff(DDC) moment.
ssel_tadj_coarse	U	0120	0C	8	3	0	Selects the sync source for the course time adjust delay selection.
ssel_gain	U	0120	0C	4	3	0	Selects the sync source for the DUC gain register or DDC AGC gain register.
ssel_nco	U	0120	0D	12	3	0	Selects the sync source for the NCO accumulator reset.
ssel_dither	U	0120	0D	8	3	0	Selects the sync source for the NCO phase dither generator reset.
ssel_freq	U	0120	0D	4	3	0	Selects the sync source for the NCO frequency register.
ssel_phase	U	0120	0D	0	3	0	Selects the sync source for the NCO phase offset register.

5.1.8 DDC Channel Controls

These controls are used by the DDC channels. These follow the *ddc <channel_number>* keyword in the cmd5316 configuration file.

VARIABLE NAME	TYPE	PAGE	ADDRESS	LSB POSITION	BIT WIDTH	DEFAULT	DESCRIPTION
pmeter_integration_ddc	D	0100	2	0	16	0	This is the number of four sample sets to accumulate for a power measurement. In CDMA mode, one sample set is the I and Q of the signal and diversity. Ia and Qa (signal) are each squared and accumulated and Ib and Qb (diversity) are squared and accumulated. In UMTS mode, each I and Q pair are squared and accumulated. Four samples are equal to one integration count. The count is initiated when the sync is asserted or when the interval start time is reached. When the integration count is reached, the accumulated powers are made available for MPU access and an interrupt is generated.
pmeter_sync_delay_ddc	D	0100	3	8	8	0	The delay from selected sync source to when the power calculation starts.
pmeter_interval_ddc	D	0100	3	0	8	0	The start interval timer is the interval over which the integration is restarted and must be greater than the integration count. The interval start counter and RMS power accumulation is started at the sync pulse after the programmed delay and every time the interval counter reaches its limit. This value is in 1024 sample units.
cic_gain_ddc	U	0100	0E	5	1	0	Adds a fixed gain of 12 dB at the CIC output when asserted.
test_ena	E	0100	19	10	1	0	TEST BIT. Set to '0' for normal operation.
agc_dbelow	D	0100	1D	12	4	0	The value to shift the gain that is then added to the accumulator when the value of the incoming data x current gain value is below the Threshold.
agc_dabove	D	0100	1D	8	4	0	The value to shift the gain that is then subtracted from the accumulator when the value of the incoming data x the current gain value is above the Threshold.
agc_dzero	D	0100	1D	4	4	0	The value to shift the gain that is then added to the accumulator when the value of the incoming data x current gain values consistently equal to zero.
agc_dsat	D	0100	1D	0	4	0	The value to shift the gain that is then subtracted form the accumulator when the value of the incoming data x the current gain value is consistently equal to maximum.
agc_zero_msk	D	0100	1E	12	4	0	Masks the lower 4 bits of the magnitude of the input signal so that they are counted as zeros.
agc_rnd	D	0100	1E	8	4	0	Determines where to round the output of the AGC. 0000 is 18 bits are out. The number of bits out of the agc is 18 − agc_rnd.
agc_thres	D	0100	1E	0	8	0	This is the threshold that the data x gain is compared to. This value is compared to the magnitude of the upper eight bits of the agc output. (Input x gain).

agc_gaina_msb	U	0100	1F	13	3	0	Upper 3 bits of the CDMA channel A (or UMTS) gain value.
agc_freeze	U	0100	1F	12	1	1	Keeps the agc from adapting and only multiplies the input data by the programmed gain. Should be asserted when the AGC algorithm is to be bypassed.
agc_max_cnt	D	0100	1F	8	4	0	when the agc_output (input x gain) is at full scale for this number of times then the gain shift value is changed to D3.
agc_gainb_msb	U	0100	1F	5	3	0	Upper 3 bits of the CDMA channel B gain value.
agc_clear	U	0100	1F	4	1	0	Clears the AGC accumulator. Should assert this when the AGC is in bypass mode.
agc_zero_cnt	D	0100	1F	0	4	0	When the agc_output (input x gain) is zero value for this number of times then the gain shift value is changed to agc_dzero.
agc_gaina_lsb	U	0120	0	0	16	4096	This is the lower 16 bits of the total 19 bits of programmable gain. The gaina value is always positive with the upper 7 bits being the integer value and the lower 12 bits being the fractional. This gain value is used for all UMTS operations and for channel A data when in CDMA mode. This holds the lower four integer bits and the 12 fractional bits. The upper 3 integer bits are stored in the agc_gaina_msb variable. A value of 0001000000000000 is unity gain.
agc_gainb_lsb	U	0120	1	0	16	4096	This is the lower 16 bits of the total 19 bits of programmable gain. The gainb value is always positive with the upper 7 bits being the integer value and the lower 12 bits being the fractional. This gain value is used for channel B data when in CDMA mode. This holds the lower four integer bits and the 12 fractional bits. The upper 3 integer bits are stored in the agc_gainb_msb variable. A value of 0001000000000000 is unity gain.
agc_amax	D	0120	2	0	16	512	The maximum value that gain can be adjusted up to. The top 7 bits are integer and bottom the 9 bits are fractional.
agc_amin	D	0120	3	0	16	512	The minimum value that gain can be adjusted down to. The top 7 bits are integer and the bottom 9 bits are fractional.
pser_recv_fsinvl	U	0120	4	8	7	25	Receive serial interface frame sync interval in bit clocks.
pser_recv_bits	U	0120	4	0	5	17	Number of output bits per sample−1; for 18 bits, this is set to {10001}.
pser_recv_clkdiv	U	0120	5	12	4	1	Receive serial interface clock divider rate−1; 0 is full rate and 15 divides the clock by 16. For example, to run the receive serial interface at 1/4 the receive clock, set pser_recv_clkdiv(3:0) = 0011.

pser_recv_8pin	D	0120	5	7	1	0	When set, four pins are used for I and four pins for Q in UMTS mode. When cleared, two pins are used for I and two pins for Q. This is used in combination with the pser_recv_alt bit. When this bit is set, it would be set in two adjacent DDC channels; one would also set the pser_recv_alt bit. This causes the I channel to be serialized on four pins and the Q channel to be serialized on the adjacent channels four pins.
pser_recv_alt	D	0120	5	6	1	0	When set, this channel's receive serial interface outputs the Q data from the adjacent DDC channel. (set to 0 for even DDC and to 1 for ODD DDC)
pser_recv_fsdel	D	0120	5	0	2	1	Delay between the receive frame sync output and the MSB of serial data {3, 2, 1, 0}.
ddcmux_sel_a	U	0120	6	12	4	0	Controls which samples go to the mixer for I/Q. (for CDMA channel A or UMTS channel).
ddcmux_sel_b	U	0120	6	4	4	0	Controls which samples go to the mixer for I/Q. (for CDMA channel B).
gain_mon	D	0120	6	10	1	0	Combines the gain with the I/Q output signals when asserted. Look at the AGC description for more info about the status bits.
rnd_disable	D	0120	6	11	1	1	Turns off rounding at the AGC output if set. Normal AGC output otherwise.
ch_rate_sel	U	0120	6	8	2	0	Tells the DDC what the input clock rate for the channel is. 0 − rxclk, 1 – rxclk/2, 2 – rxclk/4, 3 – rxclk/8. For example, if the resampler_ena =1, the output of the resampler block is at rxclk/2 rate. So ch_rate_sel should be set to 1.
remix_only	U	0120	6	3	1	0	Assert this when only real input is available at the DDC's mixer inputs. This bit holds the Q portion of the signal to 0.
cic_bypass	D	0120	6	2	1	0	(TESTING PURPOSES) If asserted then the data from the rxin_a and rxin_b are fed directly into the cfir input as I and Q respectively. rxin_a(0) also functions as the sync_cfir signal and should rise at the beginning of input data.
double_tap	D	0120	6	0	2	0	Set to 0 for normal mode. In double tap mode, data out of the last PFIR ram in the main DDC (even numbered DDC) is sent to the adjacent secondary DDC (odd numbered DDC) PFIR as input thus forming a 128-tap delay line. Also data received from the secondary PFIR summers is added into the Main DDC's PFIR sum to form the output. This enables using a PFIR of length up to 128 instead of 64 as in the normal mode. When using double tap mode, set double_tap to 2 for the main (even) DDC and to 1 for the secondary (odd) DDC.
ssel_cic	U	0120	0B	12	3	0	Selects the sync source for the DDC CIC filter decimation moment. No effect for DUC.

5.1.9 DUC Channel Controls

These controls are used by the DUC channels. These follow the *duc <channel_number>* keyword in the cmd5316 configuration file.

VARIABLE NAME	TYPE	PAGE	ADDRESS	LSB POSITION	BIT WIDTH	DEFAULT	DESCRIPTION
symmetric_pfir	C	0100	00	14	1	0	When asserted the block's PFIR is symmetric. DUC only
symmetric_cfir	C	0100	00	13	1	0	When asserted the block's CFIR is symmetric. DUC only
pmeter_integration_duc	D	0100	02	0	13	0	This is the number of four sample sets to accumulate for a power measurement. In CDMA mode, one sample set is the I and Q of the signal and diversity. Ia and Qa (signal) are each squared and accumulated and Ib and Qb (diversity) are squared and accumulated. In UMTS mode, each I and Q pair are squared and accumulated. Four samples are equal to one integration count. The count is initiated when the sync is asserted or when the interval start time is reached. When the integration count is reached, the accumulated powers are made available for MPU access and an interrupt is generated.
pmeter_sync_delay_duc	D	0100	03	9	7	0	The delay from selected sync source to when the power calculation starts.
pmeter_interval_duc	D	0100	03	0	9	0	The start interval timer is the interval over which the integration is restarted and must be greater than the integration count. The interval start counter and RMS power accumulation is started at the sync pulse after the programmed delay and every time the interval counter reaches its limit. This value is in 64 sample units.
pilot_gain_0	D	0100	04	0	16	0	Pilot channel gain word, aligned with MSB of the input data. 0xFFFF generates a full scale complex pilot signal added to the user signal. Setting the gain to 0x0000 causes no pilot signal to be added. Only valid for UMTS, should be set to 0x0000 for CDMA.
pilot_gain_1	D	0100	05	0	16	0	This value MUST be set to the same value as pilot_gain_0.
pilot_psc_lsb	D	0100	06	0	16	1	The lower 16 bits of the 18-bit pilot X LFSR initial value. This 18b word is loaded on pilot sync event. The value loaded here that corresponds to 3gpp primary scrambling code (PSC) 0 is 0x00001. Users must calculate the correct initial value to implement the other 511 PSCs.
pilot_psc_msb	D	0100	07	14	2	0	The upper 2 bits of the 18-bit pilot X LFSR initial value. This 18b word is loaded on pilot sync events. The value loaded here that corresponds to 3gpp primary scrambling code (PSC) 0 is 0x00001. Users must calculate the correct initial value to implement the other 511 PSCs.
pilot_diversity	D	0100	07	13	1	0	Select between main and diversity pilot symbol generation. (0=main, 1=diversity)
pilot_delay	D	0100	08	0	16	0	Unsigned delay value in chips from the pilot sync event, from 0 to 38399 chips.

gainfora	U	0100	0C	0	16	8192	This is the unsigned gain that is multiplied with the CDMA channel A or UMTS channel input signal. The gain multiply is calculated as gainfora/8192.
gainforb	U	0100	0D	0	16	8192	This is the unsigned gain that is multiplied with the CDMA channel B input signal.
cic_auto_flush_dis	E	0100	10	12	4	0	Disables the automatic flush feature in the CIC accumulators.
cic_flush_test	E	0100	10	8	4	0	Forces an overflow detection in the CIC only on a rising edge of this bit, therefore it must be programmed to '0' and then back to '1' for the edge to occur.
cic_flush_clear	E	0100	10	4	4	0	Clears an overflow error manually when set, again only on a rising edge does this occur.
serp_tran_bits	U	0100	1B	11	5	17	Number of input bits per sample−1; for 18 bits, this is set to {10001}.
serp_tran_fsdel	D	0100	1B	8	2	1	Delay between frame sync output and MSB of serial data {3, 2, 1, 0}.
serp_tran_4pin	D	0100	1B	7	1	0	Selects 2-pin mode when cleared and 4-pin mode when set.
serp_tran_fsinvl	U	0100	1B	0	7	50	Transmit serial interface frame sync interval in bit clocks.
serp_tran_clkdiv	U	0100	1C	0	4	1	Transmit serial interface clock divider rate−1; 0 is full rate, and 15 divides the clock by 16. For example, to run the serial interface at 1/4 the transmit clock, set serp_tran_clkdiv(3:0) = 0011.
ssel_pilot	U	0120	0B	4	3	0	Selects the sync source for the DUC pilot code generator.

6 GC5316 Pin Description

6.1 Transmit Section Signals

SIGNAL NAME	BALL DESIG	TYPE	DESCRIPTION
txclk	K26	input	Transmit clock input
txin_0_a	T23	input	DUC 0 serial in data. CDMA A: I/Q UMTS: I
txin_1_a	U25	input	DUC 1 serial in data. CDMA A: I/Q UMTS: I
txin_0_b	T24	input	DUC 0 serial in data. CDMA B: I/Q UMTS: Q
txin_1_b	U26	input	DUC 1 serial in data. CDMA B: I/Q UMTS: Q
txin_2_a	W26	input	DUC 2 serial in data. CDMA A: I/Q UMTS: I
txin_3_a	V25	input	DUC 3 serial in data. CDMA A: I/Q UMTS: I
txin_2_b	U24	input	DUC 2 serial in data. CDMA B: I/Q UMTS: Q
txin_3_b	V26	input	DUC 3 serial in data. CDMA B: I/Q UMTS: Q
txin_4_a	Y26	input	DUC 4 serial in data. CDMA A: I/Q UMTS: I
txin_5_a	W25	input	DUC 5 serial in data. CDMA A: I/Q UMTS: I
txin_4_b	V24	input	DUC 4 serial in data. CDMA B: I/Q UMTS: Q
txin_5_b	U23	input	DUC 5 serial in data. CDMA B: I/Q UMTS: Q
txin_6_a	W23	input	DUC 6 serial in data. CDMA A: I/Q UMTS: I
txin_7_a	AA26	input	DUC 7 serial in data. CDMA A: I/Q UMTS: I
txin_6_b	Y25	input	DUC 6 serial in data. CDMA B: I/Q UMTS: Q
txin_7_b	W24	input	DUC 7 serial in data. CDMA B: I/Q UMTS: Q
txin_8_a	Y23	input	DUC 8 serial in data. CDMA A: I/Q UMTS: I

txin_9_a	AB26	input	DUC 9 serial in data. CDMA A: I/Q UMTS: I
txin_8_b	AA25	input	DUC 8 serial in data. CDMA B: I/Q UMTS: Q
txin_9_b	Y24	input	DUC 9 serial in data. CDMA B: I/Q UMTS: Q
txin_10_a	AA23	input	DUC 10 serial in data. CDMA A: I/Q UMTS: I
txin_11_a	AC26	input	DUC 11 serial in data. CDMA A: I/Q UMTS: I
txin_10_b	AB25	input	DUC 10 serial in data. CDMA B: I/Q UMTS: Q
txin_11_b	AA24	input	DUC 11 serial in data. CDMA B: I/Q UMTS: Q
tx_sync_out_0	AB24	output	Transmit serial interface strobe for DUC 0,1 (txin_[0,1]_[a,b])
tx_sync_out_1	AC25	output	Transmit serial interface strobe for DUC 2,3 (txin_[2,3]_[a,b])
tx_sync_out_2	AD26	output	Transmit serial interface strobe for DUC 4,5 (txin_[4,5]_[a,b])
tx_sync_out_3	AB23	output	Transmit serial interface strobe for DUC 6,7 (txin_[6,7]_[a,b])
tx_sync_out_4	AC24	output	Transmit serial interface strobe for DUC 8,9 (txin_[8,9]_[a,b])
tx_sync_out_5	AD23	output	Transmit serial interface strobe for DUC 10,11 (txin_[10,11]_[a,b])
tx_synca	K24	input	Transmit sync input
tx_syncb	J25	input	Transmit sync input
tx_syncc	H26	input	Transmit sync input
tx_syncd	K23	input	Transmit sync input
tx_sync_out	F23	output	Transmit general purpose output sync
txclk_out	E23	output	Transmit output clock
tx_i_flag	D24	output	Transmit output iflag
txout_a_17	B19	output	Transmit output bus a MSB
txout_a_16	A20	output	Transmit output bus a
txout_a_15	C19	output	Transmit output bus a
txout_a_14	B20	output	Transmit output bus a
txout_a_13	A21	output	Transmit output bus a
txout_a_12	D19	output	Transmit output bus a
txout_a_11	C20	output	Transmit output bus a
txout_a_10	B21	output	Transmit output bus a
txout_a_9	A22	output	Transmit output bus a
txout_a_8	D20	output	Transmit output bus a
txout_a_7	C21	output	Transmit output bus a
txout_a_6	B22	output	Transmit output bus a
txout_a_5	A23	output	Transmit output bus a
txout_a_4	C22	output	Transmit output bus a
txout_a_3	B23	output	Transmit output bus a
txout_a_2	A24	output	Transmit output bus a
txout_a_1	D22	output	Transmit output bus a
txout_a_0	C23	output	Transmit output bus a LSB
txout_b_17	B14	output	Transmit output bus b MSB
txout_b_16	C14	output	Transmit output bus b
txout_b_15	D14	output	Transmit output bus b
txout_b_14	A15	output	Transmit output bus b
txout_b_13	B15	output	Transmit output bus b
txout_b_12	C15	output	Transmit output bus b
txout_b_11	A16	output	Transmit output bus b
txout_b_10	B16	output	Transmit output bus b
txout_b_9	A17	output	Transmit output bus b
txout_b_8	C16	output	Transmit output bus b

txout_b_7	B17	output	Transmit output bus b
txout_b_6	D16	output	Transmit output bus b
txout_b_5	A18	output	Transmit output bus b
txout_b_4	C17	output	Transmit output bus b
txout_b_3	B18	output	Transmit output bus b
txout_b_2	A19	output	Transmit output bus b
txout_b_1	D17	output	Transmit output bus b
txout_b_0	C18	output	Transmit output bus b LSB
txout_c_17	C9	output	Transmit output bus c MSB
txout_c_16	D10	output	Transmit output bus c
txout_c_15	A8	output	Transmit output bus c
txout_c_14	B9	output	Transmit output bus c
txout_c_13	C10	output	Transmit output bus c
txout_c_12	A9	output	Transmit output bus c
txout_c_11	D11	output	Transmit output bus c
txout_c_10	B10	output	Transmit output bus c
txout_c_9	C11	output	Transmit output bus c
txout_c_8	A10	output	Transmit output bus c
txout_c_7	B11	output	Transmit output bus c
txout_c_6	A11	output	Transmit output bus c
txout_c_5	C12	output	Transmit output bus c
txout_c_4	B12	output	Transmit output bus c
txout_c_3	A12	output	Transmit output bus c
txout_c_2	D13	output	Transmit output bus c
txout_c_1	C13	output	Transmit output bus c
txout_c_0	B13	output	Transmit output bus c LSB
txout_d_17	C4	output	Transmit output bus d MSB
txout_d_16	D5	output	Transmit output bus d
txout_d_15	A3	output	Transmit output bus d
txout_d_14	B4	output	Transmit output bus d
txout_d_13	C5	output	Transmit output bus d
txout_d_12	A4	output	Transmit output bus d
txout_d_11	B5	output	Transmit output bus d
txout_d_10	C6	output	Transmit output bus d
txout_d_9	D7	output	Transmit output bus d
txout_d_8	A5	output	Transmit output bus d
txout_d_7	B6	output	Transmit output bus d
txout_d_6	C7	output	Transmit output bus d
txout_d_5	D8	output	Transmit output bus d
txout_d_4	A6	output	Transmit output bus d
txout_d_3	B7	output	Transmit output bus d
txout_d_2	C8	output	Transmit output bus d
txout_d_1	A7	output	Transmit output bus d
txout_d_0	B8	output	Transmit output bus d LSB

6.2 Receive Section Signals

SIGNAL NAME	BALL DESIG	TYPE	DESCRIPTION
rxclk	L24	input	Receive clock input
adcclk	D3	input	adc input clock
rxin_a_15	E4	input	Receive input data bus a MSB
rxin_a_14	C1	input	Receive input data bus a
rxin_a_13	D2	input	Receive input data bus a
rxin_a_12	E3	input	Receive input data bus a
rxin_a_11	F4	input	Receive input data bus a
rxin_a_10	D1	input	Receive input data bus a
rxin_a_9	E2	input	Receive input data bus a
rxin_a_8	F3	input	Receive input data bus a
rxin_a_7	G4	input	Receive input data bus a
rxin_a_6	E1	input	Receive input data bus a
rxin_a_5	F2	input	Receive input data bus a
rxin_a_4	G3	input	Receive input data bus a
rxin_a_3	H4	input	Receive input data bus a
rxin_a_2	F1	input	Receive input data bus a
rxin_a_1	G2	input	Receive input data bus a
rxin_a_0	H3	input	Receive input data bus a LSB
rxin_b_15	G1	input	Receive input data bus b MSB
rxin_b_14	H2	input	Receive input data bus b
rxin_b_13	J3	input	Receive input data bus b
rxin_b_12	K4	input	Receive input data bus b
rxin_b_11	H1	input	Receive input data bus b
rxin_b_10	J2	input	Receive input data bus b
rxin_b_9	K3	input	Receive input data bus b
rxin_b_8	J1	input	Receive input data bus b
rxin_b_7	L4	input	Receive input data bus b
rxin_b_6	K2	input	Receive input data bus b
rxin_b_5	L3	input	Receive input data bus b
rxin_b_4	K1	input	Receive input data bus b
rxin_b_3	L2	input	Receive input data bus b
rxin_b_2	M4	input	Receive input data bus b
rxin_b_1	L1	input	Receive input data bus b
rxin_b_0	M3	input	Receive input data bus b LSB
rxin_c_15	M2	input	Receive input data bus c MSB
rxin_c_14	M1	input	Receive input data bus c
rxin_c_13	N3	input	Receive input data bus c
rxin_c_12	N2	input	Receive input data bus c
rxin_c_11	P2	input	Receive input data bus c
rxin_c_10	P3	input	Receive input data bus c
rxin_c_9	P4	input	Receive input data bus c
rxin_c_8	R1	input	Receive input data bus c
rxin_c_7	R2	input	Receive input data bus c
rxin_c_6	R3	input	Receive input data bus c
rxin_c_5	T1	input	Receive input data bus c
rxin_c_4	R4	input	Receive input data bus c

rxin_c_3	T2	input	Receive input data bus c
rxin_c_2	U1	input	Receive input data bus c
rxin_c_1	T3	input	Receive input data bus c
rxin_c_0	U2	input	Receive input data bus c LSB
rxin_d_15	T4	input	receive input data bus MSB
rxin_d_14	V1	input	Receive input data bus d
rxin_d_13	U3	input	Receive input data bus d
rxin_d_12	V2	input	Receive input data bus d
rxin_d_11	W1	input	Receive input data bus d
rxin_d_10	U4	input	Receive input data bus d
rxin_d_9	V3	input	Receive input data bus d
rxin_d_8	W2	input	Receive input data bus d
rxin_d_7	Y1	input	Receive input data bus d
rxin_d_6	W3	input	Receive input data bus d
rxin_d_5	Y2	input	Receive input data bus d
rxin_d_4	AA1	input	Receive input data bus d
rxin_d_3	W4	input	Receive input data bus d
rxin_d_2	Y3	input	Receive input data bus d
rxin_d_1	AA2	input	Receive input data bus d
rxin_d_0	AB1	input	Receive input data bus d LSB
rx_synca	J24	input	Receive sync input
rx_syncb	H25	input	Receive sync input
rx_syncc	G26	input	Receive sync input
rx_syncd	H24	input	Receive sync input
rx_sync_out	AF7	output	Receive general purpose output sync
rx_sync_out_0	AF23	output	Receive serial interface strobe for DDC 0, 1 (rxout_[0,1]_[a−d])
rx_sync_out_1	AE20	output	Receive serial interface strobe for DDC 2, 3 (rxout_[2,3]_[a−d])
rx_sync_out_2	AF18	output	Receive serial interface strobe for DDC 4, 5 (rxout_[4,5]_[a−d])
rx_sync_out_3	AF15	output	Receive serial interface strobe for DDC 6, 7 (rxout_[6,7]_[a−d])
rx_sync_out_4	AD12	output	Receive serial interface strobe for DDC 8, 9 (rxout_[8,9]_[a−d])
rx_sync_out_5	AE9	output	Receive serial interface strobe for DDC 10,11 (rxout_[10,11]_[a−d])
rxout_0_a	AF22	output	DDC 0 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_0_b	AC20	output	DDC 0 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_0_c	AD21	output	DDC 0 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_0_d	AE22	output	DDC 0 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_1_a	AD22	output	DDC 1 serial out data. CDMA A: I data. UMTS: I_{msb}
rxout_1_b	AE23	output	DDC 1 serial out data. CDMA B: I data. UMTS: $I_{msb - 1}$
rxout_1_c	AF24	output	DDC 1 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_1_d	AC22	output	DDC 1 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_2_a	AD18	output	DDC 2 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_2_b	AE19	output	DDC 2 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_2_c	AF20	output	DDC 2 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_2_d	AD19	output	DDC 2 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_3_a	AF21	output	DDC 3 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_3_b	AC19	output	DDC 3 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_3_c	AD20	output	DDC 3 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_3_d	AE21	output	DDC 3 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_4_a	AF17	output	DDC 4 serial out data. CDMA A: I data UMTS: I_{msb}

rxout_4_b	AD16	output	DDC 4 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_4_c	AE17	output	DDC 4 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_4_d	AC16	output	DDC 4 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_5_a	AD17	output	DDC 5 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_5_b	AE18	output	DDC 5 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_5_c	AF19	output	DDC 5 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_5_d	AC17	output	DDC 5 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_6_a	AE13	output	DDC 6 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_6_b	AE14	output	DDC 6 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_6_c	AD14	output	DDC 6 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_6_d	AC14	output	DDC 6 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_7_a	AE15	output	DDC 7 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_7_b	AD15	output	DDC 7 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_7_c	AF16	output	DDC 7 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_7_d	AE16	output	DDC 7 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_8_a	AD11	output	DDC 8 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_8_b	AF10	output	DDC 8 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_8_c	AE11	output	DDC 8 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_8_d	AF11	output	DDC 8 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_9_a	AE12	output	DDC 9 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_9_b	AF12	output	DDC 9 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_9_c	AC13	output	DDC 9 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_9_d	AD13	output	DDC 9 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_10_a	AE8	output	DDC 10 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_10_b	AD9	output	DDC 10 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_10_c	AC10	output	DDC 10 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_10_d	AF8	output	DDC 10 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$
rxout_11_a	AD10	output	DDC 11 serial out data. CDMA A: I data UMTS: I_{msb}
rxout_11_b	AF9	output	DDC 11 serial out data. CDMA B: I data UMTS: $I_{msb - 1}$
rxout_11_c	AC11	output	DDC 11 serial out data. CDMA A: Q data UMTS: Q_{msb}
rxout_11_d	AE10	output	DDC 11 serial out data. CDMA B: Q data UMTS: $Q_{msb -1}$

6.3 Microprocessor Signals

SIGNAL NAME	BALL DESIG	TYPE	DESCRIPTION
d0	AD8	input/output	MPU register interface data bus LSB
d1	AE7	input/output	MPU register interface data bus
d2	AF6	input/output	MPU register interface data bus
d3	AC8	input/output	MPU register interface data bus
d4	AD7	input/output	MPU register interface data bus
d5	AE6	input/output	MPU register interface data bus
d6	AF5	input/output	MPU register interface data bus
d7	AC7	input/output	MPU register interface data bus
d8	AD6	input/output	MPU register interface data bus
d9	AE5	input/output	MPU register interface data bus
d10	AF4	input/output	MPU register interface data bus
d11	AD5	input/output	MPU register interface data bus
d12	AE4	input/output	MPU register interface data bus
d13	AF3	input/output	MPU register interface data bus
d14	AC5	input/output	MPU register interface data bus
d15	AD4	input/output	MPU register interface data bus MSB
a0	AB4	input	MPU register interface address bus LSB
a1	AD1	input	MPU register interface address bus
a2	AC2	input	MPU register interface address bus
a3	AB3	input	MPU register interface address bus
a4	AA4	input	MPU register interface address bus
a5	AC1	input	MPU register interface address bus MSB
rd_n	Y4	input	MPU register interface read – active low
wr_n	AA3	input	MPU register interface write – active low
ce_n	AB2	input	MPU register interface chip enable – active low
reset_n	R24	input	Chip reset – active low
interrupt	AC3	output	Chip interrupt

6.4 JTAG Signals

SIGNAL NAME	BALL DESIG	TYPE	DESCRIPTION
tdi	N23	input	JTAG test data in
tms	M26	input	JTAG test mode select
trst_n	M25	input	JTAG test reset (same as trst – the “_n” is for consistency − being active low)
tck	M24	input	JTAG test clock
tdo	L26	output	JTAG test data out

6.5 Factory Test and No Connect Signals

SIGNAL NAME	BALL DESIG	TYPE	NOTE
testmode0	R26	input	Do not connect
testmode1	P24	input	Do not connect
scanen	P25	input	Do not connect
aflag_tst	E24	output	Do not connect
sync_tst	D25	output	Do not connect
clk_tst	C26	output	Do not connect
fa002_scan	T26	input	Do not connect
fa002_clk	R23	input	Do not connect
fa002_out	T25	output	Do not connect
zero	N25	input	Do not connect
	F26, G24, G25, H23, L23	input	Tie each pin high through 100-Ω resistors to VPAD
	K25, M23, L25, N24, R25, D26, E25, E26, F24, F25, G23, J26	no connect	Do not connect

6.6 Power and Ground Signals

SIGNAL NAME	BALL DESIG	DESCRIPTION
GND	A1, A2, A13, A14, A25, A26, B1, B3, B24, B26, C2, C25, N1, N26, P1, P26, AD2, AD25, AE1, AE24, AE3, AE26, AF1, AF2, AF13, AF14, AF25, AF26, L11, L12, L13, L14, L15, L16, M11−M16, N11−N16, P11−P16, R11−R16, T11−T16	Ground
VCORE	B2, D4, N4, AC4, AE2, B25, D23, P23, AC23, AE25, C3, J4, V4, AD3, C24, J23, V23, AD24	Core power
VPAD	D6, D12, D18, AC6, AC12, AC18, D9, D15, D21, AC9, AC15, AC21	I/O power

6.7 Power Monitoring

SIGNAL NAME	BALL DESIG	DESCRIPTION
vcoremon	N24	These pins monitor the internal power distribution. They cannot carry significant current and should not be connected to normal power and ground. It is recommended that this pin be brought to a small probe point for future monitoring/debugging purposes.
gndmon	R25	It is recommended that this pin be brought to a probe point for future monitoring/debugging purposes.

6.8 JTAG

The JTAG standard for boundary scan testing is implemented for board testing purposes. Internal scan test is not be supported. Five device pins are dedicated for JTAG support: tdi, tdo, tms, tck, and trst_n. The BSDL file is available on the web.

7 Specifications

7.1 Absolute Maximum Ratings

PARAMETER	SYMBOL	MIN	MAX	UNITS
Pad ring supply voltage	V_{PAD}	−0.3	4	V
Core supply voltage	V_{CORE}	−0.3	1.8	V
Input voltage (undershoot and overshoot)	V_{IN}	−0.5	V_{PAD}+0.5	V
Clamp current for an input or output		−20	20	mA
Storage temperature	T_{stg}	−65	140	°C
Junction temperature	T_J		105	°C
Lead soldering temperature (10 seconds)			300	°C
ESD classification	Class 2 (Passed 2.5-kV HBM, 500-V CDM, 150-V MM)			
Moisture sensitivity	Class 4 (4 days floor life at 30°C/60%H)			
Reflow conditions	JEDEC standard, 240°C max			

CAUTION: Exceeding the absolute maximum ratings (min or max) may cause permanent damage to the part. These are stress only ratings and are not intended for operation.

7.2 Recommended Operating Conditions

PARAMETER	MIN	MAX	UNITS
Pad ring supply voltage, V_{PAD}	3	3.6	V
Core supply voltage, V_{CORE}	1.5	1.65	V
Supply voltage difference $V_{PAD} - V_{CORE}$		2	V
Temperature ambient, no air flow(1), T_A	−40	85	°C
Junction temperature(2), T_J		105	°C

(1) Chips specifications in Tables 6.4 and 6.5 are production tested to100°C case temperature. QA tests are performed at 85°C.
(2) Thermal management will be required for full rate operation, see the following table and Section 7.4. The circuit is designed for junction temperatures up to 125°C. Sustained operation at elevated temperatures reduces long-term reliability. Lifetime calculations based on maximum junction temperature of 105°C.

7.3 Thermal Characteristics

THERMAL CONDUCTIVITY	388 BGA	UNITS
	3 W	
Theta junction-to-ambient (still air), θ_{JA}	13.5	°C/W
Theta junction-to-ambient (2m/s estimated), θ_{JA2m}	9.3	°C/W
Theta junction-to-case, θ_{JC}	2.4	°C/W

(3) Air flow reduces θ_{JA} and is highly recommended.

7.4 Power Consumption

The maximum power consumption is a function of the operating mode of the chip. The cmd5316 estimates the typical power supply current for the chip in a specific configuration. The *AC Characteristics* table provides maximum current in a maximum configuration used in production test.

Current consumption on the pad supply is primarily due to the external loads and follows C x V x F. Internal loads are estimated at 2 pF per pin. Data outputs have a transition density of going from a zero to a one, once per four clocks, while clock outputs transition every cycle. The frame strobes consume negligible power due to the low transition frequency. In general:

Ipad = Σ DataPad/4 x C x F x V + Σ ClockPad x C x F x V

A worst case current would be all transmit and receive ports operating at 125 MHz.

Ipad = (1+(4 x 18 + 4 x 2 x 6)/4) x (C + 2pF) x Fout x Vpad = 31 x 22 pF x 125 MHz x 3.3 V = 280 mA.

A more typical application with two ports active would use roughly 150 mA.

7.5 DC Operating Conditions (−40°C to 85°C case unless otherwise noted)

PARAMETER		V_{PAD} = 3 V to 3.6 V			UNITS
		MIN	TYP	MAX	
V_{IL}	Voltage input low (4)			0.8	V
V_{IH}	Voltage input high (4)	2			V
V_{OL}	Voltage output low (4) (I_{OL} = 2 mA)			0.5	V
V_{OH}	Voltage output high (4) (I_{OH} = −2 mA)	2.4		V_{PAD}	V
$\lvert I_{PU} \rvert$	Pullup current (V_{IN} = 0 V) (tdi, tms, trst_n, reset_n) (nominal 20 µA) (4)	5		35	µA
$\lvert I_{PD} \rvert$	Pulldown current (V_{IN} = V_{PAD}) (all other inputs and bidirs) (nominal 20 µA) (4)	5		35	µA
$\lvert I_{IN} \rvert$	Leakage (V_{IN} = V_{PAD}) (tdi, tms, trst_n, reset_n) (4)			2	µA
	Leakage (V_{IN} = 0) (all other inputs and bidirs) (4)			2	
	Leakage (V_{IN} = 0 or V_{PAD}) (all outputs) (4)			2	
I_{CCQ}	Quiescent supply current, I_{CORE} (4)			8	mA
C_{IN}	Capacitance for inputs (5)		5		pF
C_{BI}	Capacitance for bidirectionals (5)		5		pF

NOTE: Voltages are measured at low speed. Output voltages are measured with the indicated current load.
NOTE: Currents are measured at nominal voltages, high temperature (100°C for production test, 85°C for QA).
(4) Each part is tested at 100°C case temperature for the given specification. Lots are sample tested at −40°C.
(5) Controlled by design and process and not directly tested.

7.6 AC Characteristics

PARAMETER		MIN	MAX	UNIT
F_{CK}	Clock frequency (adcclk, rxclk, txclk) in selected modes(6)(9)		125	MHz
F_{CK}	Clock frequency (adcclk, rxclk, txclk) unrestricted(6)		80	MHz
t_{ADCKL} t_{RXCKL} t_{TXCKL}	Clock low period (below V_{IL}) (adcclk, rxclk, txclk)(6)	3		ns
t_{ADCKH} t_{RXCKH} t_{TXCKH}	Clock high period (above V_{IH}) (adcclk, rxclk, txclk)(6)	3		ns
t_r, t_f	Clock rise and fall times (V_{IL} to V_{IH}) (adcclk, rxclk, txclk)(8)		2	ns
$t_{su(TX)}$	Input setup (txin_[0−11]_[a−b], tx_sync[a−d]) before txclk rises(6)	2.2		ns
$t_{su(RX)}$	Input setup (rx_sync[a−d]) before rxclk rises(6)	2.5		ns
$t_{su(RXB)}$	Input setup (rxin_[a−d]_[0−15]) before rxclk rises adc_fifo bypassed(6)	0.4		ns
$t_{su(AD)}$	Input setup (rxin[a−d]_[0−15]) before adcclk rises adc_fifo active(6)	2.2		ns
$t_{h(TX)}$	Input hold (txin_[0−11]_[a−b], tx_sync[a−d]) after txclk rises(6)	1.1		ns
$t_{h(RX)}$	Input hold (rx_sync[a−d]) after rxclk rises(6)	0.5		ns
$t_{h(RXB)}$	Input hold (rxin[a−d]_[0−15]) after rxclk rises adc_fifo bypassed(6)	3.5		ns
$t_{h(AD)}$	Input hold (rxin[a−d]_[0−15]) after adcclk rises adc_fifo active(6)	1		ns
$t_{d(TX)}$	Data output delay (tx_sync_out_[0−5], tx_iflag, txout_[a−d]_[0−17]) after txclk rises(6)		6.5	ns
$t_{d(RX)}$	Data output delay (rx_sync_out_[0−5], rxout_[0−11] _[a−d]) after rxclk rises(6)		6.5	ns
$t_{OH(TX)}$	Data output hold (tx_sync_out_[0−5], tx_iflag, txout_[a−d]_[0−17]) after txclk rises(6)	1.5		ns
$t_{OH(RX)}$	Data output hold (rx_sync_out_[0−5], rxout_[0−11] _[a−d]) after rxclk rises(6)	1.5		ns
F_{JCK}	JTAG clock frequency (tck)(6)		40	MHz
t_{JCKL}	JTAG clock low period (below V_{IL}) (tck)(6)	8		ns
t_{JCKH}	JTAG clock high period (above V_{IH}) (tck)(6)	8		ns
$t_{su(J)}$	JTAG input (tdi or tms) setup before tck goes high(6)	2		ns
$t_{h(J)}$	JTAG input (tdi or tms) hold time after tck goes high(6)	9		ns
$t_{d(J)}$	JTAG output (tdo) delay from falling edge of tck(6)		6	ns
$t_{su(UPA)}$	Microprocessor address setup to falling edge of controls(6)	2.5		ns
$t_{h(UPA)}$	Microprocessor address hold from rising edge of controls(6)	2		ns
$t_{su(UPD)}$	Microprocessor data setup to rising edge of controls during writes(6)	12		ns
$t_{h(UPD)}$	Microprocessor data hold from rising edge of controls during writes(6)	2.6		ns
t_h	Microprocessor data output hold from rising edge of controls (read)(7)	0		ns
$t_{d(UP)}$	Microprocessor data output delay from falling edge of controls (read)(6)		36	ns
t_{UPCKL}	Microprocessor control low time(6)	30		ns
t_{UPCKH}	Microprocessor control high time(6)	8.4		ns

NOTE: Timing is measured from the respective clock at V_{PAD} /2 to input or output at V_{PAD} /2. Output loading is a 50-Ω transmission line whose delay is calibrated out.

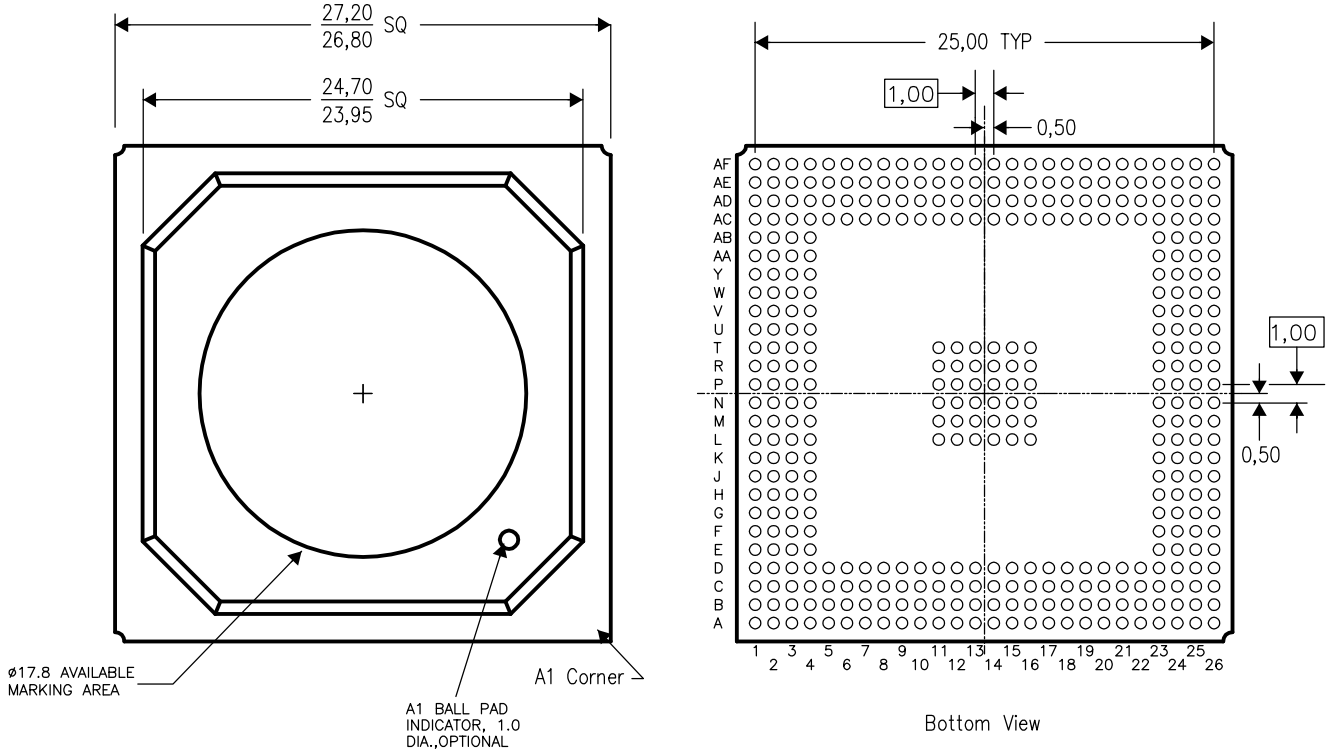
(6) Each part is tested at 90°C case temperature for the given specification. Lots are sample tested at −40°C.

(7) Controlled by design and process and not directly tested. Verified on initial part evaluation.

(8) Recommended practice.

(9) Excluding rx_sync_out , rx_sync_out_[1−5], tx_sync_out, tx_sync_out_[1−5]. Resampler active or adcclk < 80 MHz.

ZED (S–PBGA–N388) PLASTIC BALL GRID ARRAY



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. This drawing conforms to the JEDEC registered outline MS–034/A variation AAL–1.
D. This is a Lead–free package.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)
GC5316IZED	ACTIVE	BGA	ZED	388	40	Green (RoHS & no Sb/Br)	SNAGCU	Level-3-260C-168 HR

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265