GC5018 8-CHANNEL WIDEBAND RECEIVER

Check for Samples: GC5018

1 Introduction

1.1 FEATURES

- **Four 16-Bit CMOS ADC Input Ports**
- **Programmable Closed Loop VGA Control With 6-Bit Outputs for Each ADC Input Port**
- **Provide Received Total Wide Band Power (RTWP) Measurement for the Composite Power Across Carriers With Programmable Time Window for Measurement**
- **8 UMTS Digital Down Converter (DDC) Channels or 16 CDMA or 16 TD-SCDMA DDC Channels With Programmable 18 Bit Filter Coefficients**
- **Each DDC channel includes**
  - **Real or Complex DDC Inputs**
  - **115 dB SFDR NCO**
  - **UMTS Mode Rx Filtering: 6 Stage CIC (m=1 or 2), Up to 40 Tap CFIR, Up to 64 Tap PFIR**
  - **CDMA Mode Rx Filtering: 6 Stage CIC (m=1 or 2), Up to 64 Tap CFIR, Up to 64 Tap PFIR**
  - **Power Measurements**
  - **Final AGC**
- **1.5V Digital Core Supply, 3.3V Digital I/O Supply**
- **305 Ball Plastic BGA (19 mm x 19 mm) With 1.0 mm Pitch**
- **Power Dissipation: 2.5W**

1.2 APPLICATIONS

- **Wireless Base Station Receiver**
- **Multi-Carrier Digital Receiver**
- **UMTS (4 Carriers-1 Sector With Diversity)**
- **CDMA (8 Carriers-1 Sector With Diversity)**
- **TD-SCDMA (16 Carriers-1 Sector Without Diversity, 8 Carriers-1-Sector With Diversity)**
- **Digital Radio Receivers**
- **Wide Band Receivers**
- **Software Radios**
- **Wireless Local Loop**
- **Intelligent Antenna Systems**

1.3 DESCRIPTION

The GC5018 is a multi-channel communications signal processor that provides digital downconversion optimized for cellular base transceiver systems. The device supports UMTS, CDMA-1X and TD-SCDMA air interface cellular standards.

The chip provides up to 8 UMTS digital downconverter channels (DDC), 16 CDMA DDCs or 16 TD-SCDMA DDCs. The DDC channels are independent and operate simultaneously.

The GC5018 has four 16-bit inputs. Each DDC channel can be programmed to accept data from any one (or two for complex input mode) of the four input ports.

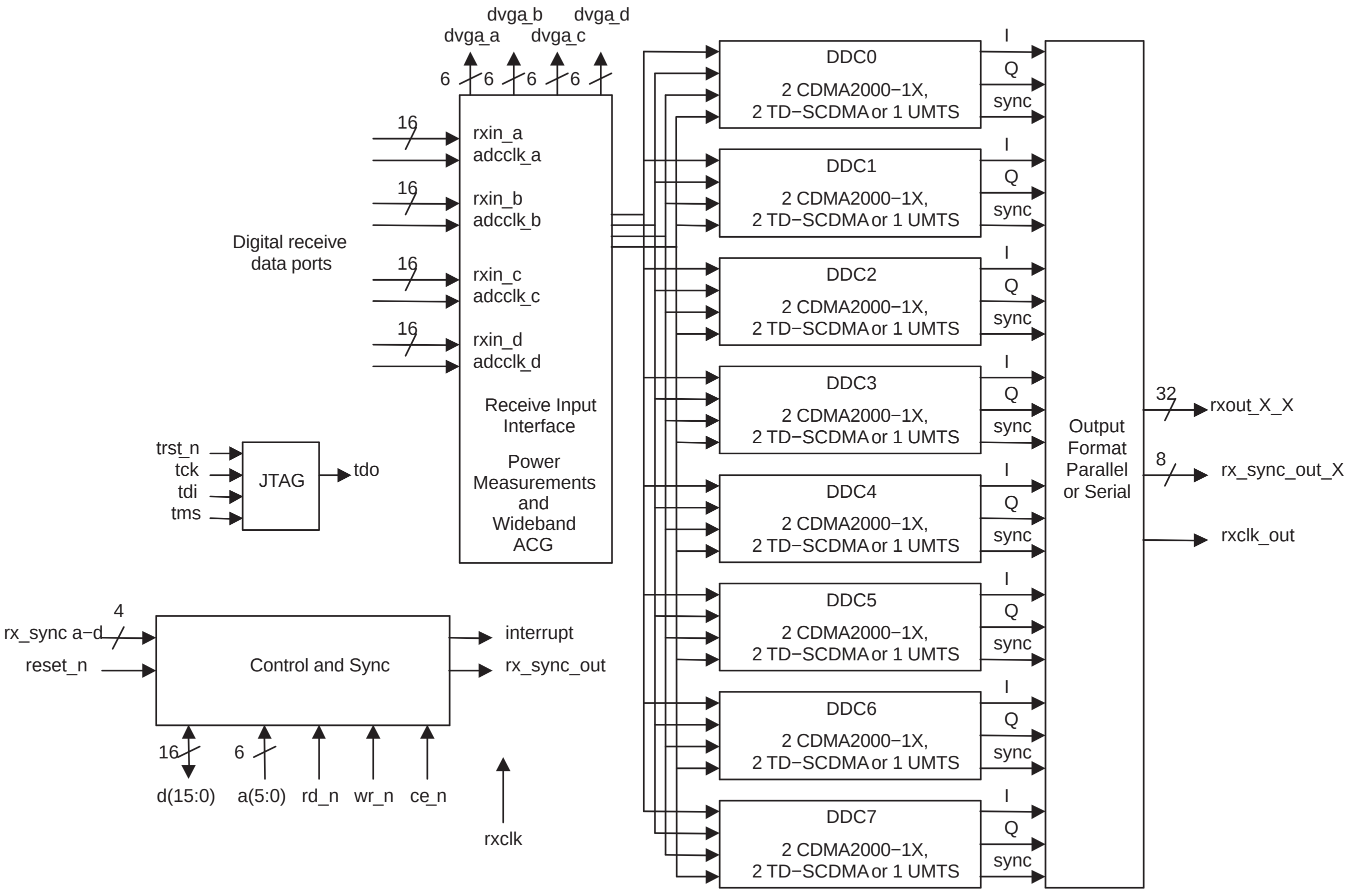


Figure 1-1. Functional Block Diagram

2 RECEIVE DIGITAL SIGNAL PROCESSING

The down conversion section of the GC5018 consists of the receive input interface, the rx_distribution bus, and 8 digital downconverter blocks.

The purpose of the receive input interface is to accept signal data from four 16 bit input ports, measure the input signal power, control the digital VGA and to distribute the data to the DDC blocks. The input interface also has a user-controlled test generator and noise source.

The rx_distribution bus distributes the four channels of signal data to each of the 8 DDC blocks.

Each DDC block selects one of the four channels (or 2 for complex input data) from the rx_distribution bus and then performs downconversion tuning, programmable delay, channel filtering with decimation, power measurement, fixed gain adjust and/or automatic gain control. Each DDC block can support 1 UMTS channel, 2 CDMA channels or 2 TD-SCDMA channels. An optional mode permits stacking two DDC blocks in UMTS mode to provide double-length final pulse shaping filtering.

Tuned, filtered, and decimated signal data is output in bit serial or parallel format.

2.1 Receive Input Interface

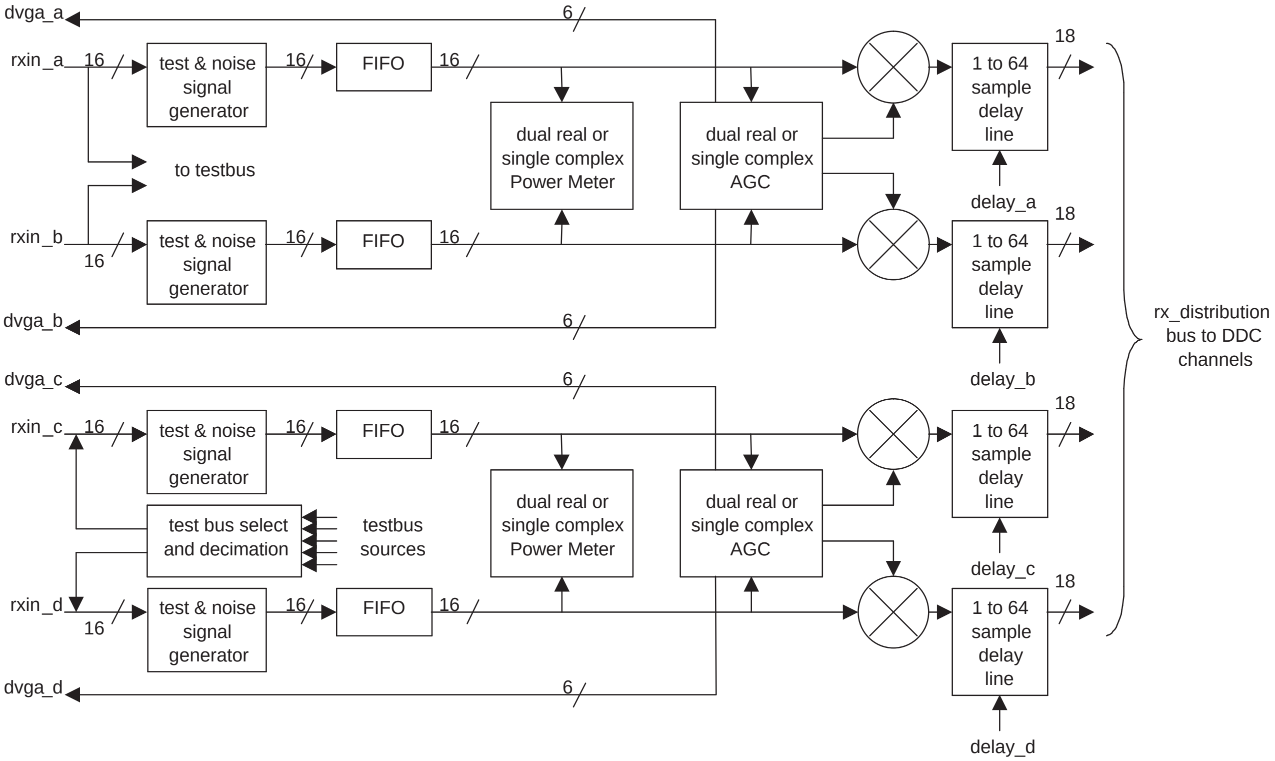

dvga_a
rxin_a
16
test & noise signal generator
16
FIFO
16
6
dual real or single complex Power Meter
dual real or single complex AGC
1 to 64 sample delay line
18
to testbus
delay_a
rxin_b
16
dvga_b
delay_b
rx_distribution bus to DDC channels
dvga_c
rxin_c
test bus select and decimation
testbus sources
delay_c
rxin_d
dvga_d
delay_d

Figure 2-1. Receive Data Input Interface

The GC5018's receive input data interface accepts data from two sources:

- Signal data presented at the four 16-bit digital data input ports.
- A LFSR test signal generator allows the GC5018 to be tested using a known repetitive data sequence.

Signal data can be provided in binary or 2's complement form. The location of the ADC's MSB can be programmed to allow for additional AGC headroom if desired. For example, a 14-bit ADC may be connected with the MSBs aligned, or shifted down to allow the AGC additional gain range before clipping the signal.

Signal data can be accepted at rates up to rxclk in UMTS mode for either 8 normal channels or 4 double length final pulse shaping filter channels. In CDMA mode the maximum input rate is rxclk for real inputs, or rxclk/2 for complex inputs. For maximum filter performance, higher clock rates generally allow longer filters.

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 using complex input mode. The mapping of I and Q data onto the four input ports is programmable.

Signal input data is clocked into 8-stage FIFOs using a matching external clock signal adcclk_a/b/c/d. Signal data is clocked out of the FIFO from a gated rxclk (the GC5018 receive section clock). The FIFO allows arbitrary phase relationship between adcclk_a/b/c/d and rxclk. The frequency relationship is mandated by the programmed configuration.

The test and noise generator can supply test sequences or add noise to the input signal data. The test sequences, when combined with the checksum generators, are useful for initial board debug or power-on self-test.

For applications that require receiver desensitization, the noise generator can add noise to input data streams.

Many internal chip signals can be routed to the testbus for evaluation and debug purposes. When the testbus is enabled, the rxin_c and rxin_d ports are driven as digital outputs.

Each of the four outputs to the DDC channels includes a 1 to 64 sample delay line.

PROGRAMMING	
VARIABLE	DESCRIPTION
ssel_ddc(2:0)	Selects the sync source for the DDC data input mux and mixer. This sets the sync source for DDC input clock generation and synchronization for all DDC channels.
offset_bin_X	Selects offset binary input when set, 2's complement input when cleared. X={a,b,c,d}
msb_pos_X(2:0)	Identifies the connection location of the ADC's MSB. Programmed values of {0..7} corresponds to msb at {rxin_x_15.. rxin_x_8}. X={a,b,c,d}

2.1.1 Receive FIFO

The receive FIFO consists of an 8 stage memory and 2 counters generating the input write pointer and output read pointer. When the FIFO receives a sync signal, the input and output pointers are initialized with a write to read pointer offset of four samples. Input samples from rxin_X (writes) are clocked with the adcclk_X input clock rising edges, and the input pointer advances on each clock rising edge. Output samples (reads) and the output pointer are clocked with the rxclk input signal rising edges, divided by the programmed sample rate loaded into the rate_sel(1:0) control register.

PROGRAMMING	
VARIABLE	DESCRIPTION
adc_fifo_bypass	When set, bypasses the input FIFOs and input data is latched directly using the rxclk. When cleared, input data is latched using the adcclk_a/b/c/d inputs.
ssel_adc_fifo(2:0)	Selects the sync source for the FIFO state machines. This sync signal initializes the FIFO input and output pointers.
rate_sel(1:0)	This selects the FIFO input and output rate; {rxclk, rxclk/2, rxclk/4 or rxclk/8 }. For example, with rxclk at 153.6MHz, set rate_sel to 0, 1, 2 or 3 respectively for adcclk_a/b/c/d 153.6, 76.8, 38.4 or 19.2MHz.
adc_fifo_strap_ab	When set, the rxin_a and rxin_b FIFO input and output pointers are synchronized to support complex input signals.
adc_fifo_strap_cd	When set, the rxin_c and rxin_d FIFO input and output pointers are synchronized to support complex input signals.

2.1.2 Receive Input Power Meters

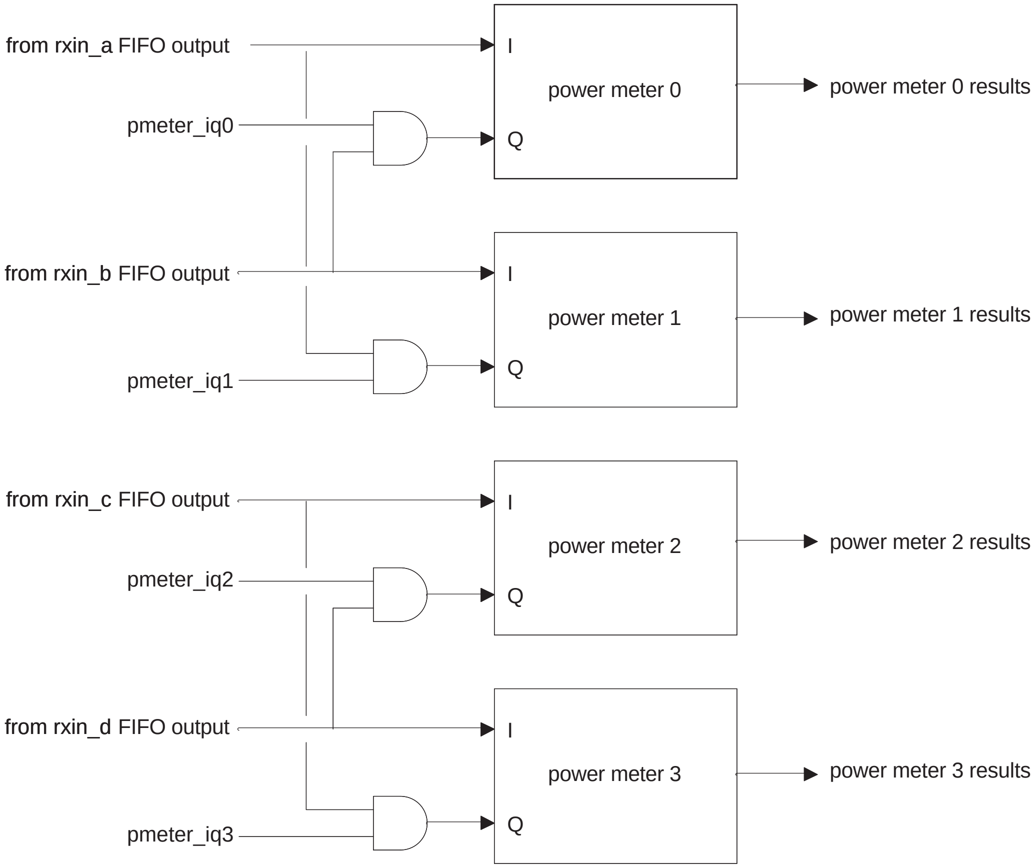


Figure 2-2. Receive Input Power Meters

Four Receive Input RMS power meters are provided. For real inputs, the four power meters can be used to measure the RMS power of the combined carriers in each of the four input signals (the Q input is held at zero). For complex inputs, two power meters can be use to measure the combined complex power and two can be disabled.

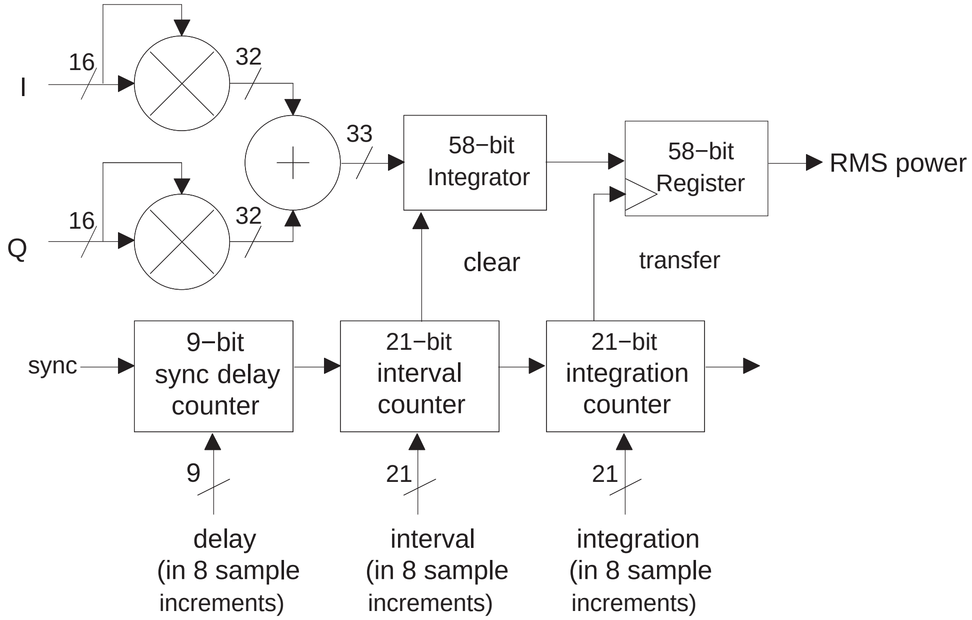

I
16
32
Q
16
32
33
58−bit Integrator
58−bit Register
RMS power
clear
transfer
sync
9−bit sync delay counter
21−bit interval counter
21−bit integration counter
9
21
21
delay (in 8 sample increments)
interval (in 8 sample increments)
integration (in 8 sample increments)

Figure 2-3. Detailed Functionality of Receive Input Power Meter

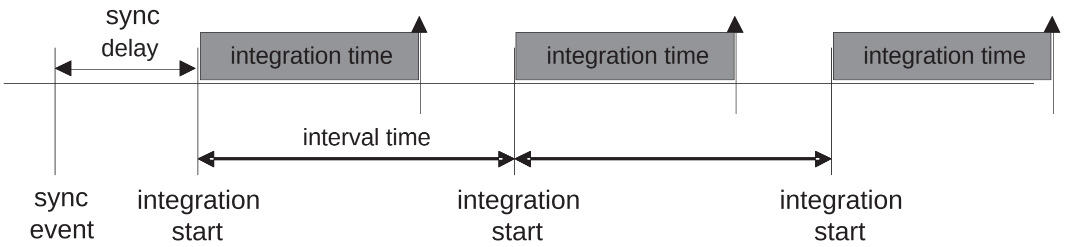


Figure 2-4. Receive Input Power Meter Timing

Power is calculated by squaring each 16 bit I (I and Q for complex inputs) sample, summing, and then integrating the summed-squared results into a 58 bit accumulator over a programmable integration period. The integration period is programmed into the 21 bit counter, in 8 sample increments. The power read is:

power = [(I^2) x (Nx8 + 1)] for real inputs where N is the integration count.

power = [(I^2 + Q^2)x (Nx8 + 1)] for complex inputs where N is the integration count.

A programmable 21 bit interval counter sets the power measurement interval (how often power will be measured) in 8 sample increments. A measurement integration period is started at the beginning of each interval period.

The process begins with a sync event starting the 9 bit delay counter. After (8xsync_delay + 2) samples, the integration interval is started. Integration continues until the integration count is met, at which point the 58 bit integrator results are transferred to the read only register and an interrupt is generated. A new measurement period will start 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.

The 21-bit counters in 8 sample increments allow up to 104.8mS interval times at 160MHz clock.

PROGRAMMING	
VARIABLE	**DESCRIPTION**
recv_pmeterX (57:0)	58 bit power measurement result. X= {0,1,2,3}.

PROGRAMMING	
VARIABLE	DESCRIPTION
recv_pmeterX_sqr_sum(20:0)	21 bit integration (square and sum) period. X= {0,1,2,3}.
recv_pmeterX_sync_delay(8:0)	Power meter delay sync period. X= {0,1,2,3}.
recv_pmeterX_strt_intrvl(20:0)	21 bit measurement interval. X= {0,1,2,3}. *The strt_intrvl value must be greater than the sqr_sum value.*
ssel_recv_pmeter_X(2:0)	Sync source. X= {0,1,2,3}.
pmeterX_iq	Selects complex power measurement input mode when set. X= {0,1,2,3}.
recv_pmeterX_ena	Enables power meter when set. X= {0,1,2,3}.

2.1.3 Receive Input AGC (RAGC)

Input signals from the ADCs can be used to create a front end composite AGC loop when combined with a digitally controlled variable gain amplifier (DVGA) connected before the ADCs. The AGC system operates by integrating the square of the ADC samples over a programmable interval and applying a table driven error signal to a loop integrator based on the squared integration output. The error table maps the signal power to a user programmed error value. The loop integrator output is used to drive map tables to control the DVGA output pins and a gain adjustment multiplier. Fast updates can be enabled if desired, to cause the loop integrator to quickly adjust to interfering signals. The ADC input signals can also be passed through a high pass filter to remove DC offset before squaring the input.

The programmable error table, integrator mapping tables, and clip thresholds, when combined with the user programmable interval timers provide a highly flexible AGC function.

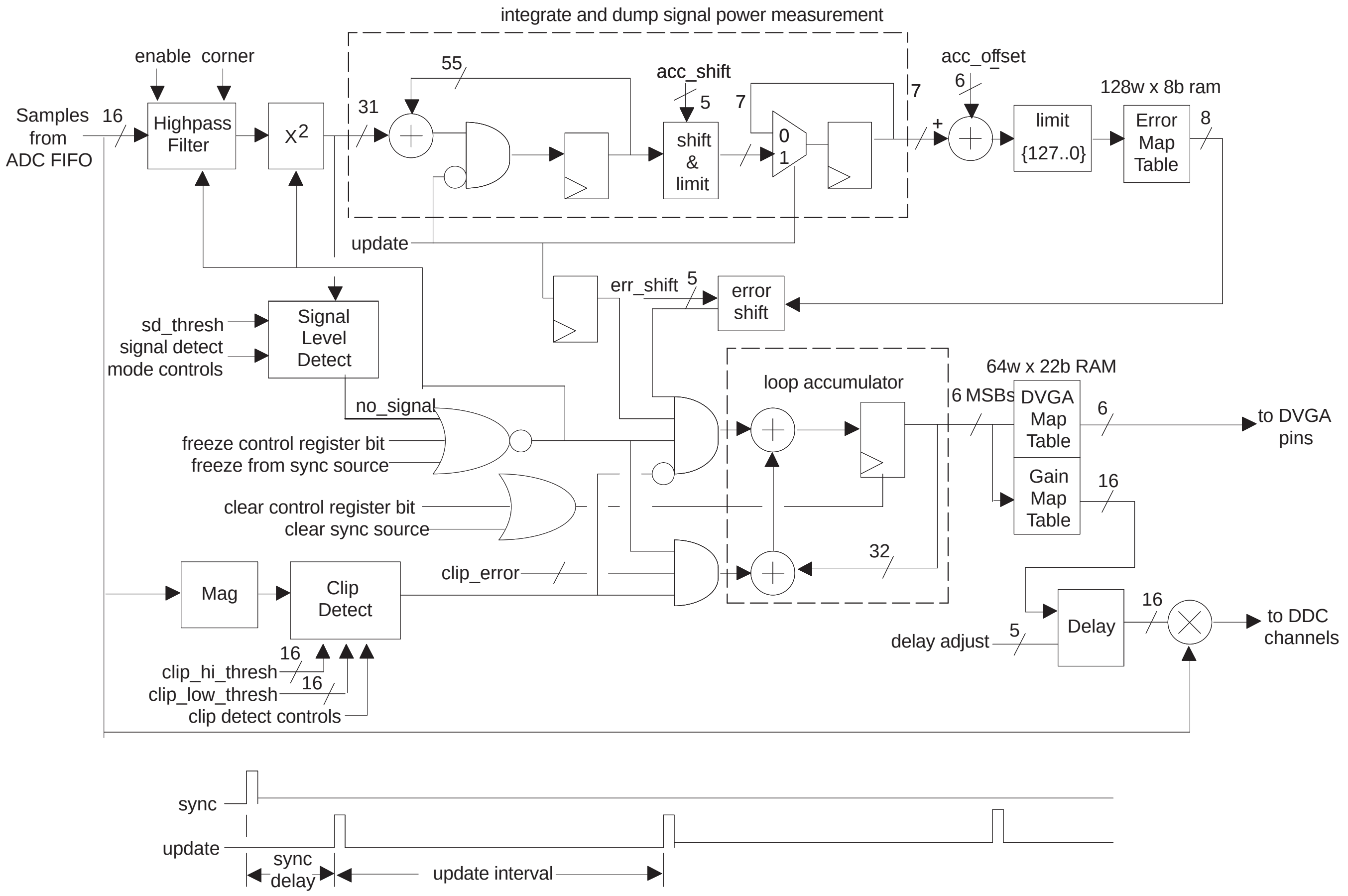


Figure 2-5. Receive Input AGC

The AGC measurement interval timer is a 24-bit timer initialized by a sync after a programmable 8-bit delay. During the integration interval, the squared input signal is shifted by the programmed value and accumulated. At the end of the interval time, an update pulse is generated, and the selected 7 bits of the 55-bit accumulated power is upper limit checked and transferred to the power holding register. A programmable offset is applied, and the following limit check produces a 7 bit address value for the error map table RAM. The user programmable error map table and following gain shift setting are used to determine the loop error signal to be added to the 32-bit AGC loop accumulator. The error value is only added to the loop accumulator once per update. The loop accumulator upper 6 MSBs are used as the address for the programmable DVGA map table and gain map table. The gain map table address can be delayed from 0 to 31 clock cycles to align DVGA changes to signal level changes at the output of the AGC.

The AGC includes four sources for freezing the loop and holding the loop accumulator constant. A general sync source can be used to directly control the freeze; when the selected sync source is high, the AGC will be held, and when low, the AGC will operate. A control register bit freezes the AGC in the same fashion; when the bit is set, the AGC is held, and when cleared, the AGC will operate. A signal level detector is provided that can be used to automatically freeze the AGC loop in the event of input signal loss. A programmable signal detection threshold value, number of samples below the signal detection threshold, and window timer are used to determine when no signal is present. Finally, a programmable number of AGC updates after sync can be programmed, and the AGC will he held until the next sync event. Freeze holds the loop accumulator constant, the integrate and dump accumulator constant and the interval timer constant. When freeze is released, the interval timer will resume counting.

A sync event will always reinitialize the integrate and dump interval timer, and terminate the pending update to the loop accumulator from the current integrate and dump measurement interval. For example, if a sync event occurs during an integrate and dump interval, that interval will be terminated without updating the loop, and the integrate and dump accumulator will be cleared. After the programmed sync delay, a new interval will start.

The AGC includes a dual threshold clip detect function, using two programmable 16-bit thresholds and programmable counters. The clip detector will cause immediate loop accumulator updates while the clip event is active. The 16-bit clip error value is aligned at the MSBs of the loop accumulator. Clip events are qualified when a programmed number of samples are above the high clip threshold during the programmable clip window time. For example, a clip event can be defined as 8 samples above the clip high threshold in a 256 sample window; the clip high threshold, the number of samples above the high clip threshold and the sample window time are programmable. Once the clip event has occurred, the clip duration is controlled by the clip low threshold value, clip low samples value and clip low timer. The clip event is cleared when the number of samples below the low clip threshold exceeds the programmed value within the clip low timer window. The clip low threshold, number of clip low samples and the clip low window timer are programmable.

The AGC blocks can be paired together, rxin_a with rxin_b, and rxin_c with rxin_d, to produce a complex input AGC mode. The clip detector output from the rxin_b/d AGCs is logically OR'ed with the rxin_a/c clip detect outputs. The squared input function before the integrate and dump and signal level detector is replaced with a $I^2 + Q^2$ power calculation. The accumulator MSBs from the rxin_a/c AGCs are connected to the rxin_c/d DVGA map table and gain map table inputs. This arrangement allows the AGCs to operate in a direct conversion receiver system by controlling the $I^2 + Q^2$ complex signal level.

The highpass filter is a 32 bit accumulator followed by an adjustable shift to control the corner frequency, a subtractor to remove the accumulated offset and a final limiter to produce a 16 bit result. The highpass filter function is enabled by setting hp_ena; clearing hp_ena holds the accumulator reset.

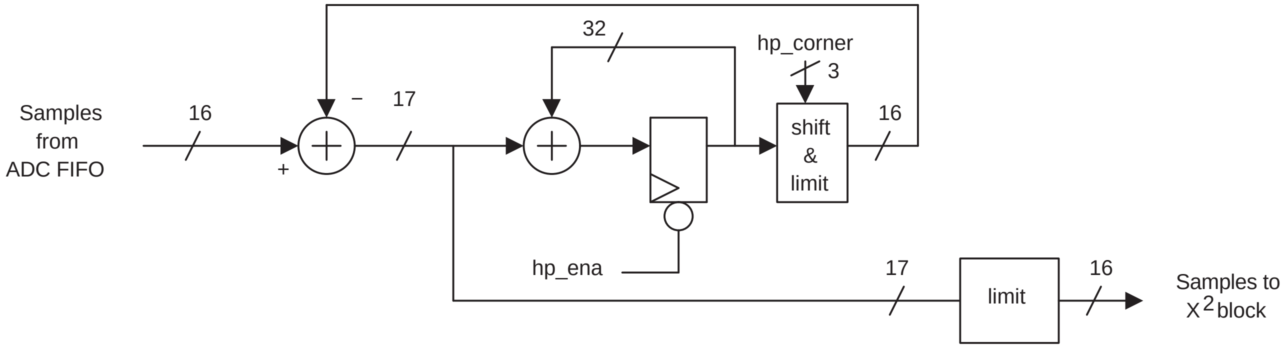


Figure 2-6. High Pass Filter in Receive Input AGC

PROGRAMMING	
VARIABLE	**DESCRIPTION**
ragc_bypass_X	Bypasses the entire receive AGC circuit when set. X = {0,1,2,3}
hp_ena_X	Enables high pass filter when set
hp_corner_X(2:0)	Adjusts the corner frequency of the high pass filter
integ_interval_X(23:0)	Integrate and dump signal power measurement interval in samples.
acc_shift_X(4:0)	Shift down amount following the integrate and dump accumulator.
acc_offset_X(5:0)	Offset value applied to the shifted integrate and dump output.
ragc_sync_delay_X(7:0)	AGC sync delay interval, from 1 to 256 samples.
ssel_ragc_interval_X(2:0)	Sync source selection for the interval timer.
ssel_ragc_freeze_X(2:0)	Sync source selection for AGC freeze
ssel_ragc_clear_X(2:0)	Sync source selection for the AGC loop accumulator clear
ragc_freeze_X	Register bit to freeze the AGC when set
ragc_clear_X	Register bit to clear the AGC accumulator when set
ragc_update_X(7:0)	Sets the number of updates per sync event, after which no further updates will occur until the next sync event. Program to 0x00 to continually update.
sd_ena_X	Enables freezing the AGC with the signal detector when set
sd_thresh_X(15:0)	Signal detection threshold for AGC channel X. This 16 bit word is lined up with bits 23 down to 8 of the square output. The smallest signal level is that can be programmed is therefore 16 LSBs on the ADC input, and the largest is 4095 LSBs at the ADC input.
sd_samples_X(15:0)	The number of samples below the signal detect threshold within the signal detect sample timer window required to freeze on the AGC.
sd _timer_X(15:0)	Window timer to qualify signal detection.
clip_hi_thresh_X(15:0)	Clip detector high threshold
clip_lo_thresh_X(15:0)	Clip detector low threshold
clip_hi_samples_X(7:0)	A clip event is detected when this number of samples above the clip high threshold within the clip high sample timer window exceeds this value.
clip_lo_samples_X(7:0)	A clip event ends when this number of samples below the clip low threshold within the clip low sample timer window exceeds this value.
clip_hi_timer_X(15:0)	Window timer to qualify clip events.
clip_lo_timer_X(15:0)	Window timer to determine when the clip event ends.
clip_error_X(15:0)	Error signal applied to the AGC accumulator when a clip event is active. This data is MSB aligned, and therefore can cause immediate changes to the accumulator.
ragc_error_map_X	128w x 8b memory holding the log to error look up table.
dvga_map_X	64w x 6b memory holding the accumulator to DVGA look up table
gain_map_X	64w x 16b memory holding the accumulator to GAIN look up table (256 decibels is unity gain).
delay_adj_X(4:0)	Delay between DVGA output updates and gain map updates to compensate for ADC pipeline delays, etc.
err_shift_X(4:0)	Error map table output shift up before adding to loop accumulator
complex_01	Enables complex AGC mode on inputs rxin_a and rxin_b when set
complex_23	Enables complex AGC mode on inputs rxin_c and rxin_d when set

PROGRAMMING	
VARIABLE	DESCRIPTION
ragc_accum_X(31:0)	32-bit read only register holding the current contents of the loop accumulator.
tristate(10:7)	3-state controls for the dvga_d/c/b/a output pins; pins are in tristate when the 3-state bits are set.
ragc_mpu_ram_read	When set, the receive AGC map rams are readable via the MPU control interface. *The GC5018 signal path is not operational when this bit is set, it is intended for debug purposes only.*

2.1.4 Test and Noise Signal Generator

The test and noise generator can generate test signals to replace the rxin_a/b/c/d inputs as a tool for debug, evaluation and self test. Checksum generators included in the individual DDC channels at the outputs can be used in conjunction with the noise generator and the internal sync timer block to create the built in self test function.

The test and noise signal source included in this block is a 23-bit linear feedback shift register (LFSR) with a fixed polynomial and fixed initialization state. A sync input is required to initialize the LFSR, and the sync source is connected to the ddc_counter output signal.

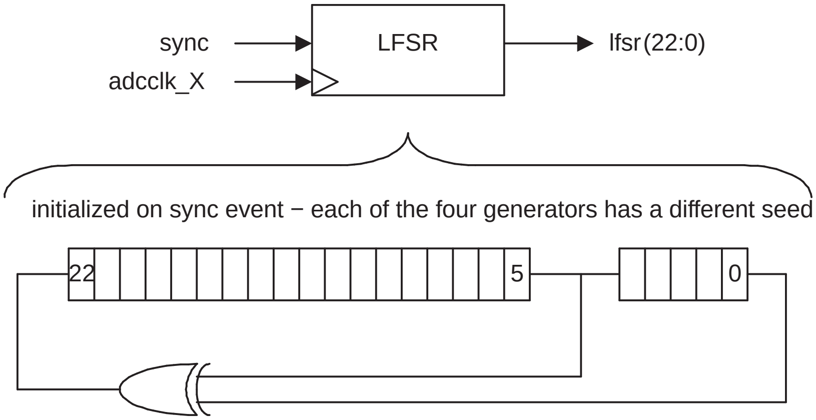

sync
adcclk_X
LFSR
lfsr(22:0)
initialized on sync event − each of the four generators has a different seed
22
5
0

Figure 2-7. Noise Signal Generator

Receive Input Port	LFSR Seed Value, MSB to LSB
rxin_a	100 0000 0000 0000 0001 0000 (0x400010)
rxin_b	010 0110 1110 0110 1100 1110 (0x26E6CE)
rxin_c	110 1110 1010 0010 1001 1000 (0x6EA298)
rxin_d	000 1011 0001 1110 1011 0111 (0x0B1EB7)

The 23-bit LFSR output signal if used to create a 16-bit “dout(15:0)” test signal using XOR combinations of the LFSR bits.

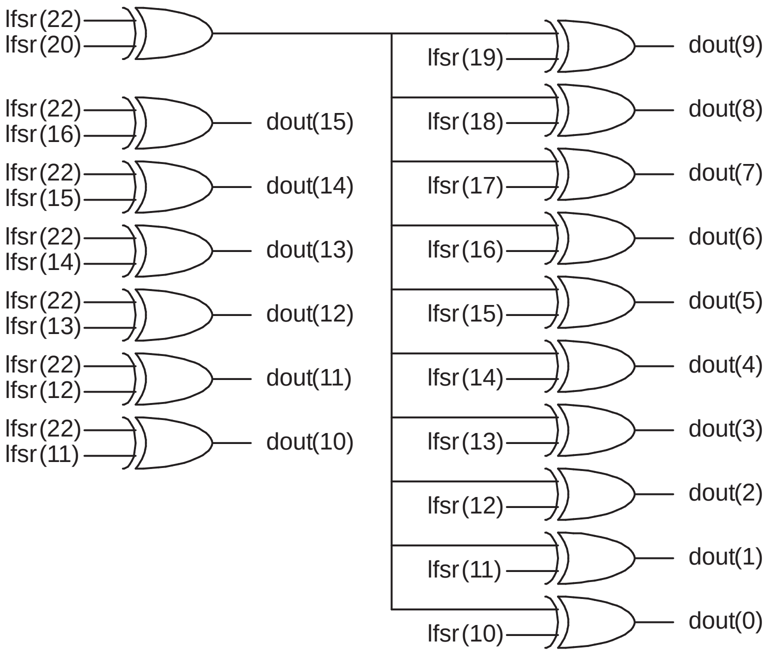


Figure 2-8. Mapping of LFSR Values to Output Bits

To enable the test signal generator, the slf_tst_ena control bit is set. The rxin_a/b/c/d signals will be then replaced by the four generator output streams. To use this test signal generator as a signal source for self test, the user must also set the adc_fifo_bypass control bit. Setting the adc_fifo_bypass control bit causes the adcclk_a/b/c/d input clocks to be internally replaced with rxclk/N, where N is as programmed with the rate_sel(1:0) control bits to {1,2,4 or 8}.

The test signal generators can also output a programmable constant value. All four test signal generators output the same programmable constant value.

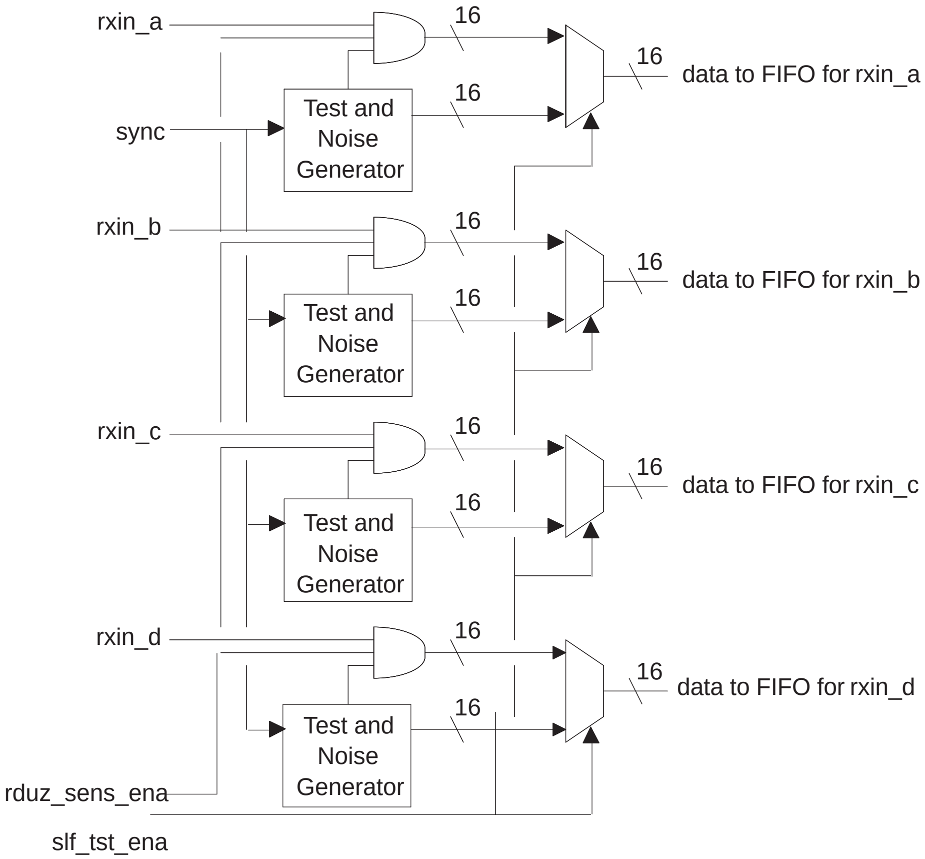


Figure 2-9. Block diagram of Noise Generator Input Options

The LFSR circuits can also be used to add noise to the rxin_a/b/c/d input signals by setting the rduz_sens_ena control register bit. The magnitude of the noise added can be adjusted by programming the nz_pwr_mask(15:0) control register. In the figure below, X = {a,b,c or d}.

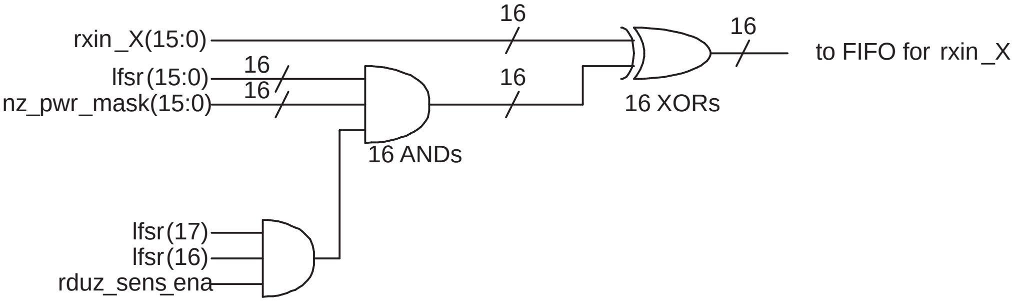


Figure 2-10. Detail Circuit for Adding Noise Generator Signal to rxin Signal

PROGRAMMING	
VARIABLE	DESCRIPTION
slf_tst_ena	When set, the test signal generators replace the rxin_a/b/c/d input signals with internally generated psuedo random sequences. The fifo_bypass bit must be set when this bit is set.
rduz_sens_ena	Enables the LFSR, adding noise to the ADC input data when set.
nz_pwr_mask(15:0)	Selects the power of the noise added to the ADC input data.
adc_fifo_bypass	When set, the FIFO is essentially bypassed, and the adcclk_a/b/c/d clock input ports are ignored.
ddc_counter(31:0)	32 bit general purpose counter interval
ddc_counter_width(7:0)	8 bit general purpose counter timeout width pulse
ssel_ddc_counter(2:0)	Sync source selection for the general purpose counter
self_test_constant(17:0)	18-bit self test constant value applied to all 4 rxin_a/b/c/d inputs when self_test_const_ena is set.
self_test_const_ena	Enables the self test constant value for rxin_a/b/c/d

2.1.5 Sample Delay Lines

The four sample delay line blocks each consist of a 64 register memory and a state machine. The state machine uses a counter to control the write (input) pointer, and the programmed read offset register data to create the read (output) pointer. Programming larger read offset register values increases the effective delay at a resolution equal to the sample rate.

The read offset registers, delay_line_X, are double buffered. Writes to these registers may occur anytime, but the actual values used by the circuit will not be updated until a delay line sync event occurs.

PROGRAMMING	
VARIABLE	DESCRIPTION
delay_line_X(5:0)	Read offset into the 64 element memory for each delay line. X= {0,1,2,3}.
ssel_delay_line_X(2:0)	Selects the sync source used to update the double buffered delay line register.

2.1.6 Test Bus

When the test bus is enabled, the rxin_c(15:0) and rxin_d(15:0) ports become outputs, and the dvga_c and dvga_d pins are combined with these pins to allow 36 bit wide signals from the DDC channels and the receive input interface to be multiplexed to this test output port. Many of these sources can be decimated to reduce the output sample rates.

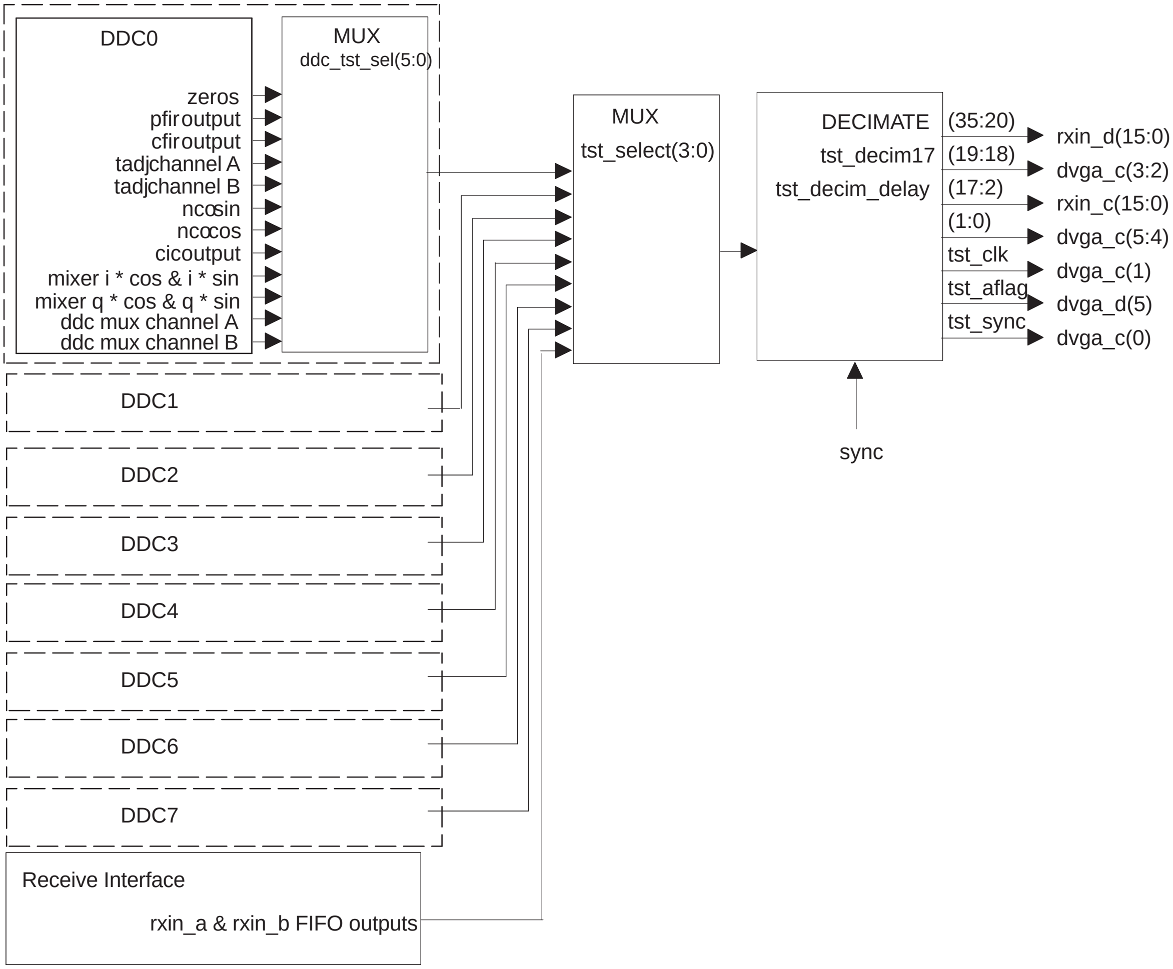


Figure 2-11. Test Bus Output Circuit Showing Options for Selecting Signal

PROGRAMMING	
VARIABLE	**DESCRIPTION**
ssel_tst_decim(2:0)	Selects the sync source for the testbus decimator
tst_decim_delay(3:0)	Sets the testbus decimator delay from sync
tst_decim17	When set the decimation factor of the test bus output block is 17X. When cleared, the decimation factor is 1X (no decimation).
tst_on	Enables the test bus; rxin_c(15:0) and rxin_d(15:0) are changed from inputs to outputs, dvga_c(5:0) and dvga_d(5) are used as part of the test bus.
tst_select(3:0)	Selects the source block for the testbus output; DDC0-7 or Receive Interface.
ddc_tst_sel(5:0)	Selects the signal to be output from the DDC block
tst_rate_sel(4:0)	Sets the testbus output clock tst_clk period to (tst_rate_sel + 1) rxclk cycles.

2.2 DDC Organization

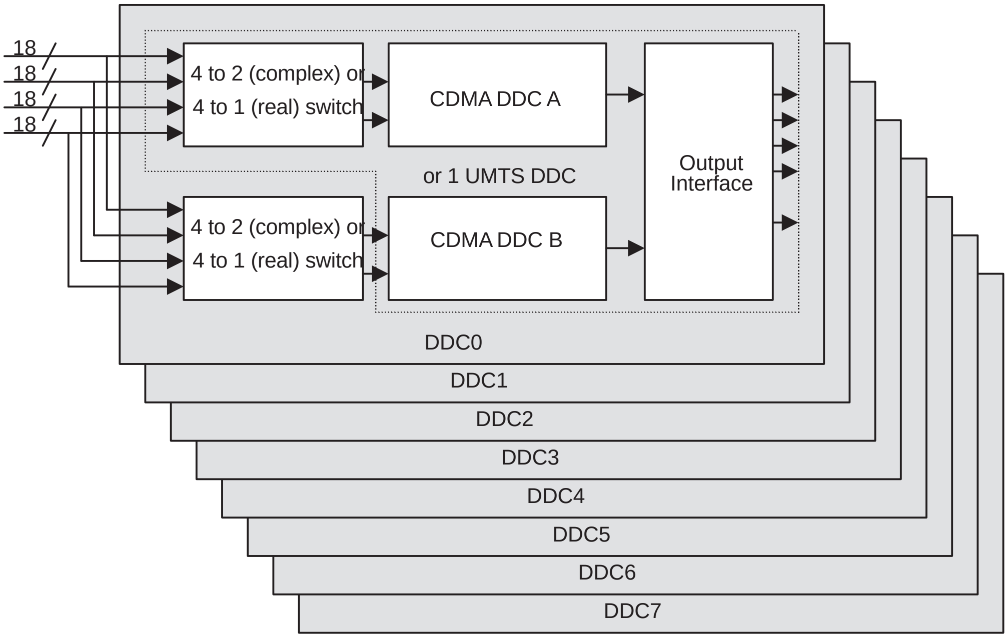


Figure 2-12. DDC Organization for Single Length Filter Mode

The GC5018 provides downconversion for up to 8 UMTS receive channels, 16 CDMA2000 receive channels or 16 TD-SCDMA receive channels. Downconversion channels are organized into 8 DDC blocks. Each individual DDC block provides 2 CDMA2000 or 2 TD-SCDMA DDC channels, A and B, or 1 UMTS channel.

Both CDMA DDC channels in a DDC 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.

Two adjacent DDC blocks (for example, DDC0 and DDC1) can be strapped together to form a single UMTS DDC channel with double-length final pulse shaping filtering. The GC5018 can therefore provide 4 UMTS DDC channels with double-length final PFIR filtering as shown in the following diagram.

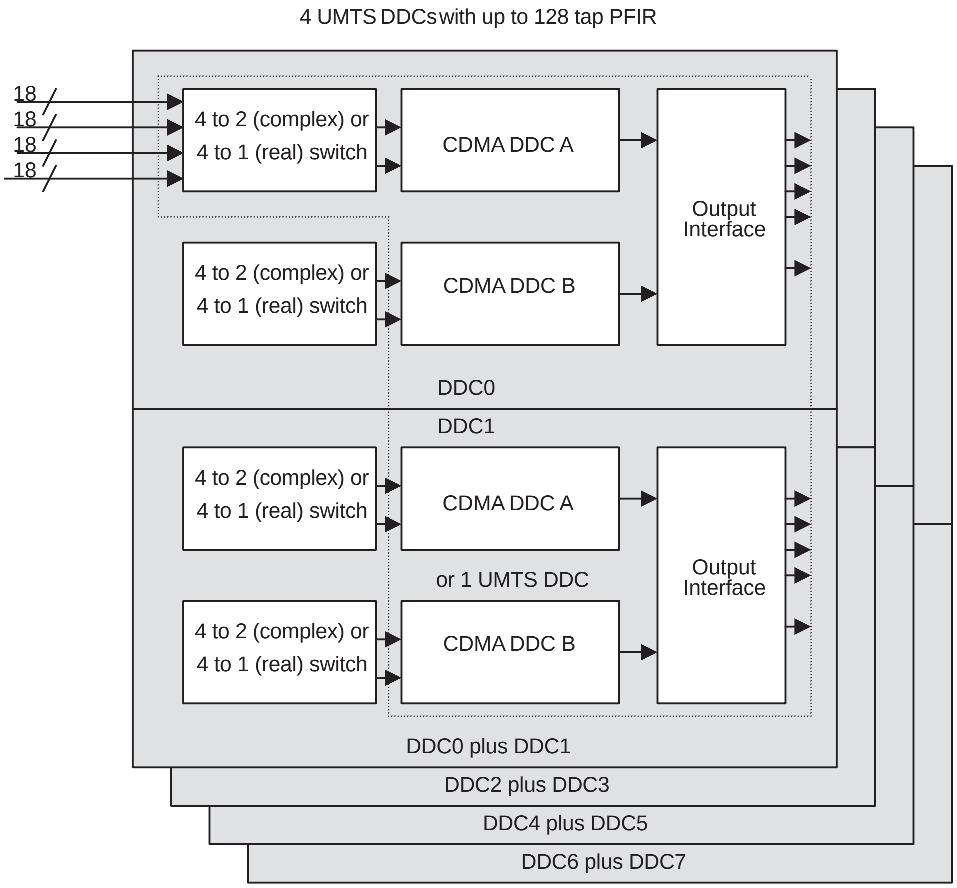


Figure 2-13. DDC Organization for Double Length Filter Mode

PROGRAMMING	
VARIABLE	**DESCRIPTION**
ddc_ena	When set, turns on the DDC.
cdma_mode	When set, puts the DDC block in dual channel CDMA mode.
gbl_ddc_write	When set, all subsequent programming (writes only) for DDC0 and DDC1 is also written to DDC2/4/6 and DDC3/5/7.

2.2.1 Downconverter Function Blocks

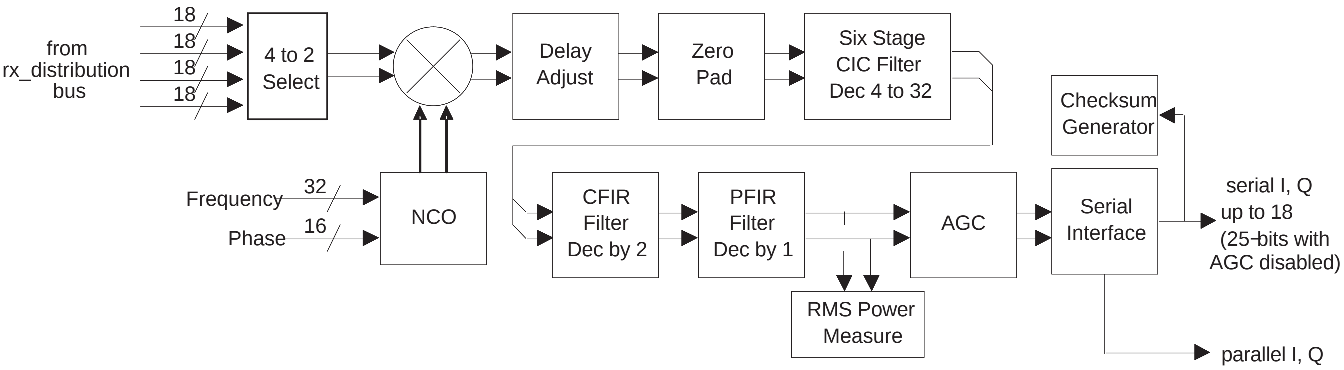


Figure 2-14. DDC Functional Block Diagram

Each GC5018 downconversion block can process two CDMA carriers or a single UMTS carrier. Signal data is selected from one of four ports for real inputs, or two of four ports for complex inputs. Data from the selected port(s) 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 output interface block can be programmed to decimate by 2 if desired.

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.

2.2.2 DDC Mixer

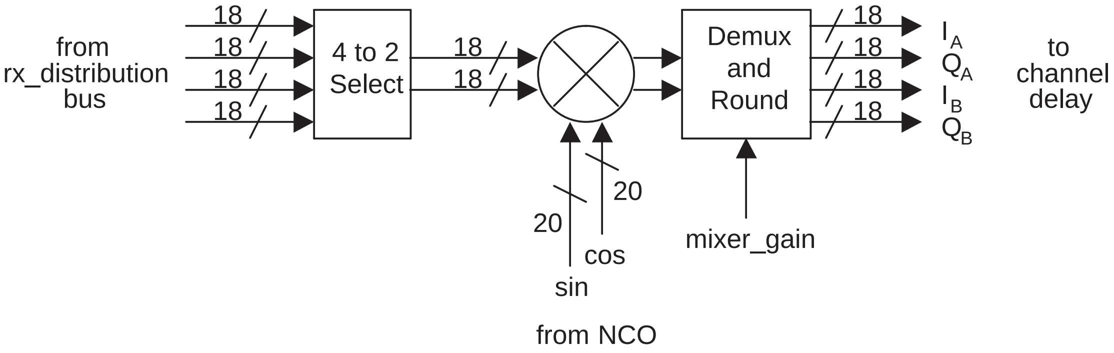


Figure 2-15. Mixer Functional Block Diagram

The receive mixer translates the input (from one of the input signal sources) 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 Hertz.

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 be set independently; the path selection and the mixer tuning and phase. 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 input rate in CDMA mode with real inputs is rxclk (remix_only is set, see below).

The maximum input rate in CDMA mode with complex inputs is rxclk/2 due to sharing of multiplier resources.

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	For CDMA mode only, set this bit for real input data at the rxclk rate. For complex inputs in CDMA mode, the maximum input data rate is rxclk/2, and this bit must be cleared. For CDMA mode with real inputs at the rxclk/2 rate or lower, this bit must be cleared
zero_qsample	When set, the Q samples used by the mixer are always zero. This bit should be set for real only inputs in UMTS mode, or real only inputs in CDMA mode when the input sample rate is rxclk/2 or lower.
ch_rate_sel(1:0)	Specifies the input channel data rate (rxclk, rxclk/2, rxclk/4, or rxclk/8 MSPS).
mixer_gain	When asserted, adds 6dB of gain in the mixer. This gain is highly recommended.

2.2.3 DDC Number Controlled Oscillator (NCO)

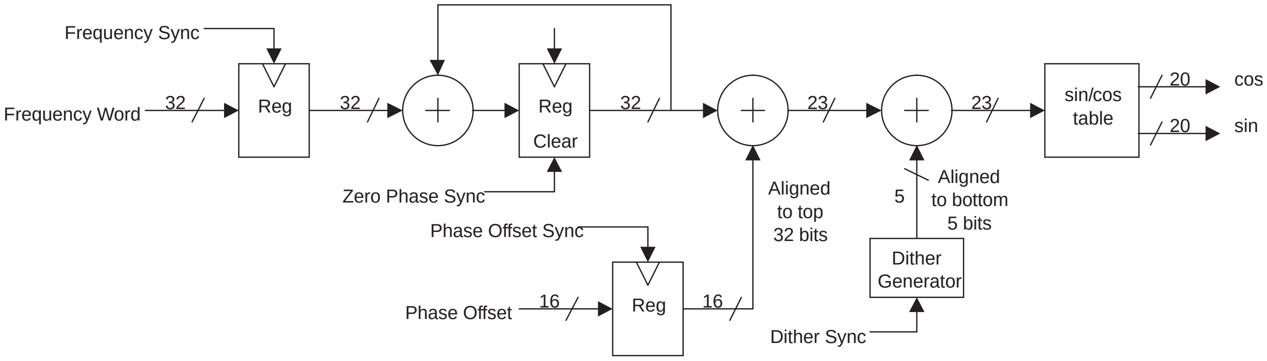


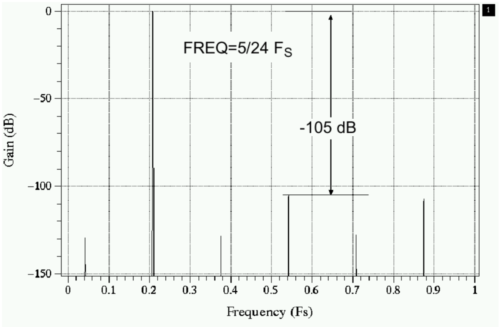
Figure 2-16. Detailed NCO Circuit

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 will change 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 bits and is useful for reducing NCO spurious outputs. This dither generation is enabled by setting the dither_ena bit; the magnitude of the dither can be reduced by setting one or both of the dither_mask bits

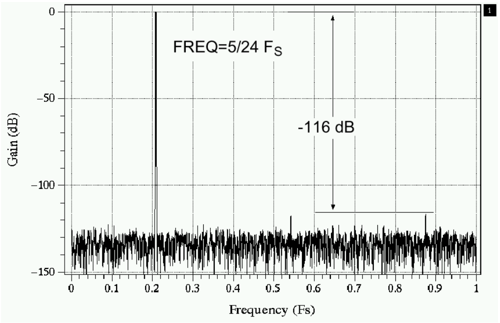
DITHER PROGRAMMING	
VARIABLE	**DESCRIPTION**
dither_ena	When set turns dither on. Clearing turns dither off.
dither_mask(1:0)	Masks the MSB and MSB-1 dither bits, respectively, when set.

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 (N x Fs/24, where N = {1,2, . . . 23}) dither is ineffective – in these cases an initial phase of 4 reduces NCO spurs. The figures below show the spur level performance of the NCO without dither, with dither, and with a phase offset value.



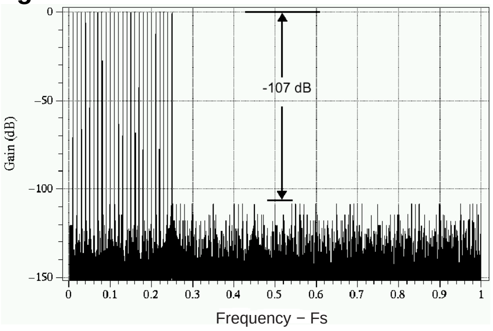
a) Worst Case Spectrum Without Dither

Figure 2-17. NCO SFDR - Without Dither



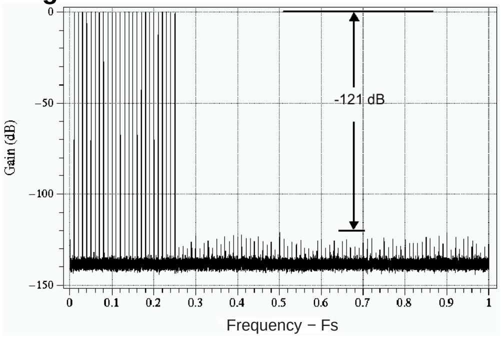
b) Spectrum With Dither (Tuned to Same Frequency

Figure 2-18. NCO SFDR - With Dither



a) Plot Without Dither or Phase Initialization

Figure 2-19. NCO Spectra (0 and Fs/4) - Without Dither or Phase Initialization



b) Plot With Dither or Phase Initialization

Figure 2-20. NCO Spectra (0 and Fs/4) - With Dither or Phase Initialization

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 frequency resolution is Fclk/ 2^{32}. As an example, at an input clock rate of 61.44 MHz, the frequency step size would be approximately 14 milli-Hertz. The Frequency Word is determined by the formula:

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

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

FREQUENCY 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 16 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 m°. The Phase Offset Word is determined by the formula:

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

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

PHASE 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 16 CDMA DDC channels 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 and Phase Offset Sync determine when frequency and phase offset changes occur. For example, generating a Frequency Sync after programming the two frequency words will cause 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. Note that there is one set of sync's provided for each DDC. When one DDC is used to process two CDMA signals, the syncs are shared between them.

SYNC 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

2.2.4 DDC Filtering and Decimation

The purpose of the receive filter chain is to isolate the signal of interest (and reject all others) that has been previously translated to baseband via the mixer and NCO. The overall decimation through the chain 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 and for CDMA the output rate should be 2.4576 MSPS. TD-SCDMA systems require the output rate be the chip rate of 1.28 MSPS. The output interface is programmed to decimate by 2 for the TD-SCDMA case.

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)
- Pulse-shape filtering via the programmable FIR filter (PFIR) with no decimation
- Output interface, serial or parallel format, with no decimation or decimate by 2

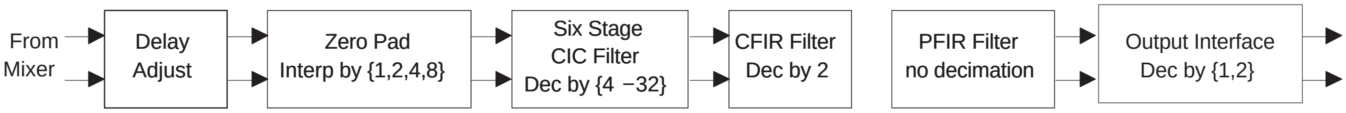

From Mixer
Delay Adjust
Zero Pad Interp by {1,2,4,8}
Six Stage CIC Filter Dec by {4 −32}
CFIR Filter Dec by 2
PFIR Filter no decimation
Output Interface Dec by {1,2}

Figure 2-21. DDC Filtering Functional Block Diagram

The table below contains some examples of decimation and sample rates at the output of each block for UMTS, CDMA and TD-SCDMA standards at various supported input samples. For each example, the differential ADC clocks are provided to the GC5018 at the input sample rate and the rxclk is provided at the zero pad output rate.

Table 2-1. Examples of Decimation and Sample Rates[1]

	Input Sample Rate (MSPS)	Zeros Added	rxclk(MHz) and Zero Pad Output Rate (MSPS)	CIC Decimation	CIC Output Rate (MSPS)	CFIR Decimation	CFIR Output Rate (MSPS)	PFIR Decimation	PFIR Output Rate (MSPS)	Output Decimation
UMTS	122.88	0	122.88	8	15.36	2	7.68	1	7.68	1
UMTS	92.16	0	92.16	6	15.36	2	7.68	1	7.68	1
UMTS	76.80	1	153.6	10	15.36	2	7.68	1	7.68	1
UMTS	61.44	1	122.88	8	15.36	2	7.68	1	7.68	1
CDMA	122.88	0	122.88	25	4.9152	2	2.4576	1	2.4576	1
CDMA	78.6432	0	78.6432	16	4.9152	2	2.4576	1	2.4576	1
CDMA	78.6432	1	157.2864	32	4.9152	2	2.4576	1	2.4576	1
CDMA	61.44	1	122.88	25	4.9152	2	2.4576	1	2.4576	1
TD-SCDMA	92.16	0	92.16	18	5.12	2	2.56	1	2.56	2
TD-SCDMA	81.92	0	81.92	16	5.12	2	2.56	1	2.56	2
TD-SCDMA	76.80	0	76.80	15	5.12	2	2.56	1	2.56	2
TD-SCDMA	76.80	1	153.6	30	5.12	2	2.56	1	2.56	2
TD-SCDMA	61.44	1	122.88	24	5.12	2	2.56	1	2.56	2

[1] The DDC output interfaces, both serial and parallel formats, can be programmed to decimate by 2. For the TD-SCDMA examples listed above, the DDC output rate is 1.28Msps (1x chip rate).

2.2.5 DDC Channel Delay Adjust and Zero Insertion

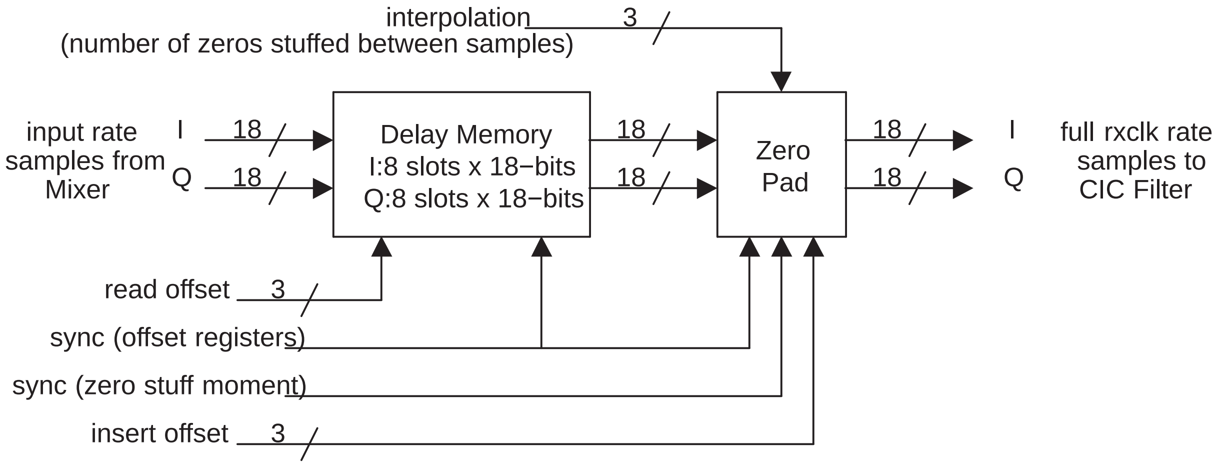


Figure 2-22. DDC Delay and Zero Insertion Block

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 GC5018 digital downconversion such as cables, splitters, analog downconverters, filters, etc.

The Delay Memory block consists of an 8 register memory and a state machine. The state machine uses a counter to control the write (input) pointer, and the programmed read offset register data to create a read (output) pointer. Programming larger read offset register values increases the effective delay at a resolution equal to the input sample rate.

The Zero Pad block is used in conjunction with the Delay Memory for delay adjustments. For example, with input rates of rxclk/8, the Zero Pad block interpolates the input data to rxclk by inserting 7 zeros. The Zero Pad's sync insert offset 3-bit control specifies when the zeros are inserted relative to the Sync signal. This permits a fine adjustment at the rxclk resolution.

The read offset register, tadf_offset_course_a/b, and the insert offset register, tadj_offset_fine_a/b, are double buffered. Writes to these registers may occur anytime, but the actual values used by the circuit will not be updated until a register sync

PROGRAMMING	
VARIABLE	**DESCRIPTION**
tadj_offset_coarse_a(2:0)	Read offset into the 8 element memory for the UMTS or CDMA mode A channel DDC.
tadj_offset_coarse_b(2:0)	Read offset into the 8 element memory for the CDMA mode B channel DDC when in CDMA mode.
tadj_offset_fine_a(2:0)	Controls the zero pad (or stuff) insert offset (fine adjust) for the UMTS or CDMA mode A channel of the DDC.
tadj_offset_fine_b(2:0)	Controls the zero pad (or stuff) insert offset (fine adjust) for the CDMA mode B channel of the DDC when in CDMA mode.
tadj_interp(2:0)	The interpolation value (1, 2, 4, or 8). Same used for both the A and B channels when in CDMA mode. Selects the number of zeros to be inserted.
ssel_tadj_fine(2:0)	Selects the sync source for the fine time adjust zero stuff moment. Same for A and B channels when in CDMA mode.
ssel_tadj_reg(2:0)	Selects the sync source used to update the double buffer course and fine delay selection registers. Same for A and B channels when in CDMA mode.

2.2.6 DDC CIC Filter

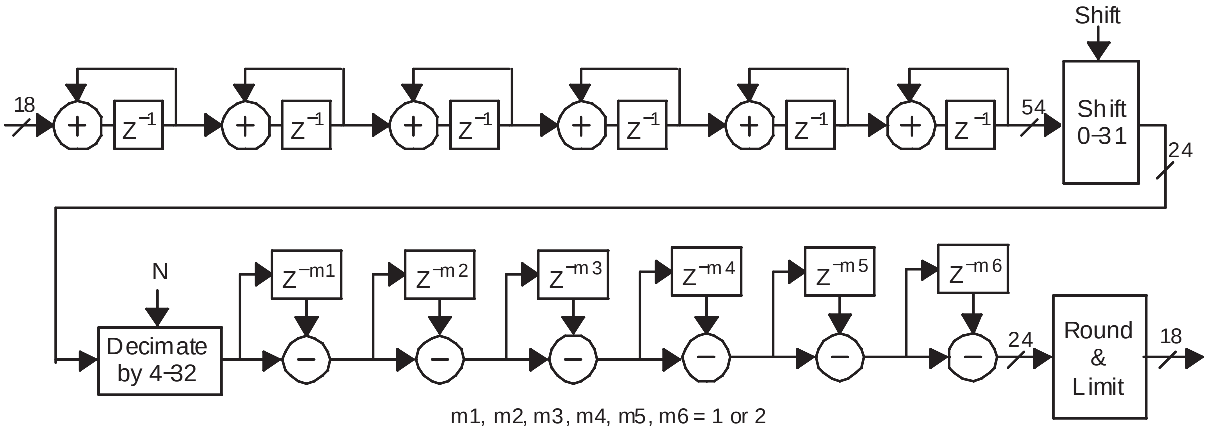


Figure 2-23. DDC CIC Filter Block Diagram

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 6x(Sin(x)/x) in character where the key attribute is that the resulting response nulls reject signal aliases from decimation. A consequence of this desirable behavior is that only a small portion of the passband can be used, less than 25% generally. 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.

The filter is equivalent to 6 stages of a FIR filter with uniform coefficients (6 combined boxcar filter stages). Each filter would be of length N if m=1, or 2N 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-rcic_shift (a programmable factor from 0 to 31 bits).

The CIC filter exhibits a droop across its frequency response. The following CFIR filter compensates for the CIC droop with a gradually rising frequency response. It is also possible to compensate for CIC droop in the PFIR filter.

The gain of the receive CIC filter is:

$\mathrm{Ncic}^6 \times 2^{(\text{number of stages where } M=2)} \times 2^{(-36+\mathrm{RCIC\_SHIFT})}$ where RCIC_SHIFT 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 can only compensate for peak gain less than one or must be less than or equal to one. A fixed gain of +12 dB at the output of the CIC can also be programmed.

PROGRAMMING	
VARIABLE	DESCRIPTION
cic_decim(4:0)	The CIC decimation ratio (4 to 32). The ratio is cic_decim + 1. This ratio applies to both A and B channels of the DDC block in CDMA mode.
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_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 UMTS or CDMA mode A channel.
cic_m2_ena_b (5:0)	Sets the differential delay value M for each of the CIC subtractor stages for the CDMA mode B channel.
cic_bypass	Bypasses the CIC filter when set, for factory testing.
ssel_cic(2:0)	Sets syncing (1 of 8 sources) for the CIC decimation moment.

2.2.7 DDC Compensating FIR Filter (CFIR)

The receive compensating FIR filter (CFIR) 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).

The filter is organized in two partial filter blocks, each containing a data RAM, a coefficient RAM and a dual multiplier, a common state machine and output accumulator.

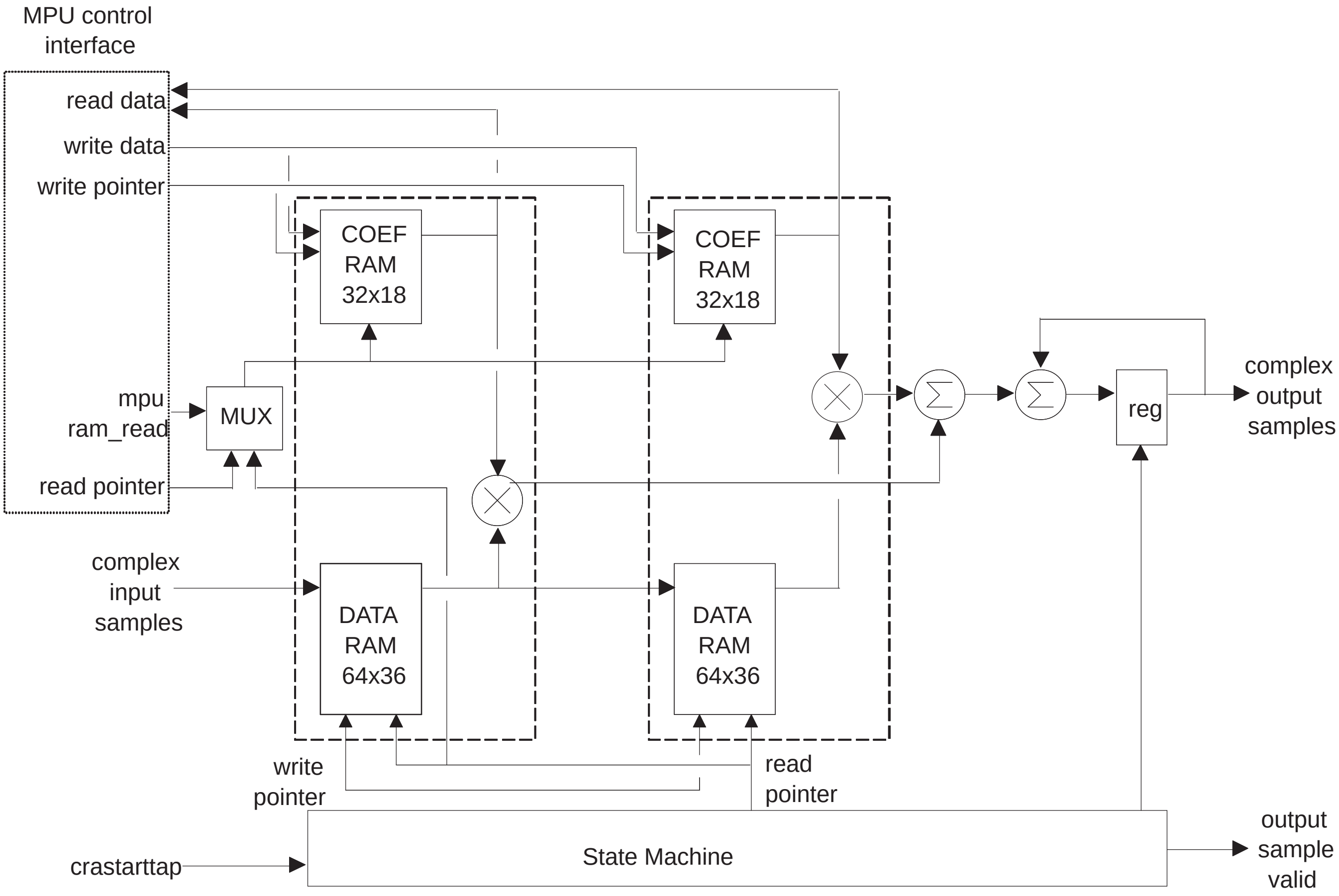


Figure 2-24. DDC CFIR Block Diagram

The maximum CFIR filter length is a function of GC5018 clock rate, output sample rate and 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:

UMTS Mode: 2 x (rxclk ÷ output sample rate)

CDMA Mode if cic_decim is even (decimating by an odd number): 2 x (cic_decim)

CDMA Mode if cic_decim is odd (decimating by an even number): 2 x (cic_decim + 1)

Example CFIR filter lengths available based on mode and rxclk frequency:

Mode	rxclk (MHz)	CIC DECIMATION	cic_decim	CFIR MAX LENGTH	CFIR MIN LENGTH	COMMENTS
UMTS	153.60	10	9	40	14	UMTS
UMTS	122.88	8	7	32	14	UMTS
CDMA	157.2864	32	31	64	14	CDMA2000
CDMA	122.88	25	24	48	14	CDMA2000
CDMA	78.6432	16	15	32	14	CDMA2000 low power configuration
CDMA	153.60	30	29	60	14	TD-SCDMA
CDMA	81.92	16	15	32	14	TD-SCDMA
CDMA	76.80	15	14	28	14	TD-SCDMA low power configuration

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.

After the CFIR filter performs the convolution, gain is applied at full precision, the signal is rounded, and then hard limited. A shifter at the output of the filter then scales the data by either 2e-19 or 2e-18. The gain through the filter is therefore:

$$\text{Sum(CFIR coefficients) x } 2^{-(18 \text{ or } 19)}$$

Coefficients are organized in two groups of 32 words, each 18 bits wide. For fully utilized filters, the 64 coefficients are loaded 0 through 31 into the first RAM, and 32 through 63 into the second RAM. The 16 bit MSBs and 2 bit LSBs are written into the RAMs using different page register values. Shorter filters require the coefficients be loaded into the 2 rams equally, starting from address 0.

For example, a CFIR coefficient set for a symmetric 58 tap TD-SCDMA CFIR is:

Taps	Coefficient	Taps	Coefficient
0 = 57	–13	15 = 42	–4975
1 = 56	–20	16 = 41	–4649
2 = 55	14	17 = 40	–232
3 = 54	101	18 = 39	6581
4 = 53	184	19 = 38	11266
5 = 52	133	20 = 37	8917
6 = 51	–147	21 = 36	–1957
7 = 50	–562	22 = 35	–16736
8 = 49	–768	23 = 34	–25469
9 = 48	–364	24 = 33	–17599
10 = 47	719	25 = 32	11560
11 = 46	1905	26 = 31	56455
12 = 45	2126	27 = 30	102215
13 = 44	567	28 = 29	131071
14 = 43	–2416		

The first 29 coefficients are loaded into addresses 0 through 28 in the first coefficient RAM, and the remaining 29 are loaded into addresses 0 through 28 in the second coefficient RAM. Loading the 18 bit coefficients requires 2 writes per coefficient, one for the upper 16 bits and another for the lower 2 bits.

To program this coefficient set for the DDC2 CFIR, the following control microprocessor interface sequence would be used.

Step	Address a[5:0]	Data d[15:0]	Description
1	0x21	0x0480	*Page register for DDC2 CFIR Coefficient RAM 0-31, LSBs.*
2	0x00	0x0003	2 lower bits of coefficient 0
3	0x01	0x0000	2 lower bits of coefficient 1
4	0x02	0x0002	2 lower bits of coefficient 2
5	0x03	0x0001	2 lower bits of coefficient 3
6	0x04	0x0000	2 lower bits of coefficient 4
7	0x05	0x0001	2 lower bits of coefficient 5
8	0x06	0x0001	2 lower bits of coefficient 6
9	0x07	0x0002	2 lower bits of coefficient 7
10	0x08	0x0000	2 lower bits of coefficient 8
11	0x09	0x0000	2 lower bits of coefficient 9
12	0x0A	0x0003	2 lower bits of coefficient 10

Step	Address a[5:0]	Data d[15:0]	Description
13	0x0B	0x0001	2 lower bits of coefficient 11
14	0x0C	0x0002	2 lower bits of coefficient 12
15	0x0D	0x0003	2 lower bits of coefficient 13
16	0x0E	0x0000	2 lower bits of coefficient 14
17	0x0F	0x0001	2 lower bits of coefficient 15
18	0x10	0x0003	2 lower bits of coefficient 16
19	0x11	0x0000	2 lower bits of coefficient 17
20	0x12	0x0001	2 lower bits of coefficient 18
21	0x13	0x0002	2 lower bits of coefficient 19
22	0x14	0x0001	2 lower bits of coefficient 20
23	0x15	0x0003	2 lower bits of coefficient 21
24	0x16	0x0000	2 lower bits of coefficient 22
25	0x17	0x0003	2 lower bits of coefficient 23
26	0x18	0x0001	2 lower bits of coefficient 24
27	0x19	0x0000	2 lower bits of coefficient 25
28	0x1A	0x0003	2 lower bits of coefficient 26
29	0x1B	0x0003	2 lower bits of coefficient 27
30	0x1C	0x0003	2 lower bits of coefficient 28
31	0x1D	0x0000	2 lower bits of unused coefficient RAM location
32	0x1E	0x0000	2 lower bits of unused coefficient RAM location
33	0x1F	0x0000	2 lower bits of unused coefficient RAM location
34	0x21	0x04A0	*Page register for DDC2 CFIR Coefficient RAM 32-63, LSBs.*
35	0x00	0x0003	2 lower bits of coefficient 29
36	0x01	0x0003	2 lower bits of coefficient 30
37	0x02	0x0003	2 lower bits of coefficient 31
38	0x03	0x0000	2 lower bits of coefficient 32
39	0x04	0x0001	2 lower bits of coefficient 33
40	0x05	0x0003	2 lower bits of coefficient 34
41	0x06	0x0000	2 lower bits of coefficient 35
42	0x07	0x0003	2 lower bits of coefficient 36
43	0x08	0x0001	2 lower bits of coefficient 37
44	0x09	0x0002	2 lower bits of coefficient 38
45	0x0A	0x0001	2 lower bits of coefficient 39
46	0x0B	0x0000	2 lower bits of coefficient 40
47	0x0C	0x0003	2 lower bits of coefficient 41
48	0x0D	0x0001	2 lower bits of coefficient 42
49	0x0E	0x0000	2 lower bits of coefficient 43
50	0x0F	0x0003	2 lower bits of coefficient 44
51	0x10	0x0002	2 lower bits of coefficient 45
52	0x11	0x0001	2 lower bits of coefficient 46
53	0x12	0x0003	2 lower bits of coefficient 47
54	0x13	0x0000	2 lower bits of coefficient 48
55	0x14	0x0000	2 lower bits of coefficient 49
56	0x15	0x0002	2 lower bits of coefficient 50
57	0x16	0x0001	2 lower bits of coefficient 51
58	0x17	0x0001	2 lower bits of coefficient 52
59	0x18	0x0000	2 lower bits of coefficient 53
60	0x19	0x0001	2 lower bits of coefficient 54

Step	Address a[5:0]	Data d[15:0]	Description
61	0x1A	0x0002	2 lower bits of coefficient 55
62	0x1B	0x0000	2 lower bits of coefficient 56
63	0x1C	0x0003	2 lower bits of coefficient 57
64	0x1D	0x0000	2 lower bits of unused coefficient RAM location
65	0x1E	0x0000	2 lower bits of unused coefficient RAM location
66	0x1F	0x0000	2 lower bits of unused coefficient RAM location
67	0x21	0x04C0	*Page register for DDC2 CFIR Coefficient RAM 0-31, MSBs.*
68	0x00	0xFFFC	Upper 16 bits of coefficient 0
69	0x01	0xFFFB	Upper 16 bits of coefficient 1
70	0x02	0x0003	Upper 16 bits of coefficient 2
71	0x03	0x0019	Upper 16 bits of coefficient 3
72	0x04	0x002E	Upper 16 bits of coefficient 4
73	0x05	0x0021	Upper 16 bits of coefficient 5
74	0x06	0xFFDB	Upper 16 bits of coefficient 6
75	0x07	0xFF73	Upper 16 bits of coefficient 7
76	0x08	0xFF40	Upper 16 bits of coefficient 8
77	0x09	0xFFA5	Upper 16 bits of coefficient 9
78	0x0A	0x00B3	Upper 16 bits of coefficient 10
79	0x0B	0x01DC	Upper 16 bits of coefficient 11
80	0x0C	0x0213	Upper 16 bits of coefficient 12
81	0x0D	0x008D	Upper 16 bits of coefficient 13
82	0x0E	0xFDA4	Upper 16 bits of coefficient 14
83	0x0F	0xFB24	Upper 16 bits of coefficient 15
84	0x10	0xFB75	Upper 16 bits of coefficient 16
85	0x11	0xFFC6	Upper 16 bits of coefficient 17
86	0x12	0x066D	Upper 16 bits of coefficient 18
87	0x13	0x0B00	Upper 16 bits of coefficient 19
88	0x14	0x08B5	Upper 16 bits of coefficient 20
89	0x15	0xFE16	Upper 16 bits of coefficient 21
90	0x16	0xEFA8	Upper 16 bits of coefficient 22
91	0x17	0xE720	Upper 16 bits of coefficient 23
92	0x18	0xEED0	Upper 16 bits of coefficient 24
93	0x19	0x0B4A	Upper 16 bits of coefficient 25
94	0x1A	0x3721	Upper 16 bits of coefficient 26
95	0x1B	0x63D1	Upper 16 bits of coefficient 27
96	0x1C	0x7FFF	Upper 16 bits of coefficient 28
97	0x1D	0x0000	Upper 16 bits of unused coefficient RAM location
98	0x1E	0x0000	Upper 16 bits of unused coefficient RAM location
99	0x1F	0x0000	Upper 16 bits of unused coefficient RAM location
100	0x21	0x04E0	*Page register for DDC2 CFIR Coefficient RAM 32-63, MSBs.*
101	0x00	0x7FFF	Upper 16 bits of coefficient 29
102	0x01	0x63D1	Upper 16 bits of coefficient 30
103	0x02	0x3721	Upper 16 bits of coefficient 31
104	0x03	0x0B4A	Upper 16 bits of coefficient 32
105	0x04	0xEED0	Upper 16 bits of coefficient 33
106	0x05	0xE720	Upper 16 bits of coefficient 34
107	0x06	0xEFA8	Upper 16 bits of coefficient 35
108	0x07	0xFE16	Upper 16 bits of coefficient 36

Step	Address a[5:0]	Data d[15:0]	Description
109	0x08	0x08B5	Upper 16 bits of coefficient 37
110	0x09	0x0B00	Upper 16 bits of coefficient 38
111	0x0A	0x066D	Upper 16 bits of coefficient 39
112	0x0B	0xFFC6	Upper 16 bits of coefficient 40
113	0x0C	0xFB75	Upper 16 bits of coefficient 41
114	0x0D	0xFB24	Upper 16 bits of coefficient 42
115	0x0E	0xFDA4	Upper 16 bits of coefficient 43
116	0x0F	0x008D	Upper 16 bits of coefficient 44
117	0x10	0x0213	Upper 16 bits of coefficient 45
118	0x11	0x01DC	Upper 16 bits of coefficient 46
119	0x12	0x00B3	Upper 16 bits of coefficient 47
120	0x13	0xFFA5	Upper 16 bits of coefficient 48
121	0x14	0xFF40	Upper 16 bits of coefficient 49
122	0x15	0xFF73	Upper 16 bits of coefficient 50
123	0x16	0xFFDB	Upper 16 bits of coefficient 51
124	0x17	0x0021	Upper 16 bits of coefficient 52
125	0x18	0x002E	Upper 16 bits of coefficient 53
126	0x19	0x0019	Upper 16 bits of coefficient 54
127	0x1A	0x0003	Upper 16 bits of coefficient 55
128	0x1B	0xFFFB	Upper 16 bits of coefficient 56
129	0x1C	0xFFFC	Upper 16 bits of coefficient 57
130	0x1D	0x0000	Upper 16 bits of unused coefficient RAM location
131	0x1E	0x0000	Upper 16 bits of unused coefficient RAM location
132	0x1F	0x0000	Upper 16 bits of unused coefficient RAM location
133	0x21	0x0500	*Page register for DDC2 control registers 0-31*
134	0x00	0x8EE0	DDC2 FIR_MODE register; cdma_mode enabled, 60 tap PFIR, 58 tap CFIR
135	0x01	0x2000	DDC2 PFIR gain = sum(taps)x2^–18 and CFIR gain = sum(taps)x2^–19

PROGRAMMING	
VARIABLE	**DESCRIPTION**
crastarttap_cfir(4:0)	Number of DDC CFIR filter taps is 2x(crastarttap + 1)
mpu_ram_read	What set, the PFIR and CFIR coefficient rams are readable via the MPU control interface. *The GC5018 signal path is not operational when this bit is set, it is intended for debug purposes only.*
cfir_gain	0 = $2e^{-19}$, 1 = $2e^{-18}$
The CFIR filter's 18 bit coefficients are loaded in two 32 word memories. **Note:** CFIR filter coefficients are shared between A and B channels of a DDC block in CDMA mode.	

2.2.8 DDC Programmable FIR Filter (PFIR)

The receive programmable FIR filter (PFIR) provides final pulse shaping of 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 & 2k+1, k=0 to 7) can be combined to yield a filter with twice the number of coefficients. This means the GC5018 can support 4 UMTS DDC channels with double-length filter coefficients (up to 128 taps).

The filter is organized in four partial filter blocks, each containing a data RAM, a coefficient RAM and a dual multiplier, a common state machine and output accumulator.

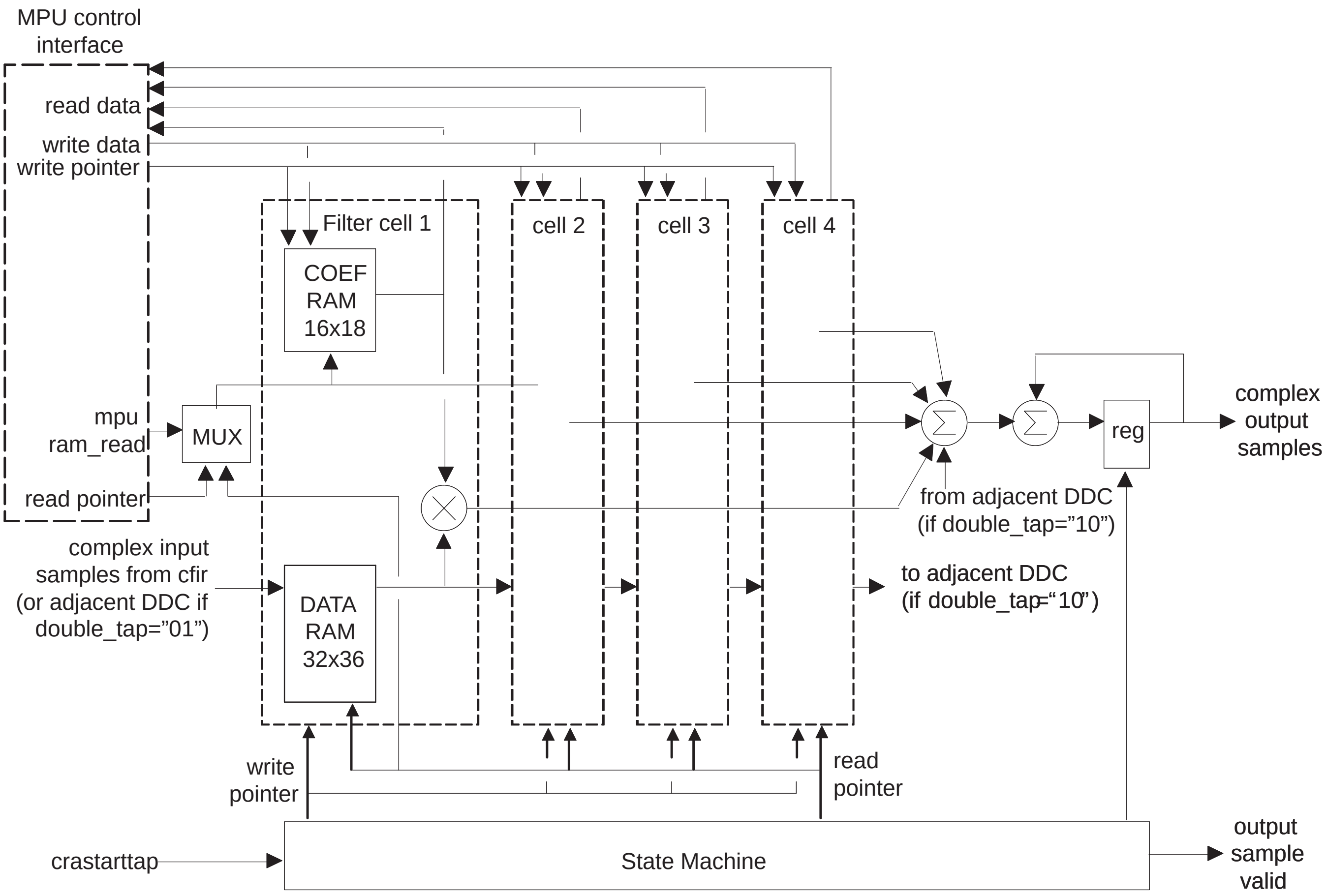


Figure 2-25. DDC PFIR Block Diagram

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: sum(coefficients) × 2^{-shift}, where shift ranges from 12 to 19.

The maximum PFIR filter length is a function of GC5018 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 32 (for both CDMA and UMTS). Lengths between these limits can be specified in increments of 4. For strapped UMTS with double length filters, the range of taps available is 64 to 128 in increments of 8.

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

UMTS Mode: 4 × (CIC DECIMATION × 2)

Strapped UMTS Mode: 8 × (CIC DECIMATION × 2)

CDMA Mode: 2 × (CIC DECIMATION × 2)

PFIR coefficients and gain shift values are shared between both A and B CDMA channels in a DDC block.

Example PFIR filter lengths available based on mode and rxclk frequency:

Mode	rxclk (MHz)	CIC DECIMATION	PFIR MAX LENGTH	PFIR MIN LENGTH	COMMENTS
UMTS	153.60	10	64	32	UMTS, 1 to 6 DDC channels
UMTS	122.88	8	64	32	UMTS, 1 to 6 DDC channels
UMTS	153.60	10	128	64	Strapped UMTS double length PFIR configuration; 1, 2 or 3 DDC channels.
UMTS	122.88	8	128	64	Strapped UMTS double length PFIR configuration; 1, 2 or 3 DDC channels
CDMA	157.2864	32	64	32	CDMA2000
CDMA	122.88	25	64	32	CDMA2000
CDMA	78.6432	16	64	32	CDMA2000 low power configuration
CDMA	153.60	30	64	32	TD-SCDMA
CDMA	81.92	16	64	32	TD-SCDMA
CDMA	76.80	15	60	32	TD-SCDMA low power configuration

Coefficients are organized in four groups of 16 words, each 18 bits wide. For fully utilized filters, the 64 coefficients are loaded 0 through 31 into the first and second RAMs, and 32 through 63 into the third and fourth RAMs. The 16 bit MSBs and 2 bit LSBs are written into the RAMs using different page register values. Shorter filters require the coefficients be loaded into the 4 rams equally, starting from address 0 and address 16.

For example, a CFIR coefficient set for a symmetric 60 tap TD-SCDMA PFIR is:

Taps	Coefficient	Taps	Coefficient
0 = 59	–2	15 = 44	420
1 = 58	1	16 = 43	–331
2 = 57	4	17 = 42	–319
3 = 56	–8	18 = 41	744
4 = 55	–2	19 = 40	–440
5 = 54	21	20 = 39	–1005
6 = 53	–13	21 = 38	2389
7 = 52	–28	22 = 37	514
8 = 51	46	23 = 36	–6182
9 = 50	1	24 = 35	1845
10 = 49	–85	25 = 34	12959
11 = 48	96	26 = 33	–8691
12 = 47	82	27 = 32	–27246
13 = 46	–266	28 = 31	34166
14 = 45	38	29 = 30	131071

The first 15 coefficients are loaded into addresses 0 through 14 in the first coefficient RAM, the second group of 15 are loaded into addresses 16 through 30 corresponding to the second coefficient RAM, the third group of 15 are loaded into the third coefficient ram at addresses 0 through 14, and the fourth group of 15 are loaded into addresses 16 through 30 in the fourth coefficient RAM. Loading the 18 bit coefficients requires 2 writes per coefficient, one for the upper 16 bits and another for the lower 2 bits.

To program this coefficient set for the DDC2 PFIR, the following control microprocessor interface sequence would be used.

Step	Address a[5:0]	Data d[15:0]	Description
1	0x21	0x0400	*Page register for DDC2 CFIR Coefficient RAMs 0-15 and 16-31, LSBs.*
2	0x00	0x0002	2 lower bits of coefficient 0

Step	Address a[5:0]	Data d[15:0]	Description
3	0x01	0x0001	2 lower bits of coefficient 1
4	0x02	0x0000	2 lower bits of coefficient 2
5	0x03	0x0000	2 lower bits of coefficient 3
6	0x04	0x0002	2 lower bits of coefficient 4
7	0x05	0x0001	2 lower bits of coefficient 5
8	0x06	0x0003	2 lower bits of coefficient 6
9	0x07	0x0000	2 lower bits of coefficient 7
10	0x08	0x0002	2 lower bits of coefficient 8
11	0x09	0x0001	2 lower bits of coefficient 9
12	0x0A	0x0003	2 lower bits of coefficient 10
13	0x0B	0x0000	2 lower bits of coefficient 11
14	0x0C	0x0002	2 lower bits of coefficient 12
15	0x0D	0x0002	2 lower bits of coefficient 13
16	0x0E	0x0002	2 lower bits of coefficient 14
17	0x0F	0x0000	2 lower bits of unused coefficient RAM location
18	0x10	0x0000	2 lower bits of coefficient 15
19	0x11	0x0001	2 lower bits of coefficient 16
20	0x12	0x0001	2 lower bits of coefficient 17
21	0x13	0x0000	2 lower bits of coefficient 18
22	0x14	0x0000	2 lower bits of coefficient 19
23	0x15	0x0003	2 lower bits of coefficient 20
24	0x16	0x0001	2 lower bits of coefficient 21
25	0x17	0x0002	2 lower bits of coefficient 22
26	0x18	0x0002	2 lower bits of coefficient 23
27	0x19	0x0001	2 lower bits of coefficient 24
28	0x1A	0x0003	2 lower bits of coefficient 25
29	0x1B	0x0001	2 lower bits of coefficient 26
30	0x1C	0x0002	2 lower bits of coefficient 27
31	0x1D	0x0002	2 lower bits of coefficient 28
32	0x1E	0x0003	2 lower bits of coefficient 29
33	0x1F	0x0000	2 lower bits of unused coefficient RAM location
34	0x21	0x0420	*Page register for DDC2 CFIR Coefficient RAMs 32-47 and 48-63, LSBs.*
35	0x00	0x0003	2 lower bits of coefficient 30
36	0x01	0x0002	2 lower bits of coefficient 31
37	0x02	0x0002	2 lower bits of coefficient 32
38	0x03	0x0001	2 lower bits of coefficient 33
39	0x04	0x0003	2 lower bits of coefficient 34
40	0x05	0x0001	2 lower bits of coefficient 35
41	0x06	0x0002	2 lower bits of coefficient 36
42	0x07	0x0002	2 lower bits of coefficient 37
43	0x08	0x0001	2 lower bits of coefficient 38
44	0x09	0x0003	2 lower bits of coefficient 39
45	0x0A	0x0000	2 lower bits of coefficient 40
46	0x0B	0x0000	2 lower bits of coefficient 41
47	0x0C	0x0001	2 lower bits of coefficient 42
48	0x0D	0x0001	2 lower bits of coefficient 43
49	0x0E	0x0000	2 lower bits of coefficient 44
50	0x0F	0x0000	2 lower bits of unused coefficient RAM location

Step	Address a[5:0]	Data d[15:0]	Description
51	0x10	0x0002	2 lower bits of coefficient 45
52	0x11	0x0002	2 lower bits of coefficient 46
53	0x12	0x0002	2 lower bits of coefficient 47
54	0x13	0x0000	2 lower bits of coefficient 48
55	0x14	0x0003	2 lower bits of coefficient 49
56	0x15	0x0001	2 lower bits of coefficient 50
57	0x16	0x0002	2 lower bits of coefficient 51
58	0x17	0x0000	2 lower bits of coefficient 52
59	0x18	0x0003	2 lower bits of coefficient 53
60	0x19	0x0001	2 lower bits of coefficient 54
61	0x1A	0x0002	2 lower bits of coefficient 55
62	0x1B	0x0000	2 lower bits of coefficient 56
63	0x1C	0x0000	2 lower bits of coefficient 57
64	0x1D	0x0001	2 lower bits of coefficient 58
65	0x1E	0x0002	2 lower bits of coefficient 59
66	0x1F	0x0000	2 lower bits of unused coefficient RAM location
67	0x21	0x0440	*Page register for DDC2 PFIR Coefficient RAMs 0-15 and 16-31, MSBs.*
68	0x00	0xFFFF	Upper 16 bits of coefficient 0
69	0x01	0x0000	Upper 16 bits of coefficient 1
70	0x02	0x0001	Upper 16 bits of coefficient 2
71	0x03	0xFFFE	Upper 16 bits of coefficient 3
72	0x04	0xFFFF	Upper 16 bits of coefficient 4
73	0x05	0x0005	Upper 16 bits of coefficient 5
74	0x06	0xFFFC	Upper 16 bits of coefficient 6
75	0x07	0xFFF9	Upper 16 bits of coefficient 7
76	0x08	0x000B	Upper 16 bits of coefficient 8
77	0x09	0x0000	Upper 16 bits of coefficient 9
78	0x0A	0xFFEA	Upper 16 bits of coefficient 10
79	0x0B	0x0018	Upper 16 bits of coefficient 11
80	0x0C	0x0014	Upper 16 bits of coefficient 12
81	0x0D	0xFFBD	Upper 16 bits of coefficient 13
82	0x0E	0x0009	Upper 16 bits of coefficient 14
83	0x0F	0x0000	Upper 16 bits of unused coefficient RAM location
84	0x10	0x0069	Upper 16 bits of coefficient 15
85	0x11	0xFFAD	Upper 16 bits of coefficient 16
86	0x12	0x0FFB0	Upper 16 bits of coefficient 17
87	0x13	0x0B0A	Upper 16 bits of coefficient 18
88	0x14	0xFF92	Upper 16 bits of coefficient 19
89	0x15	0xFF04	Upper 16 bits of coefficient 20
90	0x16	0x0255	Upper 16 bits of coefficient 21
91	0x17	0x0080	Upper 16 bits of coefficient 22
92	0x18	0xF9F6	Upper 16 bits of coefficient 23
93	0x19	0x01CD	Upper 16 bits of coefficient 24
94	0x1A	0x0CA7	Upper 16 bits of coefficient 25
95	0x1B	0xF783	Upper 16 bits of coefficient 26
96	0x1C	0xE564	Upper 16 bits of coefficient 27
97	0x1D	0x215D	Upper 16 bits of coefficient 28
98	0x1E	0x7FFF	Upper 16 bits of coefficient 29

Step	Address a[5:0]	Data d[15:0]	Description
99	0x1F	0x0000	Upper 16 bits of unused coefficient RAM location
100	0x21	0x0460	*Page register for DDC2 PFIR Coefficient RAMS 32-47 AND 48-63, MSBs.*
101	0x00	0x7FFF	Upper 16 bits of coefficient 30
102	0x01	0x215D	Upper 16 bits of coefficient 31
103	0x02	0xE564	Upper 16 bits of coefficient 32
104	0x03	0xF783	Upper 16 bits of coefficient 33
105	0x04	0x0CA7	Upper 16 bits of coefficient 34
106	0x05	0x01CD	Upper 16 bits of coefficient 35
107	0x06	0xF9F6	Upper 16 bits of coefficient 36
108	0x07	0x0080	Upper 16 bits of coefficient 37
109	0x08	0x0255	Upper 16 bits of coefficient 38
110	0x09	0xFF04	Upper 16 bits of coefficient 39
111	0x0A	0xFF92	Upper 16 bits of coefficient 40
112	0x0B	0x00BA	Upper 16 bits of coefficient 41
113	0x0C	0xFFB0	Upper 16 bits of coefficient 42
114	0x0D	0xFFAD	Upper 16 bits of coefficient 43
115	0x0E	0x0069	Upper 16 bits of coefficient 44
116	0x0F	0x008D	Upper 16 bits of unused coefficient RAM location
117	0x10	0x0009	Upper 16 bits of coefficient 45
118	0x11	0xFFBD	Upper 16 bits of coefficient 46
119	0x12	0x0014	Upper 16 bits of coefficient 47
120	0x13	0x0018	Upper 16 bits of coefficient 48
121	0x14	0xFFEA	Upper 16 bits of coefficient 49
122	0x15	0x0000	Upper 16 bits of coefficient 50
123	0x16	0x000B	Upper 16 bits of coefficient 51
124	0x17	0xFFF9	Upper 16 bits of coefficient 52
125	0x18	0xFFFC	Upper 16 bits of coefficient 53
126	0x19	0x0005	Upper 16 bits of coefficient 54
127	0x1A	0xFFFF	Upper 16 bits of coefficient 55
128	0x1B	0xFFFE	Upper 16 bits of coefficient 56
129	0x1C	0x0001	Upper 16 bits of coefficient 57
130	0x1D	0x0000	Upper 16 bits of coefficient 58
131	0x1E	0xFFFF	Upper 16 bits of coefficient 59
132	0x1F	0x0000	Upper 16 bits of unused coefficient RAM location
133	0x21	0x0500	*Page register for DDC2 control registers 0-31*
134	0x00	0x8EE0	DDC2 FIR_MODE register; cdma_mode enabled, 60 tap PFIR, 58 tap CFIR
135	0x01	0x2000	DDC2 PFIR gain = sum(taps)x2^–18 and CFIR gain = sum(taps)x2^–19

PROGRAMMING	
VARIABLE	**DESCRIPTION**
crastarttap_pfir(4:0)	Number of DDC PFIR filter taps is 4x(crastartap+1) For double length PFIR the number of taps is 8x(crastartap+1)
cdma_mode	When set, puts the CFIR & PFIR blocks in CDMA mode.
mpu_ram_read	What set, the PFIR and CFIR coefficient rams are readable via the MPU control interface. *The GC5018 signal path is not operational when this bit is set, it is intended for debug purposes only.*
pfir_gain(2:0)	Sets the gain of the PFIR filter. The range is from $2e^{-19}$ to $2e^{-12}$; “000”= $2e^{-19}$ and “111”= $2e^{-12}$
double_tap(1:0)	When set, puts two adjacent DDC (2k and 2k+1, k=0 to 2) in double length (from 64 to128 tap) UMTS mode. Set to “00” for normal mode. In double tap mode, data out of the last PFIR ram in the main DDC (DDC0, DDC2, DDC4 or DDC6) is sent to the adjacent secondary DDC (DDC1, DDC3, DDC5 or DDC7) PFIR as input thus forming a 128-tap delay line. Data received from the adjacent PFIR summers is added into the Main DDC’s PFIR sum to form the final output. When using double tap mode, set double_tap to “10” for the main DDC, and to “01” for the secondary DDC. When in double tap mode, the first half of the coefficients should be loaded into the main DDC (DDC0, DDC2, DDC4 or DDC6), the remaining coefficients are loaded into the secondary DDC (DDC1, DDC3, DDC5 or DDC7). In double tap mode, the main DDC must be turned on (ddc_ena=1), and the secondary DDC must be turned **off** (ddc_ena=0).
The PFIR filter’s 18 bit coefficients are loaded in four 16 word memories.	
Note: PFIR filter coefficients are shared between A and B channels of a DDC block when in CDMA mode.	

2.2.9 DDC RMS Power Meter

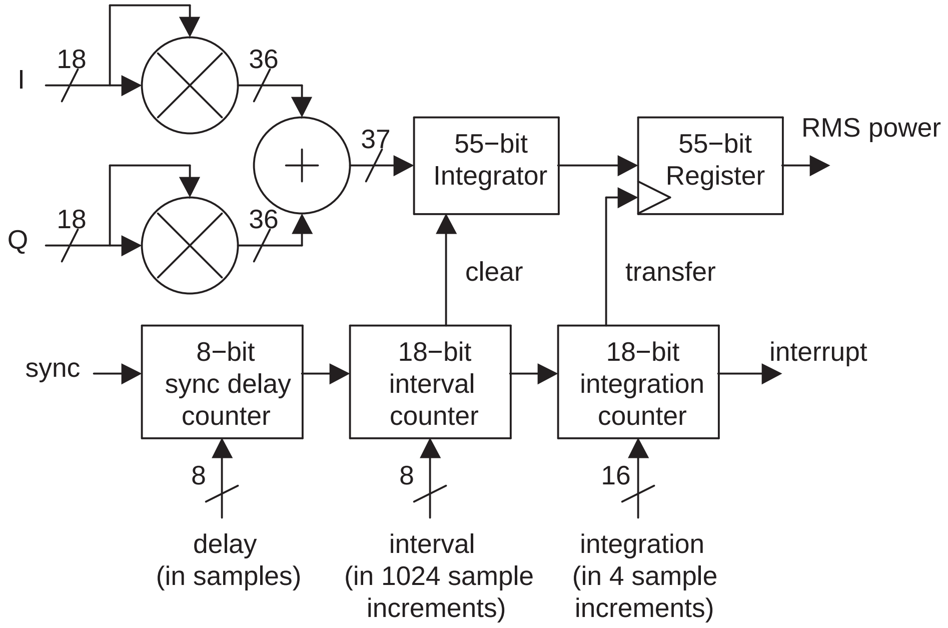


Figure 2-26. DDC RMS Power Meter Block Diagram

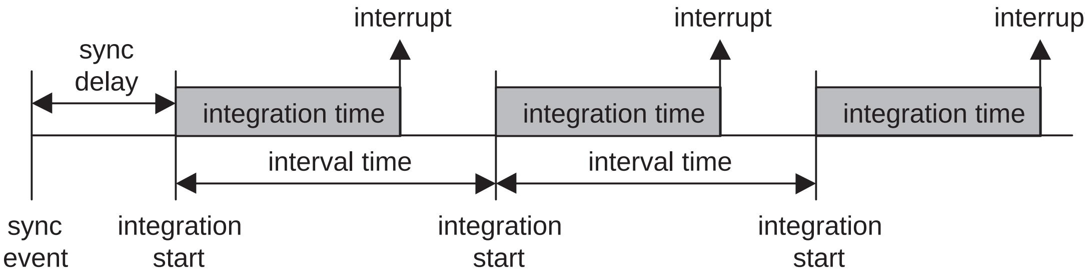


Figure 2-27. DDC RMS Power Meter Timing

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

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 period determined by a programmable counter. The integration time is a 16 bit word which is programmed into the 18 bit counter.

There is a programmable 18 bit interval timer which sets the interval over which power measurements are made. 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. For UMTS systems with sample rate rate at 7.68 MHz, the power meter interval range is from 133 µS to 34.1 mS. For a CDMA system with the sample rate at 2.4576 MHz, the power meter interval range is 417 µS to 107 mS.

The power measurement process starts with a sync event. The integration will start at sync event +3 chips + sync_delay. The 8 bit delay register permits delays from 1 to 256 samples after sync. The integration will continue until the integration count is met. At that point, the result in the 55 bit accumulator is transferred to the read holding 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 55 result bits are again transferred to the read register overwriting the previous value and an interrupt is generated signifying the data is ready to be read. Failure to read the data timely will result in overwriting the previous interval measurement.

Sync 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 (Signal) path and the B (Diversity) signal. As a result, there are two 55-bit words representing the Signal and Diversity when in CDMA mode.

The power read is:

power = $[(I^2 + Q^2) \times (N \times 4 + 1)]$ where N is the integration count.

PROGRAMMING	
VARIABLE	**DESCRIPTION**
pmeter_result_a(54:0)	55 bit UMTS or CDMA mode A channel power measurement result.
pmeter_result_b(54:0)	55 bit CDMA mode B channel power measurement result.
pmeter_sqr_sum_ddc(15:0)	Integration (square and sum) count in increments of four samples.
pmeter_sync_delay_ddc(7:0)	Sync delay count in samples.
pmeter_interval_ddc(7:0)	The measurement interval in increments of 2048 samples. This value must be greater than SQR_SUM.
ssel_pmeter(2:0)	Sync source selection.
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.

2.2.10 DDC AGC

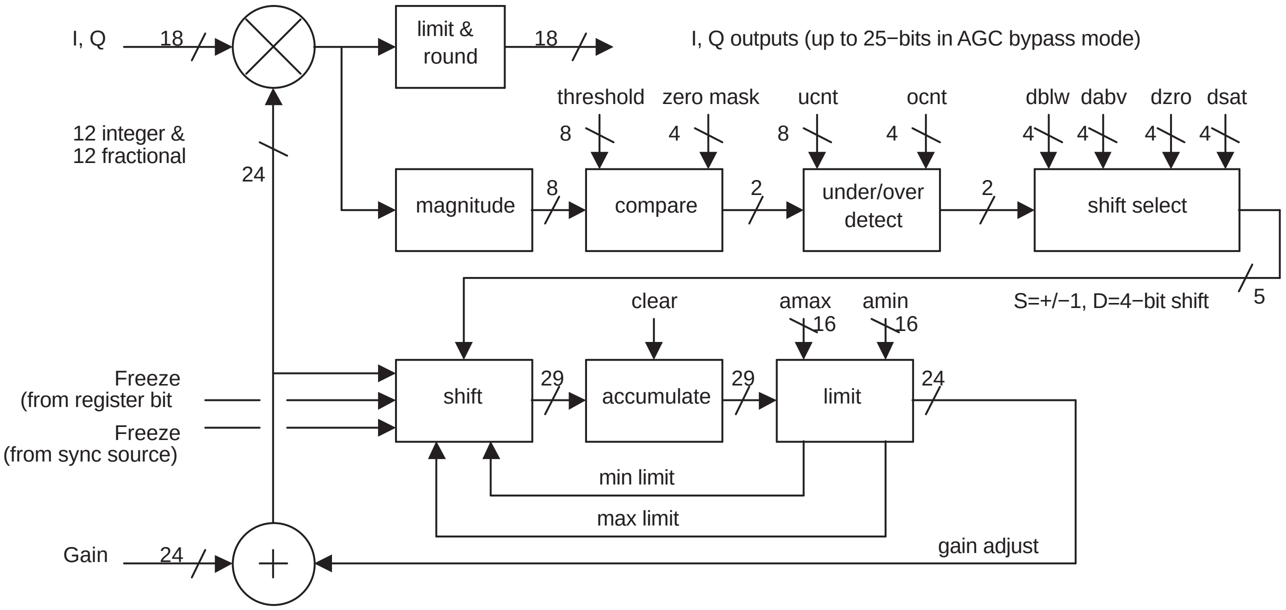


Figure 2-28. DDC AGC Block Diagram

The GC5018 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 24-bit gain word that represents a gain or attenuation in the range of 0 to 4096. The gain format is mixed integer and fraction. The 12-bit integer allows the gain to be boosted by up to factor of 4096 (72 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 4 and 18 bits in steps of one bit.

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)xSx$2^{–D}$, 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 AGC noise will be equal to ±$2^{–D}$ and the AGC attack and decay rate will be exponential, with a time constant equal to $2^{–D}$. Hence, the AGC will increase or decrease by 0.63 times G(t) in 2^{D} updates.

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 = 0.6745xRMS, 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 their gain increased rapidly.

Separate programmable D values allow the user to set different attack and decay time constants, and to set shorter time constants for when the signal falls too low (equal to zero), or is too high (saturates). The magnitude is considered to be consistently equal to zero by using a 4-bit counter that counts up every time the 8-bit magnitude value is zero, and counts down otherwise. If the counter's value exceeds a user specified threshold, then *Dabv* is used. Similarly the magnitude is considered too high 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, if the AGC's current gain at a particular moment in time is 5.123, and the magnitude of the signal is greater than zero, but less than the user-programmed threshold. Step size Dblw will be used to modify 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 x 2^{-5} = 5.123 + 0.160 = 5.283. If the signal's magnitude is still less than the user-programmed threshold, then the gain for the next sample will be 5.283 + 5.283 x 2^{-5} = 5.283 + 0.165 = 5.448. This continues until the 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 a 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 12dB (i.e., in the fast range detected by consistently zero or saturated data). This gives a step size of 0.5 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 724 samples (63% decay every 724 samples).

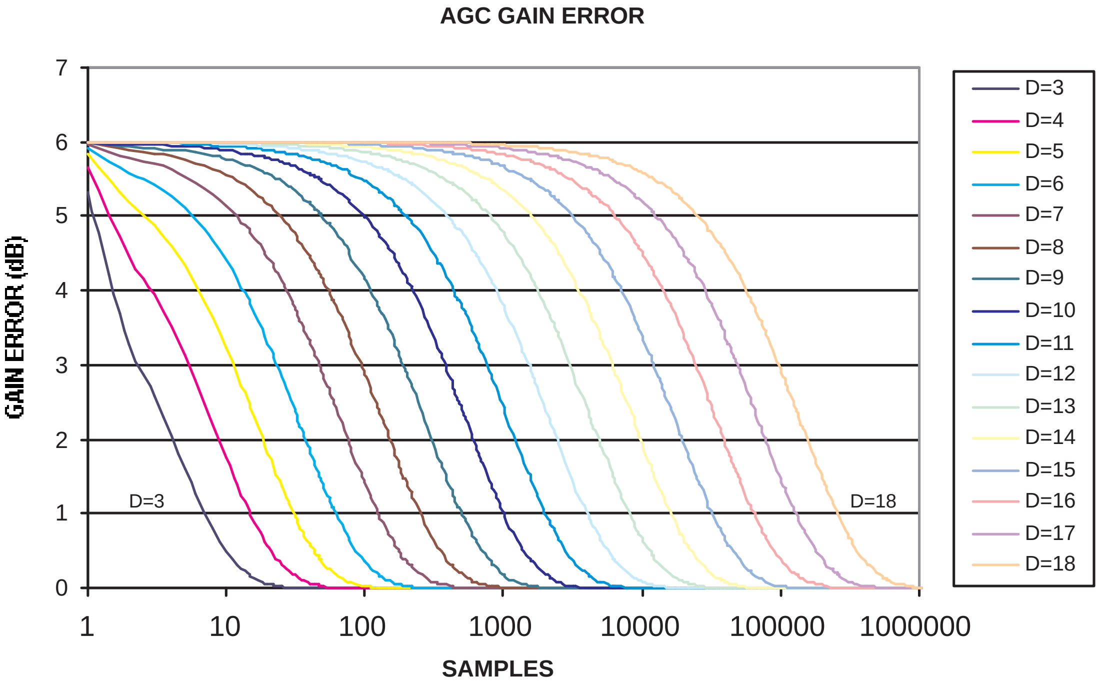


Figure 2-29. AGC Gain Error Over Time vs. D

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 –6xD dB below the signal RMS level. At D=10 (–60 dB) the error is negligible. The plot above 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 will put 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.

For TDD applications, freeze mode can be controlled using a sync source. This allows rxsync_a/b/c/d to be assigned as a AGC hold signal to keep the AGC from responding during the transmit interval and run during the receive interval. The freeze register bit is logically Ored with the freeze sync source.

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 appended to the 8 bit AGC-modified I and Q output data.

Output	Bits(17:10)	Bits(9:4)	Bits(3:2)	Bits(1:0)
I	I output data	Gain(23:16)		“00”
Q	Q output data	Gain(15:10)	AGC State(1:0)	“00”

PROGRAMMING	
VARIABLE	**DESCRIPTION**
agc_dblw(3:0)	**Below** threshold gain. Sets the value of gain step size Dblw (data x current gain below threshold). Ranges from 3 to 18, and maps to a 4 bit field. For example: 3 = “0000”, 4= “0001”, … 18= “1111”
agc_dabv(3:0)	**Above** threshold gain. Sets the value of gain step size Dabv (data x current gain above threshold). Ranges from 3 to 18, and maps to a 4 bit field. For example: 3 = “0000”, 4= “0001”, … 18= “1111”
agc_dzro(3:0)	**Zero** signal gain. Sets the value of gain step size Dzro (data x current gain consistently zero). Ranges from 3 to 18, and maps to a 4 bit field. For example: 3 = “0000”, 4= “0001”, … 18= “1111”
agc_dsat (3:0)	**Saturated** signal gain. Sets the value of gain step size Dsat (data x current gain consistently saturated). Ranges from 3 to 18, and maps to a 4 bit field. For example: 3 = “0000”, 4= “0001”, … 18= “1111”
agc_zero_msk(3:0)	Masks the lower 4 bits of signal data so as to be considered zeros.
agc_md(3:0)	AGC rounding. 0000= 18 bits out, 1111= 3 bits out.
agc_thresh(7:0)	AGC threshold. Compared with magnitude of 8 bits of input x gain.
agc_rnd_disable	AGC rounding is disabled when this bit is set.
agc_freeze	The AGC gain adjustment updates are disable when set.
agc_clear	The AGC gain adjustment accumulator is cleared when set
agc_gaina(23:0)	24 bit gain word for DDC A
agc_gainb(23:0)	24 bit gain word for DDC B (in CDMA mode)
agc_zero_cnt(3:0)	When the AGC output (input x gain) is zero value this number of times, the shoft value is changed to agc_dzero.
agc_max_cnt(3:0)	When the AGC output (input x gain) is zero value this number of times, the shift value is changed to agc_dsat.
agc_amax(15:0)	The maximum value that gain can be adjusted up to. Top 12 bits are integer, bottom 4 bits are fractional.
agc_amin(15:0)	The minimum value that gain can be adjusted down to. Top 12 bits are integer, bottom 4 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(23:16) - 00 Q Portion: 8 bits of AGC’d Q data - Gain(15:10) - Status(1:0) - 00.

PROGRAMMING	
VARIABLE	DESCRIPTION
	Note: Bit 0 of Status, when set, indicates the data is saturated. Bit 1 of Status, when set, indicates the data is zero.
ssel_agc_freeze(2:0)	Sync selection for freeze mode, 1 of 8 sources. This source is ORed with the freeze register bit
ssel_gain(2:0)	Sync selection for the double buffered agc_gaina and agc_gainb register.
ssel_ddc_agc(2:0)	Sync selection used to initialize the AGC, primarily for test purposes.

2.2.11 DDC Output Interface

The baseband I/Q sample interface can be configured as serial or parallel formatted data. The serial interface closely matches the GC5316 style interface. The parallel interface is provided to interface directly to the TMS320TCI110 when delayed antenna streams used to implement channel estimation buffering and/or transport format combination indicator (TFCI) buffering are not required.

The DDC output data is 2’s complement format.

2.2.11.1 Serial Output Interface

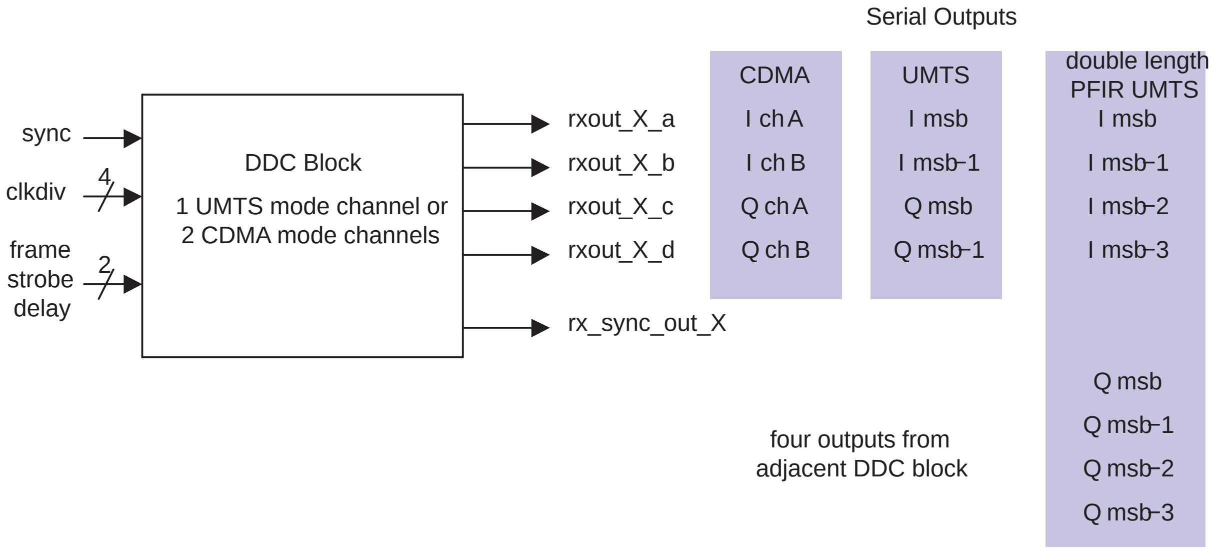


Figure 2-30. Serial Output Block Diagram and Output Pins for Each DDC Filter Mode

Each DDC block can be assigned four serial output data pins. These pins are used to transfer downconverted I/Q baseband data out of the GC5018 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.

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.

The frame strobe signal provided on the rx_sync_out_X pins can be programmed to arrive from 0 to 3 bit clocks early via a 2 bit control parameter. The frame interval can be programmed from 1 to 63 bits. A programmable 4-bit clock divider circuit is 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:

CDMA: [overall decimation × (pser_recv_8pin + 1)] / (pser_recv_clkdiv + 1) > pser_recv_bits + 1

UMTS: 2 × [overall decimation × (pser_recv_8pin + 1)] / (pser_recv_clkdiv + 1) > pser_recv_bits + 1

where overall decimation = CIC DECIMATION × 2.

Decimation by 2 in the output interface can be achieved by setting the frame strobe interval and clock divider to 1/2 the PFIR output rate. The serial interface samples the PFIR output each time the transfer interval defined by these two settings has completed. The decimation moment can be controlled using the rxsync_X input signal selected as the sync source for the serial interface.

The timing diagram below shows the DDC serial output timing.

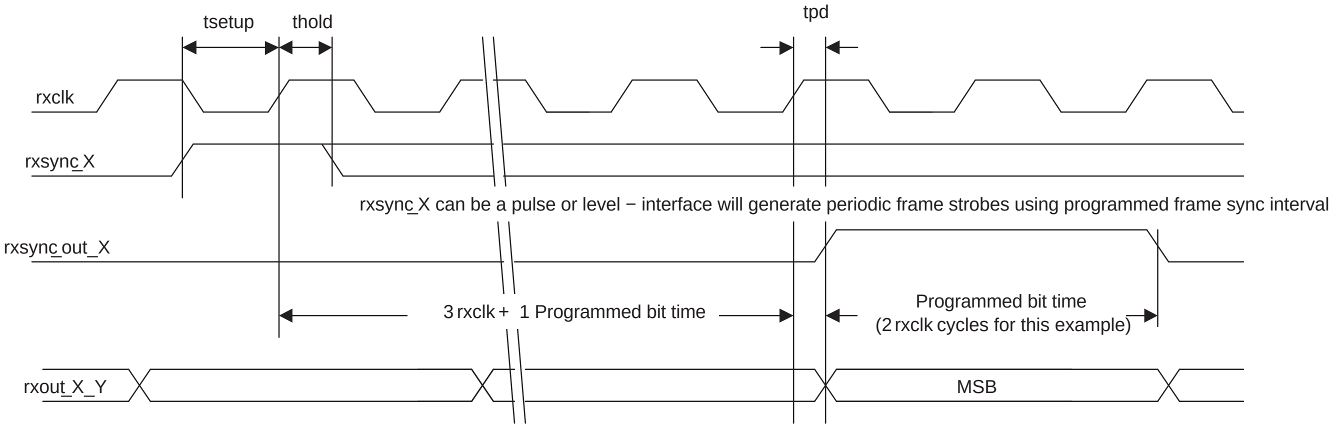

tsetup
thold
tpd
rxclk
rxsync_X
rxsync_X can be a pulse or level − interface will generate periodic frame strobes using programmed frame sync interval
rxsync_out_X
3 rxclk + 1 Programmed bit time
Programmed bit time
(2 rxclk cycles for this example)
rxout_X_Y
MSB

Figure 2-31. Serial Output Interface Timing Diagram

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= rcclk, 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 selection, 1 of 8.
tristate(6:3)	Tristate controls for the rx_sync_out_X and rxout_X_X pins. Pins are in tristate when the tristate register bits are set.

2.2.11.2 Parallel Output Interface

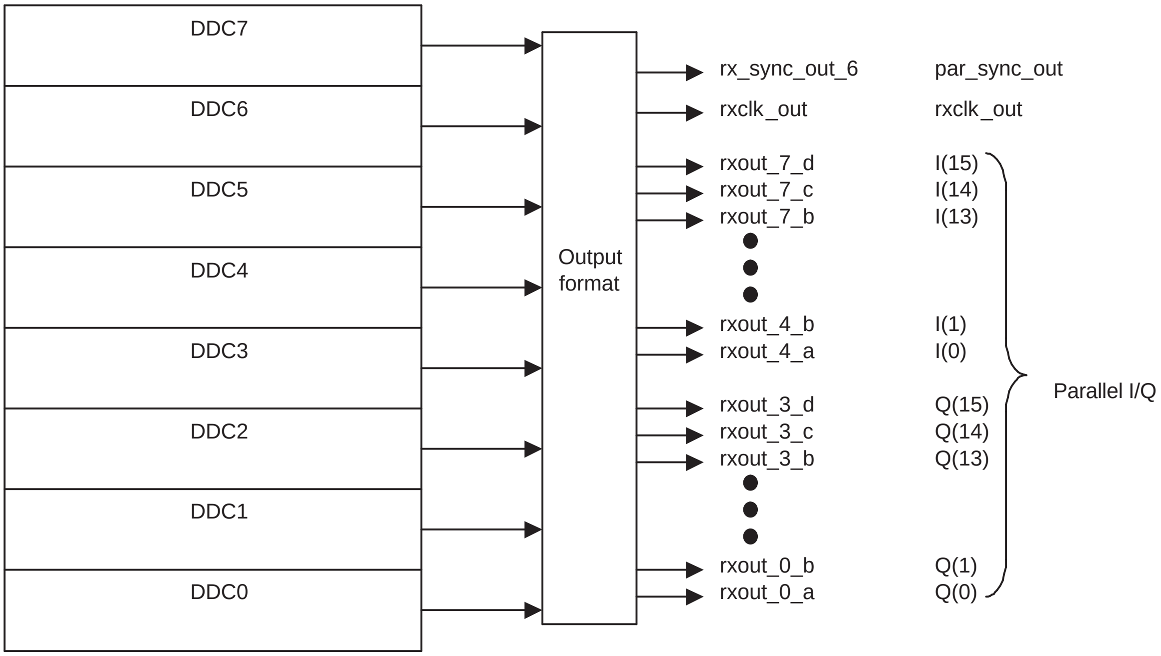


Figure 2-32. Parallel Output Interface Block Diagram and Output Pins

When a parallel I/Q interface is required, a 32 bit time division multiplexed output mode can be selected using the rxout_X_X pins. This interface is provided for direct connection to the TMS320TCI110 Receive Chip Rate ASSP when delayed antenna streams are not required. The output sample rate, rxclk_out clock polarity, par_sync_out position and number of channels to be output are all programmable.

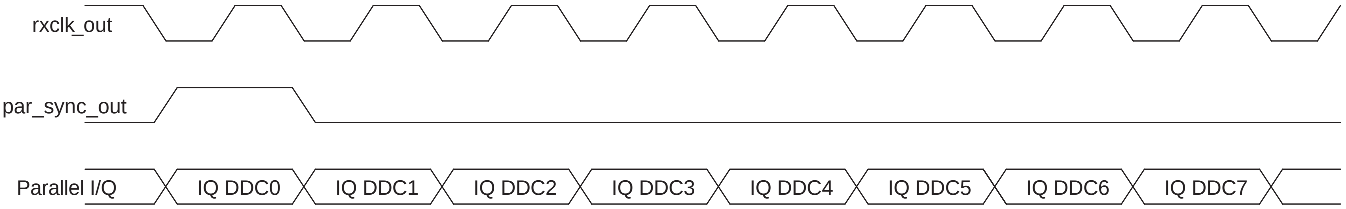


Figure 2-33. Parallel Output Interface Timing Diagram

The DDC channel serial interface synchronization source selections should all be programmed to the same value when using this parallel output interface (each DDC channel ssel_serial(2:0) in the SYNC_0 register should be programmed to the same rxsync_A/B/C/D value).

Decimation by 2 in the output interface can be achieved by setting the frame strobe interval and clock divider to 1/2 the PFIR output rate. The parallel interface samples the PFIR outputs each time the transfer interval defined by these two settings has completed.

PROGRAMMING	
VARIABLE	**DESCRIPTION**
par_recv_fsinvl(6:0)	rx_sync_out (frame strobe) sync interval. 0 is 1 rxclk cycle and 127 is 128 rxclk cycles.
par_recv_clkdiv(6:0)	rxclk_out cycles per IQ channel sample; 1 is full rate, 2 is rxclk/2, etc.
par_recv_chan(3:0)	Number channels to be output. 0 is 1 channel, and 15 is 16 channels.
par_recv_sync_del(6:0)	Delays the DDC0 pser sync source to establish the timing of IQ DDC0. Increasing the value delays the par_sync_out location.
par_recv_syncout_del(3:0)	Delays the rx_sync_out position with respect to IQ DDC0. Setting to 0 moves the rx_sync_out pulse one rxclk_out cycle before the IQ DDC0 word, setting to 1 places it as shown above, lined up with IQ DDC0, etc.
par_recv_rxclk_pol	rxclk_out polarity. Outputs data on falling edges when cleared, rising edges when set.
par_recv_sync_pol	Parallel interface par_sync_out polarity. 0 is active low, 1 for active high
par_recv_ena	Parallel TCI110 style interface enabled when set, serial interface enabled when cleared.
ssel_serial(2:0)	DDC channel serial interface sync source selection. *All DDCs should be programmed to the same sync source when using this parallel output interface.*
gain_mon	When set, the parallel output data includes 8b I at I(15:8), 8b Q at Q(15:8), 14b AGC gain at I(7:0) and Q(7:2) and 2b AGC state at Q(1:0).
tristate(6:3)	3-state controls for the rx_sync_out_X and rxout_X_X pins. Pins are in 3-state when the 3-state register bits are set.

2.2.12 DDC Checksum Generator

The checksum generator is used in conjunction with the input test signal generator to implement a self test capability.

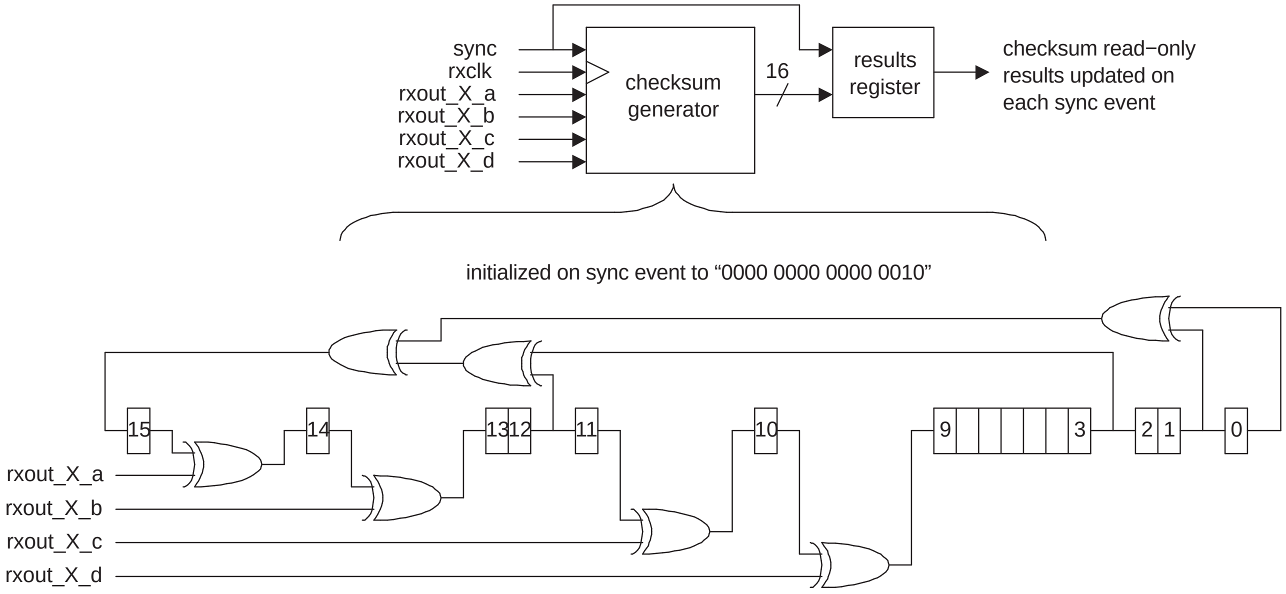


Figure 2-34. DDC Checksum Generator Block Diagram

The sync for the checksum generator is internally connected to the ddc_counter output.

PROGRAMMING	
VARIABLE	**DESCRIPTION**
ddc_chk_sum(15:0)	Read only DDC channel checksum results

3 GC5018 GENERAL CONTROL

The GC5018 is configured over a bi-directional 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 6 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 registers. This arrangement permits accessing a large number of control registers using relatively few address lines.

Global registers (Address 32 through Address 63) are used to read/write GC5018 parameters that are global in nature and can benefit from single read/write operations. Examples include chip status, reset, sync options, checksum ramp parameters, interrupt sources, interrupt masks, 3-state controls and the page register.

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.

The global 3-state register can be used to 3-state the output drivers on the GC5018, and also includes the capability of disabling the chip’s internal rxclk.

PROGRAMMING	
VARIABLE	DESCRIPTION
rxclk_ena	Enables the internal rxclk when set. When cleared, the GC5018 will ignore the rxclk input signal and hold the internal clock low.
3-state(10:0)	Various output pins are forced into tristate mode when these bits are asserted. See the GBL_3-STATE register description for pin groups to bit assignments.
arst_func	When asserted, the internal datapath is held reset. The control register programming is not affected.

3.1 Microprocessor Interface Control Data, Address, and Strobes

The microprocessor control bus consists of 16 bi-directional control data lines **d[15:0]**, 6 address lines **a[5:0]**, a read enable line **rd_n**, a write enable line **wr_n**, and a chip enable line **ce_n**. These lines usually interface to a microprocessor or DSP chip and is intended to look like a block of memory.

The interface can be operated in a 3 pin control mode (using rd_n, wr_n and ce_n) or 2 pin control mode (using wr_n and ce_n with rd_n always low).

3.1.1 MPU Timing Diagrams

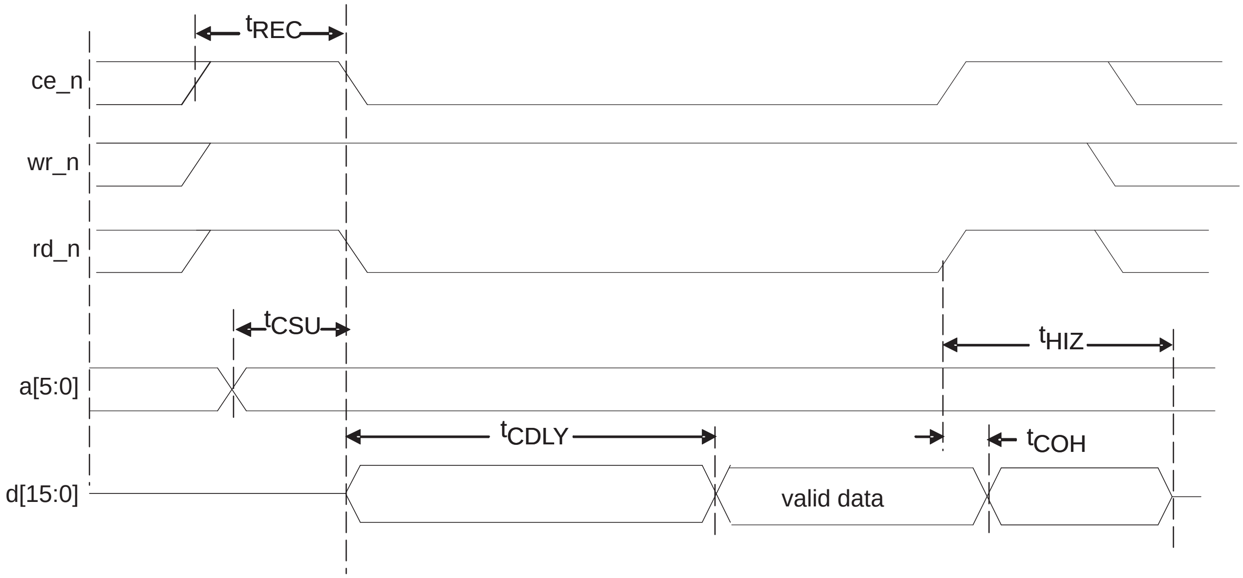


Figure 3-1. Read Operation – 3 pin control mode

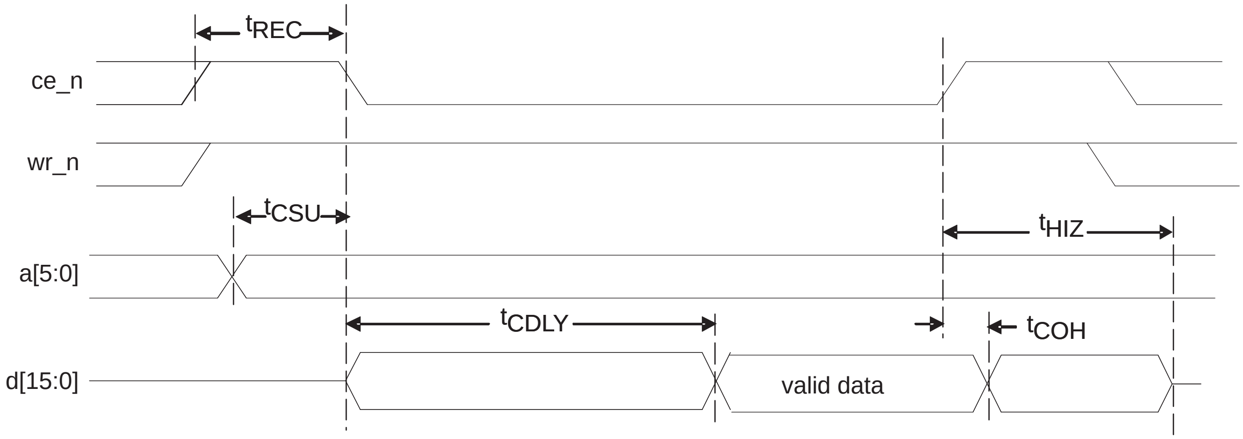


Figure 3-2. Read Operation – 2 pin control mode (rd_n tied low)

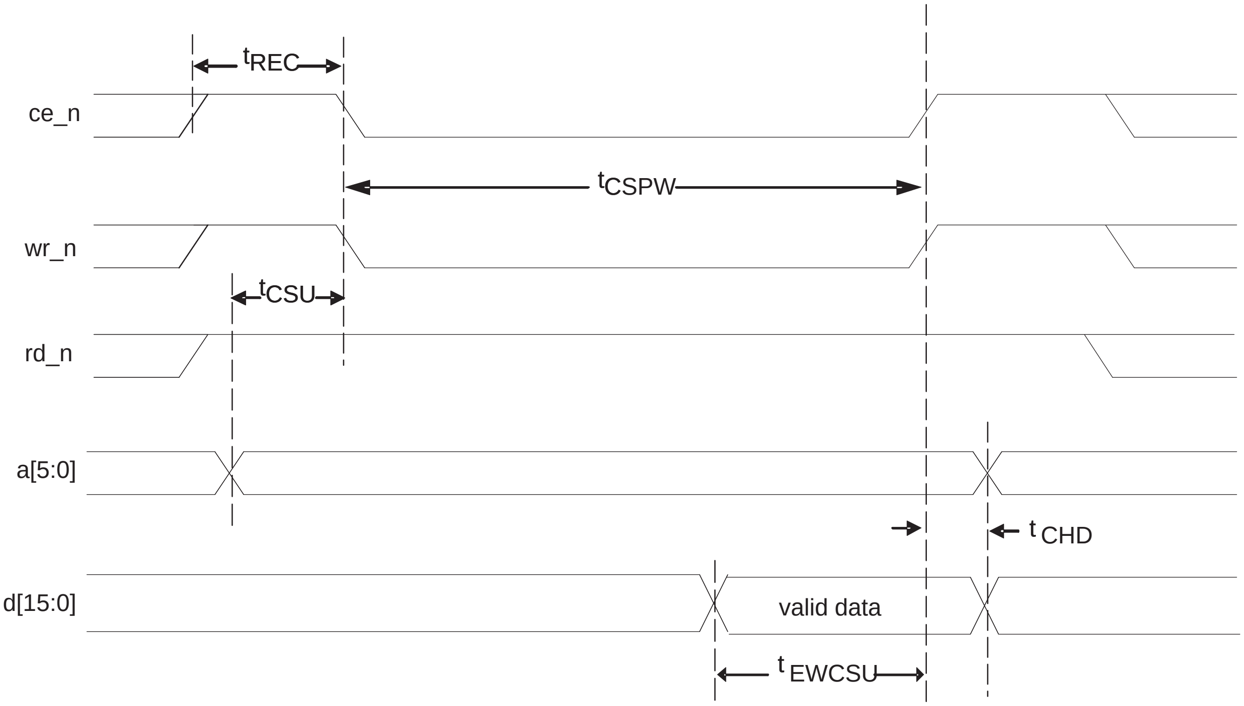


Figure 3-3. Write Operation – 3 pin control mode

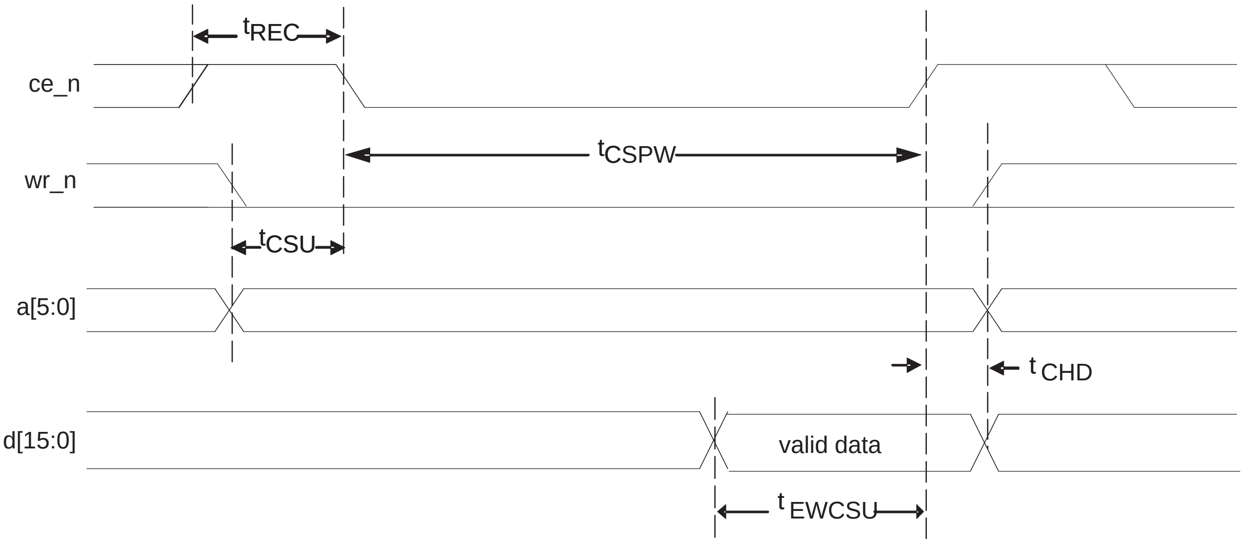


Figure 3-4. Write Operation – 2 pin control mode (rd_n tied low)

3.2 Synchronization Signals

Various function blocks within the GC5018 need to be synchronized in order to realize predictable results. The GC5018 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 four sync input pins, rxsync_a, rxsync_b, rxsync_c and rxsync_d are qualified on the rxclk rising clock edge.

Table 3-1 shows the different sync modes available.

Table 3-1. Different Sync Modes Available

SYNC SELECT CODE	RECEIVE SYNC SOURCE
000	rxsync_a
001	rxsync_b
010	rxsync_c
011	rxsync_d
100	ddc sync counter terminal count
101	ddc sync triggered by s/w oneshot (register bit)
110	0 (always off)
111	1 (always on)

Table 3-2 and Table 3-3 summarizes the blocks which have functions that can be synchronized using the above eight sync source options.

Table 3-2. Receive Common Syncs

Sync Name	Purpose
sync_ddc_counter	Initializes the receive sync counter
sync_ddc	Initializes the receive ADC interface and clock generation circuits
sync_rxsync_out	selects sync signal to be output on the rx_sync_out pin.
sync_adc_fifo	Initializes the input and output pointers in the ADC fifo circuits.
sync_tst_decim	Initializes the testbus decimation counter.
sync_recv_pmeterX	Initializes the rxin power meters. {X = 0,1,2 or 3}
sync_ragc_interval_X	Initializes the rxin receive AGC timers. {X = 0,1,2 or 3}
sync_ragc_freeze_X	rxin receive AGC freeze mode control. {X = 0,1,2 or 3}
sync_ragc_clear_X	Initializes the receive AGC error accumulator. {X = 0,1,2 or 3}

Table 3-3. DDC Channel Syncs

Sync Name	Purpose
sync_ddc_tadj	Selects zero stuff moment in the tadj fine adjustment section.
sync_ddc_tadj_reg	Updates the tadj output pointer register delay in the tadj coarse adjustment section.
sync_ddc_nco	Resets the NCO accumulator.
sync_ddc_freq	Updates the NCO freq registers.
sync_ddc_phase	Updates the NCO phase register.
sync_ddc_dither	Initializes the NCO dither circuits.
sync_ddc_cic	Selects the CIC decimation moment.
sync_ddc_pmeter	Initializes the receive channel power meters.
sync_ddc_gain	Updates the DDC channel AGC gain registers
sync_ddc_agc	Initializes the AGC accumulator.
sync_ddc_agc_freeze	AGC freeze mode control.
sync_ddc_serial	Initializes the receive serial interface.

A 32-bit general purpose timer is included in the synchronization function. The timer loads the user programmed terminal count on a sync event, and counts down to zero using rxclk. The width of the terminal count pulse can also be programmed up to rxclk cycles. The timers output can be used as a sync source for any other circuits requiring a sync if desired, and can also be routed to the rx_sync_out pin.

PROGRAMMING	
VARIABLE	DESCRIPTION
ddc_counter(31:0)	32-bit programmable terminal count
ddc_counter_width(7:0)	8-bit programmable terminal count pulse width
ssel_ddc_counter(2:0)	Sync source selection for the ddc counter

PROGRAMMING	
VARIABLE	DESCRIPTION
ssel_rxsync_out(2:0)	Sync source selection for the rx_sync_out pin
tri-state(0)	When set, the interrupt and rx_sync_out pins are tri-stated.
rx_oneshot	Register bit used to generate the S/W oneshot signal for sync. This bit must be programmed from cleared to set in order to generate a rising edge sync signal.

3.3 Interrupt Handling

When a GC5018 block sets an interrupt, the interrupt pin will go active if the interrupt source is not masked. The microprocessor should then read the interrupt register to determine the source of the interrupt. The microprocessor will then have to write the interrupt register to clear the interrupt pin and the interrupt source. The interrupt register and interrupt mask are located in the global registers section of the control registers.

The GC5018 has 16 interrupt sources; power meters in each of the eight DDC blocks, power meters in the four receive input interface, and four rxin_X_ovr (adc overflow) input pins where X={a,b,c,d}.

PROGRAMMING	
VARIABLE	DESCRIPTION
pmeterX_im(7:0)	Channel pmeter interrupt mask bits. Interrupt source is masked when set.
recv_pmeterX_im(3:0)	Receive input power meter interrupt masks.
rxin_X_ovr_im	ADC overflow input pin interrupt masks.
pmeterX(7:0)	Channel pmeter interrupt status.
recv_pmeterX(3:0)	Receive input power meter interrupt status.
rxin_X_ovr	ADC overflow input pin interrupt status.
intr_clr	When asserted, holds all interrupt status bits cleared. The interrupt pin will be inactive (always low) when this bit is set. Intended for lab/debug use only
tri-state(0)	When set, the interrupt and rx_sync_out pins are tri-stated.

3.4 GC5018 Programming

The GC5018 includes over 3000 internal configuration registers and therefore implements a paged addressing scheme. The register map includes a *global control variables* register address space that is accessed directly when the a5 signal is high. This *global control variables* address space includes the page register. All other registers are addressed using a combination of an address comprised of the internal page register contents and the 6-bit external address; a5, a4, a3, a2, a1 and a0.

The page register is accessed when the 6-bit address a5:a0 is 0x21 (or binary “100001”).

Page Register Contents in Hex	Address Pin a5	Registers Addressed With 5 Bit Address Space, Pins (a4:a0)
don't care	1	Global Control Variables 0x00 through 0x1F
0x0000	0	DDC0 PFIR taps 0 through 31 coefficient lsbs (1:0)
0x0020	0	DDC0 PFIR taps 32 through 63 coefficient lsbs (1:0)
0x0040	0	DDC0 PFIR taps 0 through 31 coefficient msbs (17:2)
0x0060	0	DDC0 PFIR taps 32 through 63 coefficient msbs (17:2)
0x0080	0	DDC0 CFIR taps 0 through 31 coefficient lsbs (1:0)
0x00A0	0	DDC0 CFIR taps 32 through 63 coefficient lsbs (1:0)
0x00C0	0	DDC0 CFIR taps 0 through 31 coefficient msbs (17:2)
0x00E0	0	DDC0 CFIR taps 32 through 63 coefficient msbs (17:2)
0x0100	0	DDC0 Control Registers 0x00 through 0x1F
0x0120	0	DDC0 Control Registers 0x20 through 0x3F
0x0200	0	DDC1 PFIR taps 0 through 31 coefficient lsbs (1:0)
0x0220	0	DDC1 PFIR taps 32 through 63 coefficient lsbs (1:0)

Page Register Contents in Hex	Address Pin a5	Registers Addressed With 5 Bit Address Space, Pins (a4:a0)
0x0240	0	DDC1 PFIR taps 0 through 31 coefficient msbs (17:2)
0x0260	0	DDC1 PFIR taps 32 through 63 coefficient msbs (17:2)
0x0280	0	DDC1 CFIR taps 0 through 31 coefficient lsbs (1:0)
0x02A0	0	DDC1 CFIR taps 32 through 63 coefficient lsbs (1:0)
0x02C0	0	DDC1 CFIR taps 0 through 31 coefficient msbs (17:2)
0x02E0	0	DDC1 CFIR taps 32 through 63 coefficient msbs (17:2)
0x0300	0	DDC1 Control Registers 0x00 through 0x1F
0x0320	0	DDC1 Control Registers 0x20 through 0x3F
0x0400	0	DDC2 PFIR taps 0 through 31 coefficient lsbs (1:0)
0x0420	0	DDC2 PFIR taps 32 through 63 coefficient lsbs (1:0)
0x0440	0	DDC2 PFIR taps 0 through 31 coefficient msbs (17:2)
0x0460	0	DDC2 PFIR taps 32 through 63 coefficient msbs (17:2)
0x0480	0	DDC2 CFIR taps 0 through 31 coefficient lsbs (1:0)
0x04A0	0	DDC2 CFIR taps 32 through 63 coefficient lsbs (1:0)
0x04C0	0	DDC2 CFIR taps 0 through 31 coefficient msbs (17:2)
0x04E0	0	DDC2 CFIR taps 32 through 63 coefficient msbs (17:2)
0x0500	0	DDC2 Control Registers 0x00 through 0x1F
0x0520	0	DDC2 Control Registers 0x20 through 0x3F
0x0600	0	DDC3 PFIR taps 0 through 31 coefficient lsbs (1:0)
0x0620	0	DDC3 PFIR taps 32 through 63 coefficient lsbs (1:0)
0x0640	0	DDC3 PFIR taps 0 through 31 coefficient msbs (17:2)
0x0660	0	DDC3 PFIR taps 32 through 63 coefficient msbs (17:2)
0x0680	0	DDC3 CFIR taps 0 through 31 coefficient lsbs (1:0)
0x06A0	0	DDC3 CFIR taps 32 through 63 coefficient lsbs (1:0)
0x06C0	0	DDC3 CFIR taps 0 through 31 coefficient msbs (17:2)
0x06E0	0	DDC3 CFIR taps 32 through 63 coefficient msbs (17:2)
0x0700	0	DDC3 Control Registers 0x00 through 0x1F
0x0720	0	DDC3 Control Registers 0x20 through 0x3F
0x0800	0	DDC4 PFIR taps 0 through 31 coefficient lsbs (1:0)
0x0820	0	DDC4 PFIR taps 32 through 63 coefficient lsbs (1:0)
0x0840	0	DDC4 PFIR taps 0 through 31 coefficient msbs (17:2)
0x0860	0	DDC4 PFIR taps 32 through 63 coefficient msbs (17:2)
0x0880	0	DDC4 CFIR taps 0 through 31 coefficient lsbs (1:0)
0x08A0	0	DDC4 CFIR taps 32 through 63 coefficient lsbs (1:0)
0x08C0	0	DDC4 CFIR taps 0 through 31 coefficient msbs (17:2)
0x08E0	0	DDC4 CFIR taps 32 through 63 coefficient msbs (17:2)
0x0900	0	DDC4 Control Registers 0x00 through 0x1F
0x0920	0	DDC4 Control Registers 0x20 through 0x3F
0x0A00	0	DDC5 PFIR taps 0 through 31 coefficient lsbs (1:0)
0x0A20	0	DDC5 PFIR taps 32 through 63 coefficient lsbs (1:0)
0x0A40	0	DDC5 PFIR taps 0 through 31 coefficient msbs (17:2)
0x0A60	0	DDC5 PFIR taps 32 through 63 coefficient msbs (17:2)
0x0A80	0	DDC5 CFIR taps 0 through 31 coefficient lsbs (1:0)
0x0AA0	0	DDC5 CFIR taps 32 through 63 coefficient lsbs (1:0)
0x0AC0	0	DDC5 CFIR taps 0 through 31 coefficient msbs (17:2)
0x0AE0	0	DDC5 CFIR taps 32 through 63 coefficient msbs (17:2)
0x0B00	0	DDC5 Control Registers 0x00 through 0x1F

Page Register Contents in Hex	Address Pin a5	Registers Addressed With 5 Bit Address Space, Pins (a4:a0)
0x0B20	0	DDC5 Control Registers 0x20 through 0x3F
0x0C00	0	DDC6 PFIR taps 0 through 31 coefficient lsbs (1:0)
0x0C20	0	DDC6 PFIR taps 32 through 63 coefficient lsbs (1:0)
0x0C40	0	DDC6 PFIR taps 0 through 31 coefficient msbs (17:2)
0x0C60	0	DDC6 PFIR taps 32 through 63 coefficient msbs (17:2)
0x0C80	0	DDC6 CFIR taps 0 through 31 coefficient lsbs (1:0)
0x0CA0	0	DDC6 CFIR taps 32 through 63 coefficient lsbs (1:0)
0x0CC0	0	DDC6 CFIR taps 0 through 31 coefficient msbs (17:2)
0x0CE0	0	DDC6 CFIR taps 32 through 63 coefficient msbs (17:2)
0x0D00	0	DDC6 Control Registers 0x00 through 0x1F
0x0D20	0	DDC6 Control Registers 0x20 through 0x3F
0x0E00	0	DDC7 PFIR taps 0 through 31 coefficient lsbs (1:0)
0x0E20	0	DDC7 PFIR taps 32 through 63 coefficient lsbs (1:0)
0x0E40	0	DDC7 PFIR taps 0 through 31 coefficient msbs (17:2)
0x0E60	0	DDC7 PFIR taps 32 through 63 coefficient msbs (17:2)
0x0E80	0	DDC7 CFIR taps 0 through 31 coefficient lsbs (1:0)
0x0EA0	0	DDC7 CFIR taps 32 through 63 coefficient lsbs (1:0)
0x0EC0	0	DDC7 CFIR taps 0 through 31 coefficient msbs (17:2)
0x0EE0	0	DDC7 CFIR taps 32 through 63 coefficient msbs (17:2)
0x0F00	0	DDC7 Control Registers 0x00 through 0x1F
0x0F20	0	DDC7 Control Registers 0x20 through 0x3F
0x1000	0	Receive Input AGC0 Error RAM addresses 0 through 31
0x1020	0	Receive Input AGC0 Error RAM addresses 32 through 63
0x1040	0	Receive Input AGC0 DVGA RAM addresses 0 through 31
0x1080	0	Receive Input AGC0 Gain RAM addresses 0 through 31
0x10A0	0	Receive Input AGC0 Gain RAM addresses 32 through 63
0x1100	0	Receive Input AGC1 Error RAM addresses 0 through 31
0x1120	0	Receive Input AGC1 Error RAM addresses 32 through 63
0x1140	0	Receive Input AGC1 DVGA RAM addresses 0 through 31
0x1180	0	Receive Input AGC1 Gain RAM addresses 0 through 31
0x11A0	0	Receive Input AGC1 Gain RAM addresses 32 through 63
0x1400	0	Receive Input AGC2 Error RAM addresses 0 through 31
0x1420	0	Receive Input AGC2 Error RAM addresses 32 through 63
0x1440	0	Receive Input AGC2 DVGA RAM addresses 0 through 31
0x1480	0	Receive Input AGC2 Gain RAM addresses 0 through 31
0x14A0	0	Receive Input AGC2 Gain RAM addresses 32 through 63
0x1500	0	Receive Input AGC3 Error RAM addresses 0 through 31
0x1520	0	Receive Input AGC3 Error RAM addresses 32 through 63
0x1540	0	Receive Input AGC3 DVGA RAM addresses 0 through 31
0x1580	0	Receive Input AGC3 Gain RAM addresses 0 through 31
0x15A0	0	Receive Input AGC3 Gain RAM addresses 32 through 63
0x1800	0	Receive Input Control Registers 0x00 through 0x1F
0x1820	0	Receive Input Control Registers 0x20 through 0x3F
0x1840	0	Receive Input AGC Control Registers 0x00 through 0x1F
0x1860	0	Receive Input AGC Control Registers 0x20 through 0x3F

3.4.1 Control Register Index

Table 3-4. Control Register Index

REGISTER NAME	SECTION
AGC_AMAX	Section 3.4.5.23
AGC_AMIN	Section 3.4.5.24
AGC_CONFIG1	Section 3.4.5.17
AGC_CONFIG2	Section 3.4.5.18
AGC_CONFIG3	Section 3.4.5.19
AGC_GAINA	Section 3.4.5.21
AGC_GAINB	Section 3.4.5.22
AGC_GAINMSB	Section 3.4.5.20
CIC_MODE1	Section 3.4.5.5
CIC_MODE2	Section 3.4.5.6
CONFIG	Section 3.4.2.3
CONFIG1	Section 3.4.5.15
CONFIG2	Section 3.4.5.16
DDC_CHK_SUM	Section 3.4.5.31
DDCCONFIG1	Section 3.4.5.27
FIR_GAIN	Section 3.4.5.2
FIR_MODE	Section 3.4.5.1
GBL_IMASK0	Section 3.4.2.8
GBL_INTERRUPT0	Section 3.4.2.9
GBL_ONESHOT	Section 3.4.2.7
GBL_PAR_CONFIG0	Section 3.4.2.4
GBL_PAR_CONFIG1	Section 3.4.2.5
GBL_TRISTATE	Section 3.4.2.6
NZ_PWR_MASK	Section 3.4.3.7
PAGE	Section 3.4.2.2
PHASE_OFFSETA	Section 3.4.5.13
PHASE_OFFSETB	Section 3.4.5.14
PHASEADD0A	Section 3.4.5.9
PHASEADD0B	Section 3.4.5.11
PHASEADD1A	Section 3.4.5.10
PHASEADD1B	Section 3.4.5.12
PMETER_RESULT_A_LSB	Section 3.4.5.32
PMETER_RESULT_A_MID	Section 3.4.5.33
PMETER_RESULT_A_MSB	Section 3.4.5.34
PMETER_RESULT_B_LSB	Section 3.4.5.35
PMETER_RESULT_B_MID	Section 3.4.5.36
PMETER_RESULT_B_MSB	Section 3.4.5.37
PMETER_RESULT_AB_UMSB	Section 3.4.5.38
PSER_CONFIG1	Section 3.4.5.25
PSER_CONFIG2	Section 3.4.5.26
RAGC_CONFIG0	Section 3.4.4.1
RAGC_CONFIG1	Section 3.4.4.2
RAGC_CONFIG2	Section 3.4.4.3
RAGC_CONFIG3	Section 3.4.4.4
RAGC0_ACCUM_LSB	Section 3.4.4.57
RAGC0_ACCUM_MSB	Section 3.4.4.58
RAGC0_CLIP_ERROR	Section 3.4.4.17
RAGC0_CLIP_HITHRESH	Section 3.4.4.12
RAGC0_CLIP_HITIMER	Section 3.4.4.14
RAGC0_CLIP_LOTHRESH	Section 3.4.4.13
RAGC0_CLIP_LOTIMER	Section 3.4.4.15
RAGC0_CLIP_SAMPLES	Section 3.4.4.16
RAGC0_CONFIG0	Section 3.4.4.7
RAGC0_CONFIG1	Section 3.4.4.8
RAGC0_INTEGINVL_LSB	Section 3.4.4.5
RAGC0_INTEGINVL_MSB	Section 3.4.4.6
RAGC0_SD_SAMPLES	Section 3.4.4.11
RAGC0_SD_THRESH	Section 3.4.4.9
RAGC0_SD_TIMER	Section 3.4.4.10
RAGC1_ACCUM_LSB	Section 3.4.4.59
RAGC1_ACCUM_MSB	Section 3.4.4.60
RAGC1_CLIP_ERROR	Section 3.4.4.30
RAGC1_CLIP_HITHRESH	Section 3.4.4.25
RAGC1_CLIP_HITIMER	Section 3.4.4.27
RAGC1_CLIP_LOTHRESH	Section 3.4.4.26
RAGC1_CLIP_LOTIMER	Section 3.4.4.28
RAGC1_CLIP_SAMPLES	Section 3.4.4.29
RAGC1_CONFIG0	Section 3.4.4.20
RAGC1_CONFIG1	Section 3.4.4.21
RAGC1_INTEGINVL_LSB	Section 3.4.4.18
RAGC1_INTEGINVL_MSB	Section 3.4.4.19
RAGC1_SD_SAMPLES	Section 3.4.4.24
RAGC1_SD_THRESH	Section 3.4.4.22
RAGC1_SD_TIMER	Section 3.4.4.23
RAGC2_ACCUM_LSB	Section 3.4.4.61
RAGC2_ACCUM_MSB	Section 3.4.4.62
RAGC2_CLIP_ERROR	Section 3.4.4.43
RAGC2_CLIP_HITHRESH	Section 3.4.4.38
RAGC2_CLIP_HITIMER	Section 3.4.4.40
RAGC2_CLIP_LOTHRESH	Section 3.4.4.39
RAGC2_CLIP_LOTIMER	Section 3.4.4.41
RAGC2_CLIP_SAMPLES	Section 3.4.4.42
RAGC2_CONFIG0	Section 3.4.4.33
RAGC2_CONFIG1	Section 3.4.4.34
RAGC2_INTEGINVL_LSB	Section 3.4.4.31
RAGC2_INTEGINVL_MSB	Section 3.4.4.32
RAGC2_SD_SAMPLES	Section 3.4.4.37
RAGC2_SD_THRESH	Section 3.4.4.35
RAGC2_SD_TIMER	Section 3.4.4.36
RAGC3_ACCUM_LSB	Section 3.4.4.63

Table 3-4. Control Register Index (continued)

REGISTER NAME	SECTION
RAGC3_ACCUM_MSB	Section 3.4.4.64
RAGC3_CLIP_ERROR	Section 3.4.4.56
RAGC3_CLIP_HITHRESH	Section 3.4.4.51
RAGC3_CLIP_HITIMER	Section 3.4.4.53
RAGC3_CLIP_LOTHRESH	Section 3.4.4.52
RAGC3_CLIP_LOTIMER	Section 3.4.4.54
RAGC3_CLIP_SAMPLES	Section 3.4.4.55
RAGC3_CONFIG0	Section 3.4.4.46
RAGC3_CONFIG1	Section 3.4.4.47
RAGC3_INTEGINVL_LSB	Section 3.4.4.44
RAGC3_INTEGINVL_MSB	Section 3.4.4.45
RAGC3_SD_SAMPLES	Section 3.4.4.50
RAGC3_SD_THRESH	Section 3.4.4.48
RAGC3_SD_TIMER	Section 3.4.4.49
RECV_CONFIG0	Section 3.4.3.5
RECV_CONFIG1	Section 3.4.3.6
RECV_PMETER_SYNC	Section 3.4.3.8
RECV_PMETER0_CONFIG	Section 3.4.3.12
RECV_PMETER0_LMSB	Section 3.4.3.28
RECV_PMETER0_LSB	Section 3.4.3.26
RECV_PMETER0_MID	Section 3.4.3.27
RECV_PMETER0_SQR_SUM_LSB	Section 3.4.3.9
RECV_PMETER0_STRT_INTVL_LSB	Section 3.4.3.10
RECV_PMETER0_SYNC_DLY	Section 3.4.3.11
RECV_PMETER0_UMSB	Section 3.4.3.29
RECV_PMETER1_CONFIG	Section 3.4.3.16
RECV_PMETER1_LMSB	Section 3.4.3.32
RECV_PMETER1_LSB	Section 3.4.3.30
RECV_PMETER1_MID	Section 3.4.3.31
RECV_PMETER1_SQR_SUM_LSB	Section 3.4.3.13
RECV_PMETER1_STRT_INTVL_LSB	Section 3.4.3.14
RECV_PMETER1_SYNC_DLY	Section 3.4.3.15

Table 3-4. Control Register Index (continued)

REGISTER NAME	SECTION
RECV_PMETER1_UMSB	Section 3.4.3.33
RECV_PMETER2_CONFIG	Section 3.4.3.20
RECV_PMETER2_LMSB	Section 3.4.3.36
RECV_PMETER2_LSB	Section 3.4.3.34
RECV_PMETER2_MID	Section 3.4.3.35
RECV_PMETER2_SQR_SUM_LSB	Section 3.4.3.17
RECV_PMETER2_STRT_INTVL_LSB	Section 3.4.3.18
RECV_PMETER2_SYNC_DLY	Section 3.4.3.19
RECV_PMETER2_UMSB	Section 3.4.3.37
RECV_PMETER3_CONFIG	Section 3.4.3.24
RECV_PMETER3_LMSB	Section 3.4.3.40
RECV_PMETER3_LSB	Section 3.4.3.38
RECV_PMETER3_MID	Section 3.4.3.39
RECV_PMETER3_SQR_SUM_LSB	Section 3.4.3.21
RECV_PMETER3_STRT_INTVL_LSB	Section 3.4.3.22
RECV_PMETER3_SYNC_DLY	Section 3.4.3.23
RECV_PMETER3_UMSB	Section 3.4.3.41
RECV_SLF_TST_VALUE	Section 3.4.3.25
SQR_SUM	Section 3.4.5.3
SSEL_DDC_CNTR	Section 3.4.3.3
SSEL_RX_0	Section 3.4.3.4
STRT_INTRVL	Section 3.4.5.4
SYNC_0	Section 3.4.5.28
SYNC_1	Section 3.4.5.29
SYNC_2	Section 3.4.5.30
SYNC_DDC_CNTR_LSB	Section 3.4.3.1
SYNC_DDC_CNTR_MSB	Section 3.4.3.2
TADJC	Section 3.4.5.7
TADJF	Section 3.4.5.8
VER	Section 3.4.2.1

3.4.2 Global Control Variables

These registers are accessed directly without page address extension; when pin a5 is high during a read or write access, this block of 32 registers are accessed.

3.4.2.1 VER Register

Register name: VER **Address: 0x20** **READ_ONLY**

BIT 15 BIT 8

unused	unused	unused	unused	unused	unused	unused	unused
0	0	0	0	0	0	0	0

BIT 7 BIT 0

unused	unused	unused	unused	VER3	VER2	VER1	VER0
0	0	0	0	0	0	*	*

VER(3:0): A hardwired read only register that returns the version of the chip.
* valid version codes are "0001" and "0010"

3.4.2.2 PAGE Register

Register name: PAGE **Address: 0x21**

BIT 15							BIT 8
unused	unused	unused	W(2:0)			X	Y(2)
0	0	0	0	0	0	0	0

BIT 7							BIT 0
Y(1)	Y(0)	Zp	unused	unused	unused	unused	unused
0	0	0	0	0	0	0	0

W(2:0) : Selects which dual DDC block to address.

X : The DDC modules are configured as dual DDCs; an even numbered DDC and odd numbered DDC are contained in each dual DDC module, the X bit selects which DDC gets address. (DDC0/2/4/6=0, DDC1/3/5/7=1)

W(2:0)	X bit	Selected Block
000	0	DDC0
000	1	DDC1
001	0	DDC2
001	1	DDC3
010	0	DDC4
010	1	DDC5
011	0	DDC6
011	1	DDC7
100	0	Receive AGC0/1 RAMs
101	0	Receive AGC2/3 RAMs
110	0	Receive Input Interface

Y(2:0) : Within each major block, there are up to 8 different Zones that can be addressed using the Y bits.

Y(2:0)	DDC Zone	Receive Input Interface Zone	Receive AGC RAMs Zone
000	PFIR coeffient lower 2 bits	CHIPS control registers	RAGC0/2 ERRMAP
001	PFIR coeffient upper 16 bits	RAGC control registers	RAGC0/2 DVGAMAP
010	CFIR coeffient lower 2 bits	Not assigned	RAGC0/2 GAINMAP
011	CFIR coeffient upper 16 bits	Not assigned	Not assigned
100	Control registers	Not assigned	RAGC1/3 ERRMAP
101	Not assigned	Not assigned	RAGC1/3 DVGAMAP
110	Not assigned	Not assigned	RAGC1/3 GAINMAP
111	Not assigned	Not assigned	Not assigned

Zp : The Zp bit is the MSB of the address word sent to the registers and rams. This bit can be thought of as an upper/lower selector of the 64 word addressing.

3.4.2.3 CONFIG Register

Register name: CONFIG **Address: 0x22**

BIT 15 BIT 8

slf_ tst_ena	rduz_sens_ena	arst_ func	tst_rate_sel(4:0)				
0	0	0	0	0	0	0	0

BIT 7 BIT 0

par_recv_ena	gbl_ ddc_write	intr_ clr	tst_select(3:0)				tst_ on
0	0	0	1	1	1	0	0

slf_tst_ena : Turns on the checksum LFSR for the receivers. They are located in the RECEIVE INPUT INTERFACE and DDC blocks

rduz_sens_ena : When enabled, adds noise to the LSB’s of the ADC inputs.

arst_func : When asserted, resets the functional portion of the circuits. The MPU registers do not get reset and retain their programmed value

tst_rate_sel(4:0) : Sets the rate of the output test data and clock. The length of the clock cycle is the value in tst_rate_sel+1 multiplied by the RXCLK period.

par_recv_ena : When asserted, the rxout_*_* serial pins join to form a 32 bit parallel output using 32 pins as a data bus, one pin as a output clock and one pin as a sync. This is used to connect to the TCI110 Chip rate processor from TI.

gbl_ddc_write : When asserted, the mpu writes are global. This means that DDC0/2/4/6 or DDC1/3/5/7 can be programmed simultaneously with the same values. This is an effort to reduce the amount of time spent programming the device. A common setup can be used to program the DDC0/2/4/6, then all the DDC1/3/5/7. Afterwards, just individual writes to the registers which differ between DDCs can be done. To use this feature, this bit must be asserted and the DDC0/1 must be addressed. Any other DDC address will not work.

intr_clr : When asserted, this bit forces all interrupts to be cleared. To allow the interrupts to be set again, this bit must be programmed to zero. This does not stop blocks from generating interrupts, but rather just keeps the interrupts from being reported.

tst_select(3:0) : This selects which block the test output comes from:

tst_select(3:0)	Test Data Sent to Output
0000	DDC 0
0001	DDC 1
0010	DDC 2
0011	DDC 3
0100	DDC 4
0101	DDC 5
0110	DDC 6
0111	DDC 7
1000	rxin_a and rxin_b FIFO outputs
others	none selected

tst_on : When asserted, the testbus is active. The ADC input ports rxin_c(15:0), rxin_d(15:0), dvga_c(5:0) and dvga_d(5:0) become the testbus output ports. When this bit is set, the rxin_c(15:0) and rxin_d(15:0) ports become chip outputs. The dvga_c(5:0) and dvga_d(5:0) ports are enabled separately using the GBL_TRISTATE register

3.4.2.4 GBL_PAR_CONFIG0 Register

Register name: GBL_PAR_CONFIG0 **Address: 0x23**

BIT 15							BIT 8
par_recv_sync_del(6:0)							tst_clk_pol
0	0	0	0	0	0	0	0

BIT 7							BIT 0
par_recv_clkdiv(6:0)							par_recv_rxclk_pol
0	0	0	0	0	0	0	0

par_recv_sync_del(6:0) : Delays the sync source from the DDC0 AGC output by (par_recv_sync_del+1) rxclk cycles.

tst_clk_pol : Selects the polarity of the test clock output at dvga_c(1) when the test bus is enabled; 0 for rising edge in the center of valid data, 1 for falling edge in the center of valid data. *No effect when tst_rate_sel is “00000”.*

par_recv_clkdiv(6:0) : Selects the parallel interface output clock rate.

par_recv_rxclk_pol : Selects the polarity of the rxclk_out clock output; 0 for rising edge in the center of valid data, 1 for falling edge in the center of valid data.

3.4.2.5 GBL_PAR_CONFIG1 Register

Register name: GBL_PAR_CONFIG1 **Address: 0x24**

BIT 15							BIT 8
par_recv_syncout_del(3:0)				par_recv_chan(3:0)			
0	0	0	0	0	0	0	0

BIT 7							BIT 0
par_recv_fsinvl(6:0)							par_recv_sync_pol
0	0	0	0	0	0	0	0

par_recv_syncout_del(3:0) : Changes the rx_sync_out position with respect to IQ DDC0. Setting to 0 causes rx_sync_out to lead IQ DDC0 by 1 output sample, setting to 1 causes rx_sync_out to line up with IQ DDC0, setting to 2 causes rx_sync_out to trail IQ DDC0 by 1 output sample, etc.

par_recv_chan(3:0) : Selects the number of channels to be output over the parallel interface, from 1 to 16 channels.

par_recv_fsinvl(6:0) : Selects the number of rxclk cycles per parallel interface frame, from 1 to 128 cycles.

par_recv_sync_pol : Selects the polarity of the parallel interface sync pulse; 0 for active low, 1 for active high.

3.4.2.6 GBL_TRISTATE Register

Register name: GBL_TRISTATE **Address: 0x25**

BIT 15							BIT 8
rxclk_ena	unused	unused	unused	unused	tristate(10)	tristate(9)	tristate(8)
1	0	0	0	0	1	1	1

BIT 7							BIT 0
tristate(7)	tristate(6)	tristate(5)	tristate(4)	tristate(3)	tristate(2)	tristate(1)	
1	1	1	1	1	1	1	1

rxclk_ena : *Master rxclk enable.* When set, the chip's rxclk is enabled, when cleared, rxclk is disabled.

All tristates are ACTIVE LOW so a '0' turns on the output and a '1' tristates it.

tristate(10) : This bit turns on the dvga_d outputs.

tristate(9) : This bit turns on the dvga_c outputs.

tristate(8) : This bit turns on the dvga_b outputs.

tristate(7) : This bit turns on the dvga_a outputs.

tristate(6) : This bit turns on the rx_sync_out_6/7, and the rxout_6/7_a/b/c/d outputs.

tristate(5) : This bit turns on the rx_sync_out_4/5, and the rxout_4/5_a/b/c/d outputs.

tristate(4) : This bit turns on the rx_sync_out_2/3, and the rxout_2/3_a/b/c/d outputs.

tristate(3) : This bit turns on the rx_sync_out_0/1, and the rxout_0/1_a/b/c/d outputs.

tristate(2) : unused

tristate(1) : rxclk_out

tristate(0) : interrupt, and rx_sync_out.

3.4.2.7 GBL_ONESHOT Register

Register name: GBL_ONESHOT **Address: 0x26**

BIT 15							BIT 8
unused	unused	unused	unused	unused	unused	unused	unused
0	0	0	0	0	0	0	0

BIT 7							BIT 0
rx_oneshot	unused	unused	unused	unused	unused	unused	unused
0	0	0	0	0	0	0	0

rx_oneshot : When set, a one shot pulse is sent to the receive blocks for syncing. This only works if the blocks are programmed to use the oneshot as the sync source. To use the oneshot again, it must be programmed back to a '0' and then back to a '1'.

3.4.2.8 GBL_IMASK0 Register

Register name: GBL_IMASK0 **Address: 0x27**

BIT 15							BIT 8
pmeter7_im	pmeter6_im	pmeter5_im	pmeter4_im	pmeter3_im	pmeter2_im	pmeter1_im	pmeter0_im
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_ pmeter0_im	recv_ pmeter1_im	recv_ pmeter2_im	recv_ pmeter3_im	rxin_a_ ovr_im	rxin_b_ ovr_im	rxin_c_ ovr_im	rxin_d_ ovr_im
0	0	0	0	0	0	0	0

pmeterX_im : When asserted, masks the interrupt for the particular DDC pmeter, X= {0,1,2,3,4,5,6,7}.

recv_pmeterX_im : When asserted, masks the interrupt for the particular receive input pmeter, X= {0,1,2,3 }.

rxin_X_ovr_im : When asserted, masks the interrupt for the particular rxin overflow, X={a,b,c,d}.

3.4.2.9 GBL_INTERRUPT0 Register

Register name: GBL_INTERRUPT0 **Address: 0x29**

BIT 15							BIT 8
pmeter7	pmeter6	pmeter5	pmeter4	pmeter3	pmeter2	pmeter1	pmeter0
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_ pmeter0	recv_ pmeter1	recv_ pmeter2	recv_ pmeter3	rxin_a_ovr	rxin_b_ovr	rcin_c_ovr	rxin_d_ovr
0	0	0	0	0	0	0	0

pmeterX : Asserted when an interrupt has been generated by this DDC pmeterX block, X={1,2,3,4,5,6,7

recv_pmeterX : Asserted when an interrupt has been generated by this receive input pmeter, X= {0,1,2,3 }.

rxin_X_ovr : Asserted when a logic high input from the rxin_X_ovr pin occurs, X={a,b,c,d}.

3.4.3 Receive Input Interface Controls

3.4.3.1 SYNC_DDC_CNTR_LSB Register

Register name: SYNC_DDC_CNTR_LSB **Page: 0x1800** **Address: 0x00**

BIT 15							BIT 8
ddc_counter(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ddc_counter(7:0)							
0	0	0	0	0	0	0	0

3.4.3.2 SYNC_DDC_CNTR_MSB

Register name: SYNC_DDC_CNTR_MSB **Page: 0x1800** **Address: 0x01**

BIT 15							BIT 8
ddc_counter(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ddc_counter(23:16)							
0	0	0	0	0	0	0	0

ddc_counter(32:0) : 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. The value programmed into the counter is: (desired number –1). The counter increments on each RX clock rising edge.

3.4.3.3 SSEL_DDC_CNTR Register

Register name: SSEL_DDC_CNTR **Page: 0x1800** **Address: 0x02**

BIT 15							BIT 8
rxinab_mux	rxincd_mux	unused	unused	unused	ssel_ddc_counter(2:0)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ddc_counter_width(7:0)							
0	0	0	0	0	0	0	0

rxinab_mux : When asserted, the rxin_a and rxin_b inputs are internally driven by the rxin_c and rxin_d ports, respectively (Factory test use only).

rxincd_mux : When asserted, the rxin_c and rxin_d inputs are internally driven by the rxin_a and rxin_b ports, respectively (Factory test use only).

ssel_ddc_counter(2:0) : Selects the sync source for the DDC sync counter.

ddc_counter_width(7:0) : Sets the width of the counter generated sync pulse in RX clock cycles, from 1 to 256.

Sync sources are contained in this and many of the following registers. For all sync source selections:

ssel_ddc_XXXXX(2:0)	Selected Sync Source
000	rxsyncA
001	rxsyncB
010	rxsyncC
011	rxsyncD
100	DDC sync counter
101	one shot (register write triggered)
110	always 0
111	always 1

3.4.3.4 SSEL_RX_0 Register

Register name: SSEL_RX_0 **Page: 0x1800** **Address: 0x03**

BIT 15							BIT 8
unused	ssel_adc_fifo(2:0)			unused	ssel_tst_decim(2:0)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
unused	ssel_rxsync_out(2:0)			unused	ssel_ddc(2:0)		
0	0	0	0	0	0	0	0

ssel_adc_fifo(2:0) : Selects the sync source for the adc FIFO blocks. Sync reinitializes the read and write pointers of the FIFO.

ssel_tst_decim(2:0) : Selects the sync source for the test bus decimator block.

ssel_rxsync_out(2:0) : Selects the sync source for the RXSYNC_OUT pin.

ssel_ddc(2:0) : Selects the sync source for the DDC data input mux and mixer. *Controls clock generation in each DDC block (before the CIC input) which must match because the FIFO output clock is common for all DDC blocks.*

3.4.3.5 RECV_CONFIG0 Register

Register name: RECV_CONFIG0 **Page: 0x1800** **Address: 0x04**

BIT 15 BIT 8

rate_sel(1:0)		adc_ fifo_strap_ab	adc_ fifo_strap_cd	self_test_ const_ena	adc_ fifo_bypass	ragc_mpu_ram _read	tst_ decim17
0	0	0	0	0	0	0	0

BIT 7 BIT 0

tst_decim_delay(3:0)				pmeter3_iq	pmeter2_iq	pmeter1_iq	pmeter0_iq
0	0	0	0	0	0	0	0

rate_sel(1:0) : Tells the RECV_CDRV the input rate. This is the rxin_a/b/c/d input rate and the rate that the RECEIVE INPUT INTERFACE block sends data to the DDCs.

rate_sel	Input clock rate
00	rxclk
01	rxclk/2
10	rxclk/4
11	rxclk/8

adc_fifo_strap_ab : When asserted, the input pointers of the rxin_a FIFO and rxin_b FIFO are hooked together in lock step configuration. This is used for maintaining FIFO delay consistency when complex inputs are driven on rxin_a(I) and rxin_b(Q). rxin_a is the Master.

adc_fifo_strap_cd : When asserted, the input pointers of the rxin_c FIFO and rxin_d FIFO are hooked together in lock step configuration. This is used for maintaining FIFO delay consistency when complex inputs are driven on rxin_c(I) and rxin_d(Q). rxin_c is the Master.

self_test_const_ena : When asserted, (with slf_tst_ena also asserted), a constant value is output by the test and noise generator instead of the pseudo random sequence. The constant value is programmable.

adc_fifo_bypass : When asserted, the ADC FIFO circuits are bypassed. Input data is then clocked in directly using the rxclk input. *The ssel_ddc selection value will control the location of the internally generated sample clock when this bit is asserted where rate_sel is rxclk/2, rxclk/4 or rxclk/8.*

ragc_mpu_ram_read : When asserted, the RAMs in the RAGC blocks can be read. This bit should only be set when reading the RAGC map rams via the mpu interface and must be cleared for proper RAGC operation.

tst_decim17 : When set, the decimation factor of the tst_decimator block is 17X. When cleared, the decimation factor is 1X (no decimation).

tst_decim_delay(3:0) : These bits set the delay from the sync occurring until the decimator samples. In other words, the moment of the decimator is set by this delay value.

pmeter3_iq : When asserted, the pmeter3 block takes input from both rxin_c and rxin_d as a complex sample pair. When de-asserted, only input from rxin_d is used for the power measurement.

pmeter2_iq : When asserted, the pmeter2 block takes input from both rxin_c and rxin_d as a complex sample pair. When de-asserted, only input from rxin_c is used for the power measurement.

pmeter1_iq : When asserted, the pmeter1 block takes input from both rxin_a and rxin_b as a complex sample pair. When de-asserted, only input from rxin_b is used for the power measurement.

pmeter0_iq : When asserted, the pmeter0 block takes input from both rxin_a and rxin_b as a complex sample pair. When de-asserted, only input from rxin_a is used for the power measurement.

3.4.3.6 RECV_CONFIG1 Register

Register name: RECV_CONFIG1 **Page: 0x1800** **Address: 0x05**

BIT 15							BIT 8
msb_pos_d(2:0)			offset_bin_d	msb_pos_c(2:0)			offset_bin_c
0	0	0	0	0	0	0	0

BIT 7							BIT 0
msb_pos_b(2:0)			offset_bin_b	msb_pos_a(2:0)			offset_bin_a
0	0	0	0	0	0	0	0

msb_pos_X(2:0) : Places the MSB of the input word from the ADC. The value programmed into the 3 bits is the number of bit positions to the left of bit16 in the input word, that the MSB is located. For example, if a 14bit input word is driving rxin_a input and is aligned with rxin_a_0, then msb_pos_a is programmed to “010” meaning 2 bits shifted down from bit 16 is the MSB. X={a,b,c,d}.

offset_bin_X : rxin_X input data is in offset binary and not twos complement. If set, the input value will be converted to 2s complement using the MSB from the corresponding msb_pos_X value. X={a,b,c,d}

3.4.3.7 NZ_PWR_MASK Register

Register name: NZ_PWR_MASK **Page: 0x1800** **Address: 0x06**

BIT 15							BIT 8
nz_pwr_mask (15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
nz_pwr_mask (7:0)							
0	0	0	0	0	0	0	0

nz_pwr_mask(15:0) : Used with the rduz_sens_ena and selects the noise bits to be added to the ADC input sample when asserted.

3.4.3.8 RECV_PMETER_SYNC Register

Register name: RECV_PMETER_SYNC **Page: 0x1800** **Address: 0x07**

BIT 15 BIT 8

recv_pmeter0_ ena	ssel_recv_pmeter0(2:0)			recv_pmeter1_ ena	ssel_ recv_pmeter1(2:0)		
0	0	0	0	0	0	0	0

BIT 7 BIT 0

recv_pmeter2_ ena	ssel_ recv_pmeter2(2:0)			recv_pmeter3_ ena	ssel_ recv_pmeter3(2:0)		
0	0	0	0	0	0	0	0

recv_pmeter0_ena : Enables the Receive Input Interface pmeter0 block when set

recv_pmeter1_ena : Enables the Receive Input Interface pmeter1 block when set

recv_pmeter2_ena : Enables the Receive Input Interface pmeter2 block when set

recv_pmeter3_ena : Enables the Receive Input Interface pmeter3 block when set

ssel_ recv_pmeter0(2:0) : Selects the sync source for the Receive Input Interface pmeter0 block

ssel_ recv_pmeter1(2:0) : Selects the sync source for the Receive Input Interface pmeter1 block

ssel_ recv_pmeter2(2:0) : Selects the sync source for the Receive Input Interface pmeter2 block

ssel_ recv_pmeter3(2:0) : Selects the sync source for the Receive Input Interface pmeter3 block

3.4.3.9 RECV_PMETER0_SQR_SUM_LSB Register

Register name: RECV_PMETER0_SQR_SUM_LSB **Page: 0x1800** **Address: 0x08**

BIT 15							BIT 8
recv_pmeter0_sqr_sum (15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter0_sqr_sum (7:0)							
0	0	0	0	0	0	0	0

recv_pmeter0_sqr_sum(15:0) : The sqr_sum register controls the number of samples to accumulate for a power measurement. Ia is (or Ia & Qa if complex mode is selected are) squared and accumulated. Eight Ia samples (or eight sample pairs of Ia and Qa samples) equal to one sqr_sum count. The accumulation interval is initiated when the sync is asserted and the programmed (8*sync_delay+2) samples has expired or when the interval start time is reached. When the (8*sqr_sum+1) sample time is reached, the accumulated powers are made available for MPU access and an interrupt is generated.

3.4.3.10 RECV_PMETER0_STRT_INTVL_LSB Register

Register name: RECV_PMETER0_STRT_INTVL_LSB **Page: 0x1800** **Address: 0x09**

BIT 15							BIT 8
recv_pmeter0_strt_intrvl (15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter0_strt_intrvl (7:0)							
0	0	0	0	0	0	0	0

recv_pmeter0_strt_intrvl(15:0) : The start interval timer is the interval over which the sqr_sum is restarted. The timer value is (8*strt_intrvl + 1) samples and must be larger than (8*sqr_sum+1) samples. The interval start counter and RMS power accumulation is started at the sync pulse after the programmed delay and every time the STRT_INTRVL counter reaches its limit.

3.4.3.11 RECV_PMETER0_SYNC_DLY Register

Register name: RECV_PMETER0_SYNC_DLY **Page: 0x1800** **Address: 0x0A**

BIT 15							BIT 8
delay_line_0(5:0)						unused	recv_pmeter0_ sync_delay(8)
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter0_sync_delay (7:0)							
0	0	0	0	0	0	0	0

delay_line_0(5:0) : Pointer offset for the rxin_a path variable delay line. Larger values result in larger pointer offsets and therefore more path delay.

recv_pmeter0_sync_delay(8:0) : Programmable start delay from sync, in eight sample units. The actual value is (8*sync_delay + 2) samples.

3.4.3.12 RECV_PMETER0_CONFIG Register

Register name: RECV_PMETER0_CONFIG **Page: 0x1800** **Address: 0x0B**

BIT 15							BIT 8
recv_pmeter0_sqr_sum(20:16)					recv_pmeter0_strt_intrvl(20:18)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter0_strt_ intrvl(17:16)		unused	unused	unused	ssel_delay_line_0(2:0)		
0	0	0	0	0	0	0	0

recv_pmeter0_sqr_sum(20:16) : MSBs of sqr_sum value, in 8 sample units

recv_pmeter0_strt_intrvl(20:16) : MSBs of start interval value, in 8 sample units.

ssel_delay_line_0(2:0) : Sync source selection for the 64 sample delay line pointer value update

3.4.3.13 RECV_PMETER1_SQR_SUM_LSB Register

Register name: RECV_PMETER1_SQR_SUM_LSB **Page: 0x1800** **Address: 0x0C**

BIT 15							BIT 8
recv_pmeter1_sqr_sum (15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter1_sqr_sum (7:0)							
0	0	0	0	0	0	0	0

recv_pmeter1_sqr_sum(15:0) : Lower 16bits of the sqr_sum interval timer, in 8 sample units.

3.4.3.14 RECV_PMETER1_STRT_INTVL_LSB Register

Register name: RECV_PMETER1_STRT_INTVL_LSB **Page: 0x1800** **Address: 0x0D**

BIT 15 BIT 8

recv_pmeter1_strt_intrvl (15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

recv_pmeter1_strt_intrvl (7:0)							
0	0	0	0	0	0	0	0

recv_pmeter1_strt_intrvl(15:0) : Lower 16bits of the interval timer, in 8 sample units.

3.4.3.15 RECV_PMETER1_SYNC_DLY Register

Register name: RECV_PMETER1_SYNC_DLY **Page: 0x1800** **Address: 0x0E**

BIT 15 BIT 8

delay_line_1(5:0)						unused	recv_pmeter1_ sync_ delay(8)
0	0	0	0	0	0	0	0

BIT 7 BIT 0

recv_pmeter1_sync_delay (7:0)							
0	0	0	0	0	0	0	0

delay_line_1(5:0) : Pointer offset for the rxin_b path variable delay line. Larger values result in larger pointer offsets and therefore more path delay

recv_pmeter1_sync_delay(8:0) : Programmable start delay from sync, in 8 sample units.

3.4.3.16 RECV_PMETER1_CONFIG Register

Register name: RECV_PMETER1_CONFIG **Page: 0x1800** **Address: 0x0F**

BIT 15							BIT 8
recv_pmeter1_sqr_sum(20:16)					recv_pmeter1_strt_intrvl(20:18)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter1_strt_ intrvl(17:16)		unused	unused	unused	ssel_delay_line_1(2:0)		
0	0	0	0	0	0	0	0

recv_pmeter1_sqr_sum(20:16) : MSBs of sqr_sum value, in 8 sample units.

recv_pmeter1_strt_intrvl(20:16) : MSBs of start interval value, in 8 sample units.

ssel_delay_line_1(2:0) : Sync source selection for the 64 sample delay line pointer value update

3.4.3.17 RECV_PMETER2_SQR_SUM_LSB Register

Register name: RECV_PMETER2_SQR_SUM_LSB **Page: 0x1800** **Address: 0x10**

BIT 15							BIT 8
recv_pmeter2_sqr_sum (15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter2_sqr_sum (7:0)							
0	0	0	0	0	0	0	0

recv_pmeter2_sqr_sum(15:0) : Lower 16bits of the sqr_sum interval timer, in 8 sample units.

3.4.3.18 RECV_PMETER2_STRT_INTVL_LSB Register

Register name: RECV_PMETER2_STRT_INTVL_LSB **Page: 0x1800** **Address: 0X11**

BIT 15							BIT 8
recv_pmeter2_strt_intrvl (15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter2_strt_intrvl (7:0)							
0	0	0	0	0	0	0	0

recv_pmeter2_strt_intrvl(15:0) : Lower 16bits of the interval timer, in 8 sample units.

3.4.3.19 RECV_PMETER2_SYNC_DLY Register

Register name: RECV_PMETER2_SYNC_DLY **Page: 0x1800** **Address: 0x12**

BIT 15							BIT 8
delay_line_2(5:0)						unused	recv_pmeter2_ sync_ delay(8)
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter2_sync_delay (7:0)							
0	0	0	0	0	0	0	0

delay_line_2(5:0) : Pointer offset for the rxin_c path variable delay line. Larger values result in larger pointer offsets and therefore more path delay.

recv_pmeter2_sync_delay (8:0) : Programmable start delay from sync, in 8 sample units.

3.4.3.20 RECV_PMETER2_CONFIG Register

Register name: RECV_PMETER2_CONFIG **Page: 0x1800** **Address: 0X13**

BIT 15							BIT 8
recv_pmeter2_sqr_sum(20:16)					recv_pmeter2_strt_intrvl(20:18)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter2_strt_ intrvl(17:16)		unused	unused	unused	ssel_delay_line_2(2:0)		
0	0	0	0	0	0	0	0

recv_pmeter2_sqr_sum(20:16) : MSBs of sqr_sum value, in 8 sample units.

recv_pmeter2_strt_intrvl(20:16) : MSBs of start interval value, in 8 sample units.

ssel_delay_line_2(2:0) : Sync source selection for the 64 sample delay line pointer value update.

3.4.3.21 RECV_PMETER3_SQR_SUM_LSB Register

Register name: RECV_PMETER3_SQR_SUM_LSB **Page: 0x1800** **Address: 0x14**

BIT 15 BIT 8

recv_pmeter3_sqr_sum (15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

recv_pmeter3_sqr_sum (7:0)							
0	0	0	0	0	0	0	0

recv_pmeter3_sqr_sum(15:0) : Lower 16bits of the sqr_sum interval timer, in 8 sample units.

3.4.3.22 RECV_PMETER3_STRT_INTVL_LSB Register

Register name: RECV_PMETER3_STRT_INTVL_LSB **Page: 0x1800** **Address: 0x15**

BIT 15 BIT 8

recv_pmeter3_strt_intrvl (15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

recv_pmeter3_strt_intrvl (7:0)							
0	0	0	0	0	0	0	0

recv_pmeter3_strt_intrvl(15:0) : Lower 16bits of the interval timer, in 8 sample units.

3.4.3.23 RECV_PMETER3_SYNC_DLY Register

Register name: RECV_PMETER3_SYNC_DLY **Page: 0x1800** **Address: 0x16**

BIT 15 BIT 8

delay_line_3(5:0)						unused	recv_pmeter3_sync_delay(8)
0	0	0	0	0	0	0	0

BIT 7 BIT 0

recv_pmeter3_sync_delay (7:0)							
0	0	0	0	0	0	0	0

delay_line_3(5:0) : Pointer offset for the rxin_d path variable delay line. Larger values result in larger pointer offsets and therefore more path delay.

recv_pmeter3_sync_delay(8:0) : Programmable start delay from sync, in 8 sample units.

3.4.3.24 RECV_PMETER3_CONFIG Register

Register name: RECV_PMETER3_CONFIG **Page: 0x1800** **Address: 0x17**

BIT 15							BIT 8
recv_pmeter3_sqr_sum(20:16)					recv_pmeter3_strt_intrvl(20:18)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter3_strt_ intrvl(17:16)		unused	unused	unused	ssel_delay_line_3(2:0)		
0	0	0	0	0	0	0	0

recv_pmeter3_sqr_sum(20:16) : MSBs of sqr_sum value, in 8 sample units

recv_pmeter3_strt_intrvl(20:16) : MSBs of start interval value, in 8 sample units

ssel_delay_line_3(2:0) : Sync source selection for the 64 sample delay line pointer value update

3.4.3.25 RECV_SLF_TST_VALUE Register

Register name: RECV_SLF_TST_VALUE **Page: 0x1800** **Address: 0x18**

BIT 15							BIT 8
self_test_constant(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
self_test_constant(7:0)							
0	0	0	0	0	0	0	0

self_test_constant(15:0) : 16 bit constant presented at the test and noise generator output when enabled. Used for test and debug purposes.

3.4.3.26 RECV_PMETER0_LSB Register

Register name: RECV_PMETER0_LSB **Page: 0x1820** **Address: 0x20** **READ ONLY**

BIT 15							BIT 8
recv_pmeter0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter0(7:0)							
0	0	0	0	0	0	0	0

recv_pmeter0(15:0) : Lower bits of the power meter 0 measurement

3.4.3.27 RECV_PMETER0_MID Register

Register name: RECV_PMETER0_MID **Page: 0x1820** **Address: 0x21** **READ ONLY**

BIT 15							BIT 8
recv_pmeter0(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter0(23:16)							
0	0	0	0	0	0	0	0

recv_pmeter0(31:16) : Mid bits of the power meter 0 measurement

3.4.3.28 RECV_PMETER0_LMSB Register

Register name: RECV_PMETER0_LMSB **Page: 0x1820** **Address: 0x22** **READ ONLY**

BIT 15							BIT 8
recv_pmeter0(47:40)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter0(39:32)							
0	0	0	0	0	0	0	0

recv_pmeter0(47:32) : Lower MSB bits of the power meter 0 measurement

3.4.3.29 RECV_PMETER0_UMSB Register

Register name: RECV_PMETER0_UMSB **Page: 0x1820** **Address: 0x23** **READ ONLY**

BIT 15							BIT 8
unused	unused	unused	unused	unused	unused	recv_pmeter0(57:56)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter0(55:48)							
0	0	0	0	0	0	0	0

recv_pmeter0(57:48) : Upper MSB bits of the power meter 0 measurement

3.4.3.30 RECV_PMETER1_LSB Register

Register name: RECV_PMETER1_LSB **Page: 0x1820** **Address: 0x24** **READ ONLY**

BIT 15							BIT 8
recv_pmeter1(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter1(7:0)							
0	0	0	0	0	0	0	0

recv_pmeter1(15:0) : Lower bits of the power meter 1 measurement

3.4.3.31 RECV_PMETER1_MID Register

Register name: RECV_PMETER1_MID **Page: 0x1820** **Address: 0x25** **READ ONLY**

BIT 15							BIT 8
recv_pmeter1(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter1(23:16)							
0	0	0	0	0	0	0	0

recv_pmeter1(31:16) : Mid bits of the power meter 1 measurement

3.4.3.32 RECV_PMETER1_LMSB Register

Register name: RECV_PMETER1_LMSB **Page: 0x1820** **Address: 0x26** **READ ONLY**

BIT 15							BIT 8
recv_pmeter1(47:40)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter1(39:32)							
0	0	0	0	0	0	0	0

recv_pmeter1(47:32) : Lower MSB bits of the power meter 1 measurement

3.4.3.33 RECV_PMETER1_UMSB Register

Register name: RECV_PMETER1_UMSB **Page: 0x1820** **Address: 0x27** **READ ONLY**

BIT 15							BIT 8
unused	unused	unused	unused	unused	unused	recv_pmeter1(57:56)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter1(55:48)							
0	0	0	0	0	0	0	0

recv_pmeter1(57:48) : Upper MSB bits of the power meter 1 measurement

3.4.3.34 RECV_PMETER2_LSB Register

Register name: RECV_PMETER2_LSB **Page: 0x1820** **Address: 0x28** **READ ONLY**

BIT 15							BIT 8
recv_pmeter2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter2(7:0)							
0	0	0	0	0	0	0	0

recv_pmeter2(15:0) : Lower bits of the power meter 2 measurement

3.4.3.35 RECV_PMETER2_MID Register

Register name: RECV_PMETER2_MID **Page: 0x1820** **Address: 0x29** **READ ONLY**

BIT 15							BIT 8
recv_pmeter2(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter2(23:16)							
0	0	0	0	0	0	0	0

recv_pmeter2(31:16) : Mid bits of the power meter 2 measurement

3.4.3.36 RECV_PMETER2_LMSB Register

Register name: RECV_PMETER2_LMSB **Page: 0x1820** **Address: 0x2A** **READ ONLY**

BIT 15							BIT 8
recv_pmeter2(47:40)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter2(39:32)							
0	0	0	0	0	0	0	0

recv_pmeter2(47:32) : Lower MSB bits of the power meter 2 measurement

3.4.3.37 RECV_PMETER2_UMSB Register

Register name: RECV_PMETER2_UMSB **Page: 0x1820** **Address: 0x2B** **READ ONLY**

BIT 15							BIT 8
unused	unused	unused	unused	unused	unused	recv_pmeter2(57:56)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter2(55:48)							
0	0	0	0	0	0	0	0

recv_pmeter2(57:48) : Upper MSB bits of the power meter 2 measurement

3.4.3.38 RECV_PMETER3_LSB Register

Register name: RECV_PMETER3_LSB **Page: 0x1820** **Address: 0x2C** **READ ONLY**

BIT 15 BIT 8

recv_pmeter3(15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

0	0	0	0	0	0	0	0

recv_pmeter3(15:0) : Lower bits of the power meter 3 measurement

3.4.3.39 RECV_PMETER3_MID Register

Register name: RECV_PMETER3_MID **Page: 0x1820** **Address: 0x2D** **READ ONLY**

BIT 15 BIT 8

recv_pmeter3(31:24)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

recv_pmeter3(23:16)							
0	0	0	0	0	0	0	0

recv_pmeter3(31:16) : Mid bits of the power meter 3 measurement

3.4.3.40 RECV_PMETER3_LMSB Register

Register name: RECV_PMETER3_LMSB **Page: 0x1820** **Address: 0x2E** **READ_ONLY**

BIT 15 BIT 8

recv_pmeter3(47:40)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

recv_pmeter3(39:32)							
0	0	0	0	0	0	0	0

recv_pmeter3(47:32) : Lower MSB bits of the power meter 3 measurement

3.4.3.41 RECV_PMETER3_UMSB Register

Register name: RECV_PMETER3_UMSB **Page: 0x1820** **Address: 0x2F** **READ_ONLY**

BIT 15							BIT 8
unused	unused					recv_pmeter3(57:56)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
recv_pmeter3(55:48)							
0	0	0	0	0	0	0	0

recv_pmeter3(57:48) : Upper MSB bits of the power meter 3 measurement

3.4.4 Receive AGC Controls

3.4.4.1 RAGC_CONFIG0 Register

Register name: RAGC_CONFIG0 **Page: 0x1840** **Address: 0x00**

BIT 15							BIT 8
hp_ena_0	hp_ena_1	hp_ena_2	hp_ena_3	sd_ena_0	sd_ena_1	sd_ena_2	sd_ena_3
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ragc_ bypass_0	ragc_ bypass_1	ragc_ bypass_2	ragc_ bypass_3	unused	unused	unused	unused
0	0	0	0	0	0	0	0

hp_ena_X : Enables the high pass filter in receive AGC X when set.

sd_ena_X : Enables the Signal Detect block in receive AGC X when set.

ragc_bypass_X : Bypasses the receive AGC X block when set.

3.4.4.2 RAGC_CONFIG1 Register

Register name: RAGC_CONFIG1 **Page: 0x1840** **Address: 0x01**

BIT 15							BIT 8
ragc_ freeze_0	ragc_ freeze_1	ragc_ freeze_2	ragc_ freeze_3	ragc_ clear_0	ragc_ clear_1	ragc_ clear_2	ragc_ clear_3
0	0	0	0	0	0	0	0

BIT 7							BIT 0
complex01	complex23	ssel_ragc_interval_0(2:0)			ssel_ragc_interval_1(2:0)		
0	0	0	0	0	0	0	0

ragc_freeze_X : Freezes the receive AGC block when set.

ragc_clear_X : Clears the loop error accumulator when set.

complex01 : When set, receive AGC 0 uses complex input with the second sample stream coming from receive AGC 1. The clip detect, high pass, and squarer from receive AGC 1 are used to generate inputs for receive AGC 0.

complex23 : When set, receive AGC 2 uses complex input with the second sample stream coming from receive AGC 3. The clip detect, high pass, and squarer from receive AGC 3 are used to generate inputs for receive AGC 2.

ssel_ragc_interval_0(2:0) : Selects the sync source for receive AGC 0. After a programmed delay from sync, the interval update timer is started.

ssel_ragc_interval_1(2:0) : Selects the sync source for receive AGC 1. After a programmed delay from sync, the interval update timer is started.

3.4.4.3 RAGC_CONFIG2 Register

Register name: RAGC_CONFIG2 **Page: 0x1840** **Address: 0x02**

BIT 15							BIT 8
ssel_ragc_freeze_0(2:0)			ssel_ragc_freeze_1(2:0)			ssel_ragc_ freeze_2(2:1)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ssel_ragc_freeze_2(0)	ssel_ragc_freeze_3(2:0)			unused	ssel_ragc_interval_2(2:0)		
0	0	0	0	0	0	0	0

ssel_ragc_freeze_X(2:0) : Selects the sync source that will freeze the receive AGC loop when asserted.

ssel_ragc_interval_2(2:0) : Selects the sync source for receive AGC 2. After a programmed delay from sync, the interval update timer is started.

3.4.4.4 RAGC_CONFIG3 Register

Register name: RAGC_CONFIG3 **Page: 0x1840** **Address: 0x03**

BIT 15							BIT 8
ssel_ragc_clear_0(2:0)			ssel_ragc_clear_1(2:0)			ssel_ragc_ clear_2(2:1)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ssel_ragc_ clear_2(0)	ssel_ragc_clear_3(2:0)			unused	ssel_ragc_interval_3(2:0)		
0	0	0	0	0	0	0	0

ssel_agc_clear_X(2:0 : Controls the selection of the sync that will clear the receive AGC error accumulator.

ssel_agc_interval_3(2:0) : Selects the sync source for receive AGC 3. After a programmed delay from sync, the interval update timer is started.

3.4.4.5 RAGC0_INTEGINVL_LSB Register

Register name: RAGC0_INTEGINVL_LSB **Page: 0x1840** **Address: 0x04**

BIT 15							BIT 8
integ_interval_0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
integ_interval_0(7:0)							
0	0	0	0	0	0	0	0

integ_interval_0(15:0) : The 16 LSBs of the integration time for receive AGC 0.

3.4.4.6 RAGC0_INTEGINVL_MSB Register

Register name: RAGC0_INTEGINVL_MSB **Page: 0x1840** **Address: 0x05**

BIT 15							BIT 8
ragc_update_0(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
integ_interval_0(23:16)							
0	0	0	0	0	0	0	0

ragc_update_0(7:0) : Sets the number of receive AGC updates per sync event (0x00 is infinite).

integ_interval_0(23:16) : The eight MSBs of the integration time for receive AGC 0.

3.4.4.7 RAGC0_CONFIG0 Register

Register name: RAGC0_CONFIG0 **Page: 0x1840** **Address: 0x06**

<table>
<tr><td>BIT 15</td><td></td><td></td><td></td><td></td><td></td><td></td><td>BIT 8</td></tr>
<tr><td colspan="8">ragc_sync_delay_0(7:0)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

<table>
<tr><td>BIT 7</td><td></td><td></td><td></td><td></td><td></td><td></td><td>BIT 0</td></tr>
<tr><td colspan="3">hp_corner_0(2:0)</td><td colspan="5">acc_shift_0(4:0)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

ragc_sync_delay_0(7:0) : The input sync to the receive AGC block is delayed by this number of samples.

hp_corner_0(2:0) : Sets the corner frequency of the high pass filter. Larger values result in higher corner frequencies

acc_shift_0(4:0) : Selects the integrated power measurements result bits to be used as the error lookup table address. A larger number means fewer samples will have to be integrated to achieve the same result.

3.4.4.8 RAGC0_CONFIG1 Register

Register name: RAGC0_CONFIG1 **Page: 0x1840** **Address: 0x07**

<table>
<tr><td>BIT 15</td><td></td><td></td><td></td><td></td><td></td><td></td><td>BIT 8</td></tr>
<tr><td colspan="6">acc_offset_0(5:0)</td><td colspan="2">err_shift_0(4:3)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

<table>
<tr><td>BIT 7</td><td></td><td></td><td></td><td></td><td></td><td></td><td>BIT 0</td></tr>
<tr><td colspan="3">err_shift_0(2:0)</td><td colspan="5">delay_adj_0(4:0)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

acc_offset_0(5:0) : Constant subtracted from the integrated power measurement result before the error lookup table.

err_shift_0(4:0) : Adjusts the loop gain by controlling the amount of shifting applied to the error lookup table output. Larger values result in higher gain.

delay_adj_0(4:0) : Sets the delay difference, in samples, between the DVGA outputs and the value applied to the sample multiplier.

3.4.4.9 RAGC0_SD_THRESH Register

Register name: RAGC0_SD_THRESH **Page: 0x1840** **Address: 0x08**

BIT 15							BIT 8
sd_thresh_0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd_thresh_0(7:0)							
0	0	0	0	0	0	0	0

sd_thresh_0(15:0) : This is the threshold used by the Signal Detect block to determine if there is signal on the inputs. The comparison is done to the output of the squarer block, which is a 32 bit word. Because of this, these bits are aligned with bits 24 down to 8 of the 32 bit squared value.

3.4.4.10 RAGC0_SD_TIMER Register

Register name: RAGC0_SD_TIMER **Page: 0x1840** **Address: 0x09**

BIT 15							BIT 8
sd _timer_0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd _timer_0(7:0)							
0	0	0	0	0	0	0	0

sd_timer_0(15:0) : Qualification window timer for loss of input signal.

3.4.4.11 RAGC0_SD_SAMPLES Register

Register name: RAGC0_SD_SAMPLES **Page: 0x1840** **Address: 0x0A**

BIT 15							BIT 8
sd_samples_0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd_samples_0(7:0)							
0	0	0	0	0	0	0	0

sd_samples_0(15:0) : Number of samples that must be below the sd_thresh_X within the sd_timer_X timer value for the loss of signal condition to occur.

3.4.4.12 RAGC0_CLIP_HITHRESH Register

Register name: RAGC0_CLIP_HITHRESH **Page: 0x1840** **Address: 0x0B**

BIT 15							BIT 8
clip_hi_thresh_0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_hi_thresh_0(7:0)							
0	0	0	0	0	0	0	0

clip_hi_thresh_0(15:0) : The high threshold value for clip detection.

3.4.4.13 RAGC0_CLIP_LOTHRESH Register

Register name: RAGC0_CLIP_LOTHRESH **Page: 0x1840** **Address: 0x0C**

BIT 15							BIT 8
clip_lo_thresh_0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_thresh_0(7:0)							
0	0	0	0	0	0	0	0

clip_lo_thresh_0(15:0) : The low threshold value for clip detection.

3.4.4.14 RAGC0_CLIP_HITIMER Register

Register name: RAGC0_CLIP_HITIMER **Page: 0x1840** **Address: 0x0D**

BIT 15							BIT 8
clip_hi_timer_0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_hi_timer_0(7:0)							
0	0	0	0	0	0	0	0

clip_hi_timer_0(15:0) : The high timer value in Samples

3.4.4.15 RAGC0_CLIP_LOTIMER Register

Register name: RAGC0_CLIP_LOTIMER **Page: 0x1840** **Address: 0x0E**

BIT 15							BIT 8
clip_lo_timer_0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_timer_0(7:0)							
0	0	0	0	0	0	0	0

clip_lo_timer_0(15:0) : The low timer value in Samples.

3.4.4.16 RAGC0_CLIP_SAMPLES Register

Register name: RAGC0_CLIP_SAMPLES **Page: 0x1840** **Address: 0x0F**

BIT 15							BIT 8
clip_hi_samples_0(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_samples_0(7:0)							
0	0	0	0	0	0	0	0

clip_hi_samples_0(7:0) : Number of samples above the high threshold within the clip high time to enable the clip event.

clip_lo_samples_0(7:0) : Number of samples below the low threshold within the clip low time to disable the clip event.

3.4.4.17 RAGC0_CLIP_ERROR Register

Register name: RAGC0_CLIP_ERROR **Page: 0x1840** **Address: 0x10**

BIT 15							BIT 8
clip_error_0(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_error_0(7:0)							
0	0	0	0	0	0	0	0

clip_error_0(15:0) : This is the error value that is added into the loop accumulator when a clip is detected.

3.4.4.18 RAGC1_INTEGINVL_LSB Register

Register name: RAGC1_INTEGINVL_LSB **Page: 0x1840** **Address: 0x11**

BIT 15							BIT 8
integ_interval_1(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
integ_interval_1(7:0)							
0	0	0	0	0	0	0	0

integ_interval_1(15:0) : The LSBs of the integration time for receive AGC 1

3.4.4.19 RAGC1_INTEGINVL_MSB Register

Register name: RAGC1_INTEGINVL_MSB **Page: 0x1840** **Address: 0x12**

BIT 15							BIT 8
ragc_update_1(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
integ_interval_1(23:16)							
0	0	0	0	0	0	0	0

ragc_update_1(7:0) : Sets the number of receive AGC updates per sync event (0x00 is infinite).

integ_interval_1(23:16) : The MSBs of the integration time for receive AGC 1

3.4.4.20 RAGC1_CONFIG0 Register

Register name: RAGC1_CONFIG0 **Page: 0x1840** **Address: 0x13**

BIT 15							BIT 8
ragc_sync_delay_1(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
hp_corner_1(2:0)			acc_shift_1(4:0)				
0	0	0	0	0	0	0	0

ragc_sync_delay_1(7:0) : The input sync to the receive AGC block is delayed by this value of samples.

hp_corner_1(2:0) : This sets the corner frequency of the High Pass filter. Larger values result in higher corner frequencies.

acc_shift_1(4:0) : Selects the integrated power measurements result bits to be used as the error lookup table address. A larger number means fewer samples will have to be integrated to achieve the same result.

3.4.4.21 RAGC1_CONFIG1 Register

Register name: RAGC1_CONFIG1 **Page: 0x1840** **Address: 0x14**

BIT 15							BIT 8
acc_offset_1(5:0)						err_shift_1(4:3)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
err_shift_1(2:0)			delay_adj_1(4:0)				
0	0	0	0	0	0	0	0

acc_offset_1(5:0) : Constant subtracted from the integrated power measurement result before the error lookup table

err_shift_1(4:0) : Controls the loop gain by left shifting the error output. Larger values result in higher gain.

delay_adj_1(4:0) : Sets the delay difference, in samples, between the DVGA outputs and the value applied to the sample multiplier.

3.4.4.22 RAGC1_SD_THRESH Register

Register name: RAGC1_SD_THRESH **Page: 0x1840** **Address: 0x15**

BIT 15							BIT 8
sd_thresh_1(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd_thresh_1(7:0)							
0	0	0	0	0	0	0	0

sd_thresh_1(15:0) : This is the threshold used by the Signal Detect block to determine if there is signal on the inputs. The comparison is done to the output of the squarer block, which is a 32 bit word. Because of this, these bits are aligned with bits 24 down to 8 of the 32 bit squared value.

3.4.4.23 RAGC1_SD_TIMER Register

Register name: RAGC1_SD_TIMER **Page: 0x1840** **Address: 0x16**

BIT 15							BIT 8
sd_timer_1(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd_timer_1(7:0)							
0	0	0	0	0	0	0	0

sd_timer_1(15:0) : After the first no signal sample occurs, this is the amount of samples that control the length of time to determine the loss of signal condition.

3.4.4.24 RAGC1_SD_SAMPLES Register

Register name: RAGC1_SD_SAMPLES **Page: 0x1840** **Address: 0x17**

BIT 15							BIT 8
sd_samples_1(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd_samples_1(7:0)							
0	0	0	0	0	0	0	0

sd_samples_1(15:0) : Number of samples that must be below the sd_thresh_X threshold within the sd_timer_X timer value for the loss of signal condition to occur.

3.4.4.25 RAGC1_CLIP_HITHRESH Register

Register name: RAGC1_CLIP_HITHRESH **Page: 0x1840** **Address: 0x18**

BIT 15							BIT 8
clip_hi_thresh_1(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_hi_thresh_1(7:0)							
0	0	0	0	0	0	0	0

clip_hi_thresh_1(15:0) : The high threshold value for clip detection.

3.4.4.26 RAGC1_CLIP_LOTHRESH Register

Register name: RAGC1_CLIP_LOTHRESH **Page: 0x1840** **Address: 0x19**

BIT 15							BIT 8
clip_lo_thresh_1(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_thresh_1(7:0)							
0	0	0	0	0	0	0	0

clip_lo_thresh_1(15:0) The low threshold value for clip detection.

3.4.4.27 RAGC1_CLIP_HITIMER Register

Register name: RAGC1_CLIP_HITIMER **Page: 0x1840** **Address: 0x1A**

BIT 15 BIT 8

clip_hi_timer_1(15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

clip_hi_timer_1(7:0)							
0	0	0	0	0	0	0	0

clip_hi_timer_1(15:0) : The high timer value in samples.

3.4.4.28 RAGC1_CLIP_LOTIMER Register

Register name: RAGC1_CLIP_LOTIMER **Page: 0x1840** **Address: 0x1B**

BIT 15 BIT 8

clip_lo_timer_1(15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

clip_lo_timer_1(7:0)							
0	0	0	0	0	0	0	0

clip_lo_timer_1(15:0) : The low timer value in samples.

3.4.4.29 RAGC1_CLIP_SAMPLES Register

Register name: RAGC1_CLIP_SAMPLES **Page: 0x1840** **Address: 0x1C**

BIT 15 BIT 8

clip_hi_samples_1(7:0)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

clip_lo_samples_1(7:0)							
0	0	0	0	0	0	0	0

clip_hi_samples_1(7:0) : Number of samples above the high threshold within the clip high time to enable the clip event.

clip_lo_samples_1(7:0) : Number of samples below the low threshold within the clip low time to disable the clip event.

3.4.4.30 RAGC1_CLIP_ERROR Register

Register name: RAGC1_CLIP_ERROR **Page: 0x1840** **Address: 0x1D**

BIT 15							BIT 8
clip_error_1(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_error_1(7:0)							
0	0	0	0	0	0	0	0

clip_error_1(15:0) : This is the error value that is added into the loop accumulator when a clip is detected.

3.4.4.31 RAGC2_INTEGINVL_LSB Register

Register name: RAGC2_INTEGINVL_LSB **Page: 0x1840** **Address: 0x1E**

BIT 15							BIT 8
integ_interval_2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
integ_interval_2(7:0)							
0	0	0	0	0	0	0	0

integ_interval_2(15:0) : The LSBs of the integration time for receive AGC 2

3.4.4.32 RAGC2_INTEGINVL_MSB Register

Register name: RAGC2_INTEGINVL_MSB **Page: 0x1840** **Address: 0x1F**

BIT 15							BIT 8
ragc_update_2(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
integ_interval_2(23:16)							
0	0	0	0	0	0	0	0

ragc_update_2(7:0) : Sets the number of receive AGC updates per sync event (0x00 is infinite).

integ_interval_2(23:16) : The MSBs of the integration time for receive AGC 2

3.4.4.33 RAGC2_CONFIG0 Register

Register name: RAGC2_CONFIG0 **Page: 0x1860** **Address: 0x20**

BIT 15							BIT 8
ragc_sync_delay_2(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
hp_corner_2(2:0)			acc_shift_2(4:0)				
0	0	0	0	0	0	0	0

ragc_sync_delay_2(7:0) : The input sync to the receive AGC block is delayed by this value of samples.

hp_corner_2(2:0) : This sets the corner frequency of the High Pass filter. Larger values result in higher corner frequencies.

acc_shift_2(4:0) : Selects the integrated power measurements result bits to be used as the error lookup table address. A larger number means fewer samples will have to be integrated to achieve the same result.

3.4.4.34 RAGC2_CONFIG1 Register

Register name: RAGC2_CONFIG1 **Page: 0x1860** **Address: 0x21**

BIT 15							BIT 8
acc_offset_2(5:0)						err_shift_2(4:3)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
err_shift_2(2:0)			delay_adj_2(4:0)				
0	0	0	0	0	0	0	0

acc_offset_2(5:0) : Constant subtracted from the integrated power measurement result before the error lookup table.

err_shift_2(4:0) : Controls the loop gain by left shifting the error output. Larger values result in higher gain..

delay_adj_2(4:0) : Sets the delay difference, in samples, between the DVGA outputs and the value applied to the sample multiplier.

3.4.4.35 RAGC2_SD_THRESH Register

Register name: RAGC2_SD_THRESH **Page: 0x1860** **Address: 0x22**

BIT 15							BIT 8
sd_thresh_2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd_thresh_2(7:0)							
0	0	0	0	0	0	0	0

sd_thresh_2(15:0) : This is the threshold used by the Signal Detect block to determine if there is signal on the inputs. The comparison is done to the output of the squarer block, which is a 32 bit word. Because of this, these bits are aligned with bits 24 down to 8 of the 32 bit squared value.

3.4.4.36 RAGC2_SD_TIMER Register

Register name: RAGC2_SD_TIMER **Page: 0x1860** **Address: 0x23**

BIT 15							BIT 8
sd_timer_2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd_timer_2(7:0)							
0	0	0	0	0	0	0	0

sd_timer_2(15:0) : After the first no signal sample occurs, this is the amount of samples that control the length of time to determine the loss of signal condition.

3.4.4.37 RAGC2_SD_SAMPLES Register

Register name: RAGC2_SD_SAMPLES **Page: 0x1860** **Address: 0x24**

BIT 15							BIT 8
sd_samples_2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd_samples_2(7:0)							
0	0	0	0	0	0	0	0

sd_samples_2(15:0) : Number of samples that must be below the sd_thresh_X threshold within the sd_timer_X timer value for the loss of signal condition to occur.

3.4.4.38 RAGC2_CLIP_HITHRESH Register

Register name: RAGC2_CLIP_HITHRESH **Page: 0x1860** **Address: 0x25**

BIT 15							BIT 8
clip_hi_thresh_2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_hi_thresh_2(7:0)							
0	0	0	0	0	0	0	0

clip_hi_thresh_2(15:0) : The high threshold value for clip detection.

3.4.4.39 RAGC2_CLIP_LOTHRESH Register

Register name: RAGC2_CLIP_LOTHRESH **Page: 0x1860** **Address: 0x26**

BIT 15							BIT 8
clip_lo_thresh_2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_thresh_2(7:0)							
0	0	0	0	0	0	0	0

clip_lo_thresh_2(15:0) : The low threshold value for clip detection.

3.4.4.40 RAGC2_CLIP_HITIMER Register

Register name: RAGC2_CLIP_HITIMER **Page: 0x1860** **Address: 0x27**

BIT 15							BIT 8
clip_hi_timer_2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_hi_timer_2(7:0)							
0	0	0	0	0	0	0	0

clip_hi_timer_2(15:0) : The high timer value in samples

3.4.4.41 RAGC2_CLIP_LOTIMER Register

Register name: RAGC2_CLIP_LOTIMER **Page: 0x1860** **Address: 0x28**

BIT 15							BIT 8
clip_lo_timer_2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_timer_2(7:0)							
0	0	0	0	0	0	0	0

clip_lo_timer_2(15:0) : The low timer value in samples.

3.4.4.42 RAGC2_CLIP_SAMPLES Register

Register name: RAGC2_CLIP_SAMPLES **Page: 0x1860** **Address: 0x29**

BIT 15							BIT 8
clip_hi_samples_2(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_samples_2(7:0)							
0	0	0	0	0	0	0	0

clip_hi_samples_2(7:0) : Number of samples above the high threshold within the clip high time to enable the clip event.

clip_lo_samples_2(7:0) : Number of samples below the low threshold within the clip low time to disable the clip event.

3.4.4.43 RAGC2_CLIP_ERROR Register

Register name: RAGC2_CLIP_ERROR **Page: 0x1860** **Address: 0x2A**

BIT 15							BIT 8
clip_error_2(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_error_2(7:0)							
0	0	0	0	0	0	0	0

clip_error_2(15:0) : This is the error value that is added into the loop accumulator when a clip is detected.

3.4.4.44 RAGC3_INTEGINVL_LSB Register

Register name: RAGC3_INTEGINVL_LSB **Page: 0x1860** **Address: 0x2B**

BIT 15							BIT 8
integ_interval_3(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
integ_interval_3(7:0)							
0	0	0	0	0	0	0	0

integ_interval_3(15:0) : The LSBs of the integration time for receive AGC 3

3.4.4.45 RAGC3_INTEGINVL_MSB Register

Register name: RAGC3_INTEGINVL_MSB **Page: 0x1860** **Address: 0x2C**

BIT 15							BIT 8
ragc_update_3(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
integ_interval_3(23:16)							
0	0	0	0	0	0	0	0

ragc_update_3(7:0) : Sets the number of receive AGC updates per sync event (0x00 is infinite).

integ_interval_3(23:16) : The MSBs of the integration time for receive AGC 3

3.4.4.46 RAGC3_CONFIG0 Register

Register name: RAGC3_CONFIG0 **Page: 0x1860** **Address: 0x2D**

BIT 15							BIT 8
ragc_sync_delay_3(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
hp_corner_3(2:0)			acc_shift_3(4:0)				
0	0	0	0	0	0	0	0

ragc_sync_delay_3(7:0) : The input sync to the receive AGC block is delayed by this value of samples.

hp_corner_3(2:0) : This sets the corner frequency of the High Pass filter. Larger values result in higher corner frequencies.

acc_shift_3(4:0) : Selects the integrated power measurements result bits to be used as the error lookup table address. A larger number means fewer samples will have to be integrated to achieve the same result.

3.4.4.47 RAGC3_CONFIG1 Register

Register name: RAGC3_CONFIG1 **Page: 0x1860** **Address: 0x2E**

BIT 15 BIT 8

acc_offset_3(5:0)						err_shift_3(4:3)	
0	0	0	0	0	0	0	0

BIT 7 BIT 0

err_shift_3(2:0)			delay_adj_3(4:0)				
0	0	0	0	0	0	0	0

acc_offset_3(5:0) : Constant subtracted from the integrated power measurement result before the error lookup table

err_shift_3(4:0) : Controls the loop gain by left shifting the error output. Larger values result in higher gain.

delay_adj_3(4:0) : Sets the delay difference, in samples, between the DVGA outputs and the value applied to the sample multiplier.

3.4.4.48 RAGC3_SD_THRESH Register

Register name: RAGC3_SD_THRESH **Page: 0x1860** **Address: 0x2F**

BIT 15 BIT 8

sd_thresh_3(15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

sd_thresh_3(7:0)							
0	0	0	0	0	0	0	0

sd_thresh_3(15:0) : This is the threshold used by the Signal Detect block to determine if there is signal on the inputs. The comparison is done to the output of the squarer block, which is a 32 bit word. Because of this, these bits are aligned with bits 24 down to 8 of the 32 bit squared value.

3.4.4.49 RAGC3_SD_TIMER Register

Register name: RAGC3_SD_TIMER **Page: 0x1860** **Address: :
0x30**

BIT 15 BIT 8

sd_timer_3(15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

sd_timer_3(7:0)							
0	0	0	0	0	0	0	0

sd_timer_3(15:0) : After the first no signal sample occurs, this is the amount of samples that control the length of time to determine the loss of signal condition.

3.4.4.50 RAGC3_SD_SAMPLES Register

Register name: RAGC3_SD_SAMPLES **Page: 0x1860** **Address: 0x31**

BIT 15							BIT 8
sd_samples_3(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
sd_samples_3(7:0)							
0	0	0	0	0	0	0	0

sd_samples_3(15:0) : Number of samples that must be below the sd_thresh_X threshold within the sd_timer_X timer value for the loss of signal condition to occur.

3.4.4.51 RAGC3_CLIP_HITHRESH Register

Register name: RAGC3_CLIP_HITHRESH **Page: 0x1860** **Address: 0x32**

BIT 15							BIT 8
clip_hi_thresh_3(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_hi_thresh_3(7:0)							
0	0	0	0	0	0	0	0

clip_hi_thresh_3(15:0) : The high threshold value for clip detection.

3.4.4.52 RAGC3_CLIP_LOTHRESH Register

Register name: RAGC3_CLIP_LOTHRESH **Page: 0x1860** **Address: 0x33**

BIT 15							BIT 8
clip_lo_thresh_3(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_thresh_3(7:0)							
0	0	0	0	0	0	0	0

clip_lo_thresh_3(15:0) : The low threshold value for clip detection.

3.4.4.53 RAGC3_CLIP_HITIMER Register

Register name: RAGC3_CLIP_HITIMER **Page: 0x1860** **Address: 0x34**

BIT 15							BIT 8
clip_hi_timer_3(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_hi_timer_3(7:0)							
0	0	0	0	0	0	0	0

clip_hi_timer_3(15:0) : The clip high timer value in samples

3.4.4.54 RAGC3_CLIP_LOTIMER Register

Register name: RAGC3_CLIP_LOTIMER **Page: 0x1860** **Address: 0x35**

BIT 15							BIT 8
clip_lo_timer_3(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_timer_3(7:0)							
0	0	0	0	0	0	0	0

clip_lo_timer_3(15:0) : The clip low timer value in samples.

3.4.4.55 RAGC3_CLIP_SAMPLES Register

Register name: RAGC3_CLIP_SAMPLES **Page: 0x1860** **Address: 0x36**

BIT 15							BIT 8
clip_hi_samples_3(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_lo_samples_3(7:0)							
0	0	0	0	0	0	0	0

clip_hi_samples_3(7:0) : Number of samples above the high threshold within the clip high time to enable a clip event.

clip_lo_samples_3(7:0) : Number of samples below the low threshold within the clip low time to disable a clip event.

3.4.4.56 RAGC3_CLIP_ERROR Register

Register name: RAGC3_CLIP_ERROR **Page: 0x1860** **Address: 0x37**

BIT 15							BIT 8
clip_error_3(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
clip_error_3(7:0)							
0	0	0	0	0	0	0	0

clip_error_3(15:0) : Error value that is added into the loop accumulator when a clip is detected.

3.4.4.57 RAGC0_ACCUM_LSB Register

Register name: RAGC0_ACCUM_LSB **Page: 0x1860** **Address: 0x38** **READ ONLY**

BIT 15							BIT 8
ragc0_accum(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ragc0_accum (7:0)							
0	0	0	0	0	0	0	0

ragc0_accum(15:0) : lower 16 bits of the ragc0 error accumulator.

3.4.4.58 RAGC0_ACCUM_MSB Register

Register name: RAGC0_ACCUM_MSB **Page: 0x1860** **Address: 0x39** **READ ONLY**

BIT 15							BIT 8
ragc0_accum(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ragc0_accum (23:16)							
0	0	0	0	0	0	0	0

ragc0_accum(31:16) : upper 16 bits of the ragc0 error accumulator.

3.4.4.59 RAGC1_ACCUM_LSB Register

Register name: RAGC1_ACCUM_LSB **Page: 0x1860** **Address: 0x3A** **READ ONLY**

BIT 15							BIT 8
ragc1_accum(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ragc1_accum (7:0)							
0	0	0	0	0	0	0	0

ragc1_accum(15:0) : lower 16 bits of the ragc1 error accumulator.

3.4.4.60 RAGC1_ACCUM_MSB Register

Register name: RAGC1_ACCUM_MSB **Page: 0x1860** **Address: 0x3B** **READ ONLY**

BIT 15							BIT 8
ragc1_accum(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ragc1_accum (23:16)							
0	0	0	0	0	0	0	0

ragc1_accum(31:16) : upper 16 bits of the ragc1 error accumulator.

3.4.4.61 RAGC2_ACCUM_LSB Register

Register name: RAGC2_ACCUM_LSB **Page: 0x1860** **Address: 0x3C** **READ ONLY**

BIT 15							BIT 8
ragc2_accum(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ragc2_accum (7:0)							
0	0	0	0	0	0	0	0

ragc2_accum(15:0) : lower 16 bits of the ragc2 error accumulator.

3.4.4.62 RAGC2_ACCUM_MSB Register

Register name: RAGC2_ACCUM_MSB **Page: 0x1860** **Address: 0x3D** **READ ONLY**

BIT 15							BIT 8
ragc2_accum(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ragc2_accum (23:16)							
0	0	0	0	0	0	0	0

ragc2_accum(31:16) : upper 16 bits of the ragc2 error accumulator.

3.4.4.63 RAGC3_ACCUM_LSB Register

Register name: RAGC3_ACCUM_LSB **Page: 0x1860** **Address: 0x3E** **READ ONLY**

BIT 15							BIT 8
ragc3_accum(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ragc3_accum (7:0)							
0	0	0	0	0	0	0	0

ragc3_accum(15:0) : lower 16 bits of the ragc3 error accumulator.

3.4.4.64 RAGC3_ACCUM_MSB Register

Register name: RAGC3_ACCUM_MSB **Page: 0x1860** **Address: 0x3F** **READ ONLY**

BIT 15							BIT 8
ragc3_accum(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ragc3_accum (23:16)							
0	0	0	0	0	0	0	0

ragc3_accum(31:16) : upper 16 bits of the ragc3 error accumulator.

3.4.5 DDC Channel Controls

3.4.5.1 FIR_MODE Register

Register name: FIR_MODE **Page: 0x0%00** **Address: 0x00**
where:
% = 2×(DDC channel #)+1

BIT 15							BIT 8
cdma_mode	unused	unused	crastarttap_pfir(4:0)				
0	0	0	0	0	0	0	0

BIT 7							BIT 0
crastarttap_cfir(4:0)					unused	unused	unused
0	0	0	0	0	0	0	0

cdma_mode : When asserted the DDC block is in CDMA mode (2 streams per DDC block).

crastarttap_pfir : These bits define the number of taps that PFIR will use for the filtering.

crastarttap_cfir : These bits define the number of taps that CFIR will use for the filtering.

Formulas for the number of taps, in the different FIR's, using the crastarttap word.

DDC PFIR: 4*(crastarttap_pfir+1)
DDC PFIR long mode: 8*(crastarttap_pfir+1)
DDC CFIR: 2*(crastarttap_cfir+1)

3.4.5.2 FIR_GAIN Register

Register name: FIR_GAIN **Page: 0x0%00** **Address: 0x01**
where:
% = 2×(DDC channel #)+1

BIT 15							BIT 8
pfir_gain(2:0)			unused	unused	unused	unused	unused
0	0	0	0	0	0	0	0

BIT 7							BIT 0
unused	unused	cfir_gain	unused	unused	unused	unused	unused
0	0	0	0	0	0	0	0

pfir_gain(2:0) : PFIR gain, from 2e-19 to 2e-12 for the receive PFIR. (“000” = 2e-19 and “111” = 2e-12)

cfir_gain : When ‘0’ then the gain of the CFIR is 2e-19, otherwise when set to ‘1’ the gain is 2e-18.

3.4.5.3 SQR_SUM Register

Register name: SQR_SUM

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x02

BIT 15							BIT 8
pmeter_sqr_sum_ddc(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
pmeter_sqr_sum_ddc(7:0)							
0	0	0	0	0	0	0	0

pmeter_sqr_sum_ddc(15:0): The sqr_sum register is the number of 4 sample sets to accumulate for a power measurement. In CDMA mode, one sample set is the I & Q of the signal and diversity. Ia & Qa (signal) are each squared and accumulated and Ib & Qb (diversity) are squared and accumulated. In UMTS mode, each I and Q pair are squared and accumulated. 4 samples is equal to one SQR_SUM count. The count is initiated when the sync is asserted or when the interval start time is reached. When the SQR_NUM number is reached, the accumulated powers are made available for MPU access and an interrupt is generated.

3.4.5.4 STRT_INTRVL Register

Register name: STRT_INTRVL

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x03

BIT 15							BIT 8
pmeter_sync_delay_ddc(7:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
pmeter_interval_ddc(7:0)							
0	0	0	0	0	0	0	0

pmeter_sync_delay_ddc(7:0) : The delay from selected sync source to when the power calculation starts. The actual value is *sync_delay + 1*.

pmeter_interval_ddc(7:0) : The start interval timer is the interval over which the SQR_SUM is restarted and must be greater than the SQR_SUM. **The actual interval is interval +1, and must be greater than the sqr_sum interval.** 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.

3.4.5.5 CIC_MODE1 Register

Register name: CIC_MODE1

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x04

BIT 15							BIT 8
cic_scale_a(4:0)					cic_scale_b(4:2)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
cic_scale_b(1:0)		cic_gain_ ddc	cic_decim(4:0)				
0	0	0	0	0	0	0	0

cic_scale_a(4:0) : This sets the gain shift at the output of the A channel CIC. 0x00 is no shift, each increment by 1 increases the signal amplitude by 2X.

cic_scale_b(4:0) : This sets the gain shift at the output of the B channel CIC. 0x00 is no shift, each increment by 1 increases the signal amplitude by 2X.

cic_gain_ddc : Adds a fixed gain of 12dB at the CIC output when asserted.

cic_decim(4:0) : Sets the CIC decimation rate, where decimation is cic_decim + 1.

3.4.5.6 CIC_MODE2 Register

Register name: CIC_MODE2

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x05

BIT 15							BIT 8
cic_m2_ena_a(5:0)						cic_m2_ena_b(5:4)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
cic_m2_ena_b(3:0)				unused	unused	unused	unused
0	0	0	0	0	0	0	0

cic_m2_ena_a(5:0) : Programs the A 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(5:0) : Programs the 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.

3.4.5.7 TADJC Register

Register name: TADJC

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x06

DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING

BIT 15							BIT 8
unused	unused	unused	tadj_offset_coarse_a(2:0)			unused	unused
0	0	0	0	0	0	0	0

BIT 7							BIT 0
unused	tadj_offset_coarse_b(2:0)			unused	unused	unused	unused
0	0	0	0	0	0	0	0

tadj_offset_coarse_a(2:0) : This is the coarse time adjustment offset and acts as an offset from the write address in the delay ram. This value affects the A data in the path 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.

dj_offset_coarse_b(2:0) : Effects the B channel in CDMA, just as the above effects the A channel.

3.4.5.8 TADJF Register

Register name: TADJF

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x07

DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING

BIT 15							BIT 8
tadj_offset_fine_a(2:0)			tadj_offset_fine_b(2:0)			tadj_interp(2:1)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
tadj_interp(0)	unused	unused	unused	unused	unused	unused	unused
0	0	0	0	0	0	0	0

tadj_offset_fine_a(2:0) : This is the fine adjust (zero stuff offset) value. It adjusts the time delay at the rxclk rate. This value affects the A channel data in the path if CDMA mode is being used.

tadj_offset_fine_b(2:0) : Same as above except this value affects the B channel data in CDMA mode.

tadj_interp(2:0) : This is the interpolation (zero stuff) value for the fine time adjust block. Interpolation can be from 1 to 8 (tadj_interp + 1). This value affects the A and B data in the path if CDMA mode is being used.

3.4.5.9 PHASEADD0A Register

Register name: PHASEADD0A	**Page: 0x0%00** **where:** **% = 2×(DDC channel #)+1**	**Address: 0x08**	**DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING**

BIT 15							BIT 8
phase_add_a(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
phase_add_a(7:0)							
0	0	0	0	0	0	0	0

phase_add_a(15:0) This 32 bit word is used to control the frequency of the NCO. This value is added to the frequency accumulator every clock cycle *(UMTS mode and Main channel in CDMA mode)*.

3.4.5.10 PHASEADD1A Register

Register name: PHASEADD1A	**Page: 0x0%00** **where:** **% = 2×(DDC channel #)+1**	**Address: 0x09**	**DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING**

BIT 15							BIT 8
phase_add_a(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
phase_add_a(23:16)							
0	0	0	0	0	0	0	0

phase_add_a(31:16) : This 32 bit word is used to control the frequency of the NCO. This value is added to the frequency accumulator every clock cycle *(UMTS mode and A channel in CDMA mode)*.

3.4.5.11 PHASEADD0B Register

Register name: PHASEADD0	**Page: 0x0%00** **where:** **% = 2×(DDC channel #)+1**	**Address: 0x0A**	**DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING**

BIT 15							BIT 8
phase_add_b(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
phase_add_b(7:0)							
0	0	0	0	0	0	0	0

phase_add_b(15:0) : This 32 bit word is used to control the frequency of the NCO. This value is added to the frequency accumulator every clock cycle *(B channel in CDMA mode)*.

3.4.5.12 PHASEADD1B Register

Register name: PHASEADD1B	Page: 0x0%00 where: % = 2×(DDC channel #)+1	Address: 0x0B	DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING

BIT 15							BIT 8
phase_add_b(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
phase_add_b(23:16)							
0	0	0	0	0	0	0	0

phase_add_b(31:16) : This 32 bit word is used to control the frequency of the NCO. This value is added to the frequency accumulator every clock cycle *(B channel in CDMA mode).*

3.4.5.13 PHASE_OFFSETA Register

Register name: PHASE_OFFSETA	Page: 0x0%00 where: % = 2×(DDC channel #)+1	Address: 0x0C	DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING

BIT 15							BIT 8
phase_offset_a(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
phase_offset_a(7:0)							
0	0	0	0	0	0	0	0

phase_offset_a(15:0) : 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).*

3.4.5.14 PHASE_OFFSETB Register

Register name: PHASE_OFFSETB	Page: 0x0%00 where: % = 2×(DDC channel #)+1	Address: 0x0D	DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING

BIT 15							BIT 8
phase_offset_b(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
phase_offset_b(7:0)							
0	0	0	0	0	0	0	0

phase_offset_b(15:0) : This is the fixed phase offset added to the output of the frequency accumulator for sinusoid generation in the NCO. *(B channel in CDMA mode)*

3.4.5.15 CONFIG1 Register

Register name: CONFIG1

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x0E

BIT 15 BIT 8

dither_ena	dither_mask(1:0)		pmeter_ sync_ disable	ddc_ena	muxed _data	mixer_gain	mpu_ram_read
0	0	0	0	0	0	0	0

BIT 7 BIT 0

unused	unused	unused	unused	unused	zero_ qsample	mux_pos	mux_factor
0	0	0	0	0	0	0	0

dither_ena : This bit controls whether dither is turned on(1) or off(0).

dither_mask(1) : This bit controls the MASKing of the dither word’s MSB. (1= MASKed, 0=used in dither word)

dither_mask(0) : This bit controls the MASKing of the dither word’s MSB-1. (1= MASKed, 0=used in dither word)

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 available.

ddc_ena : When set this turns on the DDC. When cleared, the clocks to this block are turned off. For the DDC blocks used as the second half in the long PFIR configuration, this bit should be cleared.

muxed_data : When asserted the DDC mux block assumes that multiple channels are muxed together on one input data stream. *For factory use only*.

For a 2X muxed stream it would look like: Sa0, Sb0, Sa1, Sb1, Sa2, Sb2 …. etc...

mixer_gain : Adds a fixed 6 dB of gain to the mixer output(before round and limiting) when asserted.

mpu_ram_read : (TESTING PURPOSES) Allows the coefficient RAMs in the PFIR/CFIR to be read out the mpu data bus. Unfortunately, this cannot be done during normal operation and must be done when the state of the output data is not important. **THIS BIT MUST ONLY BE SET DURING THE MPU READ OPERATION AND MUST BE CLEARED FOR NORMAL DDC OPERATION.**

zero_qsample : When asserted, the Q sample into the mixer is held to zero. For UMTS mode at any input rate, and CDMA mode with input rates of rxclk/2 or lower, this bit must be set for real only input data mode (also for muxed input data stream modes). For real only inputs at the full rxclk rate in CDMA mode, the remix_only bit must be set in the DDCCONFIG1 register.

mux_pos : These bits set the position for selection in the muxed data stream. This value must be less than or equal to the mux_factor bits.

mux_factor : These two bits set the number of channels in the data stream. 0=1 stream, 1=2 streams. The ch_rate_sel bits for the DDC should be programmed to rxclk/2 for the 2 streams mode.

3.4.5.16 CONFIG2 Register

Register name: CONFIG2 **Page: 0x0%00**
where:
% = 2×(DDC channel #)+1 **Address: 0x0F**

BIT 15							BIT 8
unused	unused	unused	unused	unused	unused	unused	unused
0	0	0	0	0	0	0	0

BIT 7							BIT 0
unused	unused	ddc_tst_sel(5:0)					
0	0	0	0	0	0	0	0

ddc_tst_sel(5:0) : 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 tst_blk from changing. It does not stop the clock however. The 36 bits for the testbus are routed to the rxin_c, rxin_d, dvga_c and dvga_d pins on the chip.

SYNC on dvga_c(0) **AFLAG on dvga_d(5)**	**ddc_tst_sel(5:0)**	**Data selected for output (36 bits total)** **rxin_d(15:0), dvga_c(3:2), rxin_c(15:0), dvga_c(5:4)**
N	000000	constant 0
Y	000001	pfir output – (35:18) I and (17:0) Q
Y	000010	cfir output – (35:18) I and (17:0) Q
N	000011	tadj A output – (35:18) I and (17:0) Q
N	000100	tadj B output – (35:18) I and (17:0) Q
N	000101	nco SINE output – (35:20) zeroed (19:0) SINE
N	000110	nco COSINE output – (35:20) zeroed (19:0) COSINE
N	000111	cic output – (35:18) I and (17:0) Q
Y	001000	agc output – (35:11) I and (10:0) Q {full 25b I result and upper 11b Q result}
N	001001	mix A output – (35:18) i*cos-q*sin and (17:0) i*sin+q*cos
N	001010	mix B output – (35:18) i*cos-q*sin and (17:0) i*sin+q*cos
N	001011	DDC MUX A output (35:18) I and (17:0) Q
N	001100	DDC MUX B output (35:18) I and (17:0) Q

3.4.5.17 AGC_CONFIG1 Register

Register name: AGC_CONFIG1

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x10

BIT 15 BIT 8

agc_dblw(3:0)				agc_dabv(3:0)			
0	0	0	0	0	0	0	0

BIT 7 BIT 0

agc_dzro(3:0)				agc_dsat(3:0)			
0	0	0	0	0	0	0	0

agc_dblw(3:0) : The value to shift the gain that is then added to the accumulator when the value of the incoming data * current gain value is **below** the Threshold.

agc_dabv(3:0) : The value to shift the gain that is then subtracted from the accumulator when the value of the incoming data * the current gain value is **above** the Threshold.

agc_dzro(3:0) : The value to shift the gain that is then added to the accumulator when the value of the incoming data * current gain values **consistently equal to zero**. (Usually a smaller number than agc_dblw).

agc_dsat(3:0) : The value to shift the gain that is then subtracted form the accumulator when the value of the incoming data * the current gain value is **consistently equal to maximum (saturation)**.

NOTE: The larger the number in the above words, the smaller the step size. The above values control the AGC gain shifting (range is from 3 to 18).

3.4.5.18 AGC_CONFIG2 Register

Register name: AGC_CONFIG2

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x11

BIT 15 BIT 8

zero_msk(3:0)				agc_rnd(3:0)			
0	0	0	0	0	0	0	0

BIT 7 BIT 0

agc_thresh(7:0)							
0	0	0	0	0	0	0	0

zero_msk(3:0) : Masks the lower 4 bits of the magnitude of the input signal so that they are counted as zeros.

agc_rnd(3:0) : Determines where to round the output of the AGC; the number of bits output is (18 – agc_rnd). For example, 0000 is 18 bits.

agc_thresh(7:0) : Threshold for (input * gain) comparison. This value is compared to the magnitude of the upper eight bits of the agc output.

3.4.5.19 AGC_CONFIG3 Register

Register name: AGC_CONFIG3	**Page: 0x0%00** **where:** **% = 2×(DDC channel #)+1**	**Address: 0x12**

BIT 15							BIT 8
unused	unused	unused	agc_ freeze	agc_max_cnt(3:0)			
0	0	0	0	0	0	0	0

BIT 7							BIT 0
unused	unused	unused	agc_ clear	agc_zero_cnt(3:0)			
0	0	0	0	0	0	0	0

agc_freeze : Freezes the agc when set. This should be asserted when the AGC algorithm is bypassed or held constant.

agc_max_cnt(3:0) : When the agc_output (input * gain) is at full scale for this number of samples, then the gain shift value is changed to agc_dsat.

agc_clear : Clears the AGC accumulator when set. Assert this when the AGC is in bypass mode.

agc_zero_cnt(3:0) : when the agc_output (input * gain) is zero value for this number of samples, then the gain shift value is changed to agc_dzro.

3.4.5.20 AGC_GAINMSB Register

Register name: **AGC_GAINMSB**	**Page: 0x0%00** **where:** **% = 2×(DDC channel #)+1**	**Address: 0x13**	**DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING**

BIT 15							BIT 8
agc_gaina(23:16)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
agc_gainb(23:16)							
0	0	0	0	0	0	0	0

agc_gaina(23:16) : MSBs of the agc_gaina word.

agc_gainb(23:16) : MSBs of the agc_gainb word.

3.4.5.21 AGC_GAINA Register

Register name: AGC_GAINA **Page: 0x0%00 where: % = 2×(DDC channel #)+1** **Address: 0x14** **DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING**

BIT 15 BIT 8

agc_gaina(15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

agc_gaina(7:0)							
0	0	0	0	0	0	0	0

agc_gaina(15:0) : This is the lower 16 bits of the total 24 bits of programmable gain. The gain value is always positive with the upper 12 bits being the integer value and the lower 12 bits being the fractional. This gain value is used for all UMTS operations and for A channel data when in CDMA mode. A 24-bit value of 00000000001.000000000000 is unity gain.

3.4.5.22 AGC_GAINB Register

Register name: AGC_GAINB **Page: 0x0%00 where: % = 2×(DDC channel #)+1** **Address: 0x15** **DOUBLE BUFFERED, REQUIRES SYNC FOR LOADING**

BIT 15 BIT 8

agc_gainb(15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

agc_gainb(7:0)							
0	0	0	0	0	0	0	0

agc_gainb(15:0) : This is the lower 16 of the total of 24 bit of programmable gain. The gain value is always positive with the upper 12 bits being the integer value and the lower 12 bits being the fractional. This gain value is used for B channel data when in CDMA. A 24-bit value of 00000000001.000000000000 is unity gain.

3.4.5.23 AGC_AMAX Register

Register name: AGC_AMAX **Page: 0x0%00**
where:
% = 2×(DDC channel #)+1 **Address: 0x16**

BIT 15 BIT 8

agc_amax(15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

agc_amax(7:0)							
0	0	0	0	0	0	0	0

agc_amax(15:0) : This is the maximum gaina or gainb can be adjusted up. The value programmed is a positive value that is used to generate the most positive AGC gain adjust. For example, if 512 is programmed, the maximum gain will be the programmed gain (AGC_GAINA/B) + 512.

3.4.5.24 AGC_AMIN Register

Register name: AGC_AMIN **Page: 0x0%00**
where:
% = 2×(DDC channel #)+1 **Address: 0x17**

BIT 15 BIT 8

agc_amin(15:8)							
0	0	0	0	0	0	0	0

BIT 7 BIT 0

agc_amin(7:0)							
0	0	0	0	0	0	0	0

agc_amin(15:0) : This is the minimum gaina or gainb can be adjusted down. The value programmed is a positive value that is inverted internally to generate the most negative AGC gain adjust. For example, if 512 is programmed, the minimum gain will be the programmed gain (AGC_GAINA/B) – 512.

3.4.5.25 PSER_CONFIG1 Register

Register name: PSER_CONFIG1

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x18

BIT 15							BIT 8
unused	pser_recv_fsinvl(6:0)						
0	0	0	0	0	0	0	0

BIT 7							BIT 0
unused	unused	unused	pser_recv_bits(4:0)				
0	0	0	0	0	0	0	0

pser_recv_fsinvl(6:0) : Receive serial interface frame sync interval in bit clocks.

pser_recv_bits(4:0) : Number of output bits per sample-1; for 18 bits, this is set to {10001}.

3.4.5.26 PSER_CONFIG2 Register

Register name: PSER_CONFIG2

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x19

BIT 15							BIT 8
pser_recv_clkdiv(3:0)				unused	unused	unused	unused
0	0	0	0	0	0	0	0

BIT 7							BIT 0
pser_recv_8pin	pser_recv_alt	unused	unused	unused	unused	pser_recv_fsdel(1:0)	
0	0	0	0	0	0	0	0

pser_recv_clkdiv(3:0) : 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 GC5018 clock, set pser_recv_clkdiv(3:0) = 0011.

pser_recv_8pin : When set, 4 pins are used for I and 4 pins for Q in UMTS mode. When cleared, 2 pins are used for I and 2 pins for Q. This is used in combination with the pser_recv_alt bit. When this bit is set, it would be set in 2 adjacent DDC channels; one would also set the pser_recv_alt bit in the adjacent DDC. This will cause the I channel to be serialized on 4 pins and the Q channel to be serialized on the adjacent channels 4 pins.

pser_recv_alt : When set, this channel's receive serial interface will output the Q data from the adjacent DDC channel.

pser_recv_fsdel(1:0) : Delay between the receive frame sync output and the MSB of serial data {3,2,1,0}. This number is in serial output bit times, not rxclk periods.

3.4.5.27 DDCCONFIG1 Register

Register name: DDCCONFIG1

Page: 0x0%00
where:
% = 2×(DDC channel #)+1

Address: 0x1A

BIT 15							BIT 8
ddcmux_sel_a (3:0)				agc_rnd_ disable	gain_mon	ch_rate_sel(1:0)	
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ddcmux_sel_b(3:0)				remix_only	cic_ bypass	double_tap(1:0)	
0	0	0	0	0	0	0	0

ddcmux_sel_X(3:0) : Controls which samples go to the mixer for I/Q. Since in CDMA there are two streams, an A and B stream, two mux select values are used.

Select Value	I data from X input	Q data from X input
0000	RXINA	RXINA
0001	RXINB	RXINB
0010	RXINC	RXINC
0011	RXIND	RXIND
0100	RXINA	RXINB
0101	RXINA	RXINC
0110	RXINA	RXIND
0111	RXINB	RXINA
1000	RXINB	RXINC
1001	RXINB	RXIND
1010	RXINC	RXINA
1011	RXINC	RXINB
1100	RXINC	RXIND
1101	RXIND	RXINA
1110	RXIND	RXINB
1111	RXIND	RXINC

agc_rnd_disable : When set, the agc_rnd bits have no effect. The whole 29 bits are used in the rounding and the round bit is bit4.

gain_mon : Combines the gain with the I/Q output signals when asserted.

OUTPUT	Bits(17:10)	Bits(9:4)	Bits(3:2)	Bits(1:0)
I	Gained I value	Gain(18:11)		"00"
Q	Gained Q value	Gain(10:5)	Shift status(1:0)	"00"

ch_rate_sel(1:0) : Sets the DDC channel input data rate. The value set here should match the value in the Receive Input Interface rate select bits (rate_sel).

ch_rate_sel	Input data rate
00	rxclk
01	rxclk/2
10	rxclk/4
11	rxclk/8

When muxed_data is set (Factory Use Only) rate_sel should be set to rxclk "00" and ch_rate_sel should be set to rxclk/2 "01".

remix_only : Assert this when real only, full rxclk rate input data is used in CDMA mode. The signal on the Q bus selected by the ddcmux_sel_X(3:0) bits above is ignored (functions as if the Q data is 0).

cic_bypass : *Factory Use Only.* If asserted then the data from the rxin_a(15:0) and rxin_b(15:0) 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.

ONLY DDC0, DDC2, DDC4 and DDC6 can be the UMTS double tap (64 to 128 tap) PFIR Mode. DDC1, DDC3, DDC5 and DDC7 PFIRs are used to lengthen the DDC0, DDC2, DDC4 and DDC6 PFIRs.

double_tap(1) : When set, the DDC is in double length PFIR mode which sends the data out of the last PFIR sample ram in this DDC (DDC0, DDC2, DDC4, DDC6) to the adjacent secondary DDC (DDC1, DDC3, DDC5, DDC7) PFIR forming a 128-tap delay line. Output data received from the adjacent secondary DDC PFIR summer is added into the Main DDC's PFIR sum to form the final output.

double_tap(0) : When set, the PFIR input comes from the adjacent(Main) PFIR. When cleared, PFIR input is from the CFIR connected directly to this PFIR. **Only valid in DDC1, DDC3, DDC5 and DDC7. The ddc_ena bit in the CONFIG1 register should be cleared for the DDC1, DDC3, DDC5 and DDC7 when double_tap(0) is set.**

NOTE: to put 2 DDCs in to 128 tap mode:

Program DDC0/DDC2/DDC4/DDC6 double_tap(1:0) to "10" and ddc_ena to "1".

Program DDC1/DDC3/DDC5/DDC7 double_tap(1:0) to "01" and ddc_ena to "0".

3.4.5.28 SYNC_0 Register

Register name: SYNC_0 **Page: 0x0%00**
where:
% = 2×(DDC channel #)+1 **Address: 0x1B**

BIT 15							BIT 8
unused	ssel_cic(2:0)			unused	ssel_pmeter(2:0)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
unused	ssel_agc_freeze(2:0)			unused	ssel_serial(2:0)		
0	1	1	0	0	0	0	0

ssel_cic(2:0) : Selects the sync source for the DDC CIC filter, thus setting the decimation moment.

ssel_pmeter(2:0) : Selects the sync source for the channel power meter.

ssel_agc_freeze(2:0) : Selects the sync that is used to hold the AGC in freeze mode. With this functionality the user can program the AGC freeze control to look at the state of an input sync, or the one shots. It defaults to being off or not looking at any syncs and not driving the freeze control. This way, upon startup, the chip looks at the MPU register bit for AGC freezing and not the syncs.

ssel_serial(2:0) : Selects the sync source for the DDC serial interface state machines.

Sync sources are contained in this and many of the following registers. For all sync source selections:

ssel_XXXX(2:0)	Selected sync source for DDC
000	rxsyncA
001	rxsyncB
010	rxsyncC
011	rxsyncD
100	DDC sync counter
101	one shot (register write triggered)
110	always 0
111	always 1

3.4.5.29 SYNC_1 Register

Register name: SYNC_1 **Page: 0x0%00**
where:
% = 2×(DDC channel #)+1 **Address: 0x1C**

BIT 15							BIT 8
unused	ssel_tadj_fine(2:0)			unused	ssel_tadj_reg(2:0)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
unused	ssel_gain(2:0)			unused	ssel_ddc_agc(2:0)		
0	0	0	0	0	0	0	0

ssel_tadj_fine(2:0) : Selects the sync source for the fine time adjust zero stuff moment.

ssel_tadj_reg(2:0) : Selects the sync source for the fine and coarse time adjust register updates.

ssel_gain(2:0) : Selects the sync source for the DDC AGC gain register.

ssel_ddc_agc(2:0) : Selects the sync source to initialize the AGC, primarily for test purposes.

3.4.5.30 SYNC_2 Register

Register name: SYNC_2 **Page: 0x0%00**
where:
% = 2×(DDC channel #)+1 **Address: 0x1D**

BIT 15							BIT 8
unused	ssel_nco(2:0)			unused	ssel_dither(2:0)		
0	0	0	0	0	0	0	0

BIT 7							BIT 0
unused	ssel_freq(2:0)			unused	ssel_phase (2:0)		
0	0	0	0	0	0	0	0

ssel_nco : Selects the sync source for the NCO accumulator reset.

ssel_dither : Selects the sync source for the NCO phase dither generator reset.

ssel_freq : Selects the sync source for the NCO frequency register.

ssel_phase : Selects the sync source for the NCO phase offset register.

3.4.5.31 DDC_CHK_SUM Register

Register name: DDC_CHK_SUM **Page: 0x0%20**
where:
% = 2×(DDC channel #)+1 **Address: 0x20** **READ ONLY**

BIT 15							BIT 8
ddc_chk_sum(15:0)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
ddc_chk_sum(7:0)							
0	0	0	0	0	0	0	0

ddc_chk_sum : The DDC self test checksum value

3.4.5.32 PMETER_RESULT_A_LSB Register

Register name: PMETER_RESULT_A_LSB **Page: 0x0%20**
where:
% = 2×(DDC channel #)+1 **Address: 0x21** **READ ONLY**

BIT 15							BIT 8
pmeter_result_a(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
pmeter_result_a(7:0)							
0	0	0	0	0	0	0	0

pmeter_result_a(15:0) : Lower 16 bits of the UMTS mode or CDMA mode A channel power measurement.

3.4.5.33 PMETER_RESULT_A_MID Register

Register name: PMETER_RESULT_A_MID	Page: 0x0%20 where: % = 2×(DDC channel #)+1	Address: 0x22	READ ONLY

BIT 15							BIT 8
pmeter_result_a(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
pmeter_result_a(23:16)							
0	0	0	0	0	0	0	0

pmeter_result_a(31:16) : Mid 16 bits of the UMTS mode or CDMA mode A channel power measurement.

3.4.5.34 PMETER_RESULT_A_MSB Register

Register name: PMETER_RESULT_A_MSB	Page: 0x0%20 where: % = 2×(DDC channel #)+1	Address: 0x23	READ ONLY

BIT 15							BIT 8
pmeter_result_a(47:40)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
pmeter_result_a(39:32)							
0	0	0	0	0	0	0	0

pmeter_result_a(47:32) : Upper mid 16 bits of the UMTS mode or CDMA mode A channel power measurement.

3.4.5.35 PMETER_RESULT_B_LSB Register

Register name: PMETER_RESULT_B_LSB	Page: 0x0%20 where: % = 2×(DDC channel #)+1	Address: 0x24	READ ONLY

BIT 15							BIT 8
pmeter_result_b(15:8)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
pmeter_result_b(7:0)							
0	0	0	0	0	0	0	0

pmeter_result_b(15:0) : Lower 16 bits of the CDMA mode B channel power measurement

3.4.5.36 PMETER_RESULT_B_MID Register

Register name: PMETER_RESULT_B_MID **Page: 0x0%20**
where:
% = 2×(DDC channel #)+1 **Address: 0x25** **READ ONLY**

BIT 15							BIT 8
pmeter_result_b(31:24)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
pmeter_result_b(23:16)							
0	0	0	0	0	0	0	0

pmeter_result_b(31:16) : Mid 16 bits of the CDMA mode B channel power measurement.

3.4.5.37 PMETER_RESULT_B_MSB Register

Register name: PMETER_RESULT_B_MSB **Page: 0x0%20**
where:
% = 2×(DDC channel #)+1 **Address: 0x26** **READ ONLY**

BIT 15							BIT 8
pmeter_result_b(47:40)							
0	0	0	0	0	0	0	0

BIT 7							BIT 0
pmeter_result_b(39:32)							
0	0	0	0	0	0	0	0

pmeter_result_b(47:32) : Upper mid 16 bits of the CDMA mode B channel power measurement.

3.4.5.38 PMETER_RESULT_AB_UMSB Register

Register name: PMETER_RESULT_AB_UMSB	Page: 0x0%20 where: % = 2×(DDC channel #)+1	Address: 0x27	READ ONLY

BIT 15 — BIT 8

pmeter_result_a(54:48)							
0	0	0	0	0	0	0	0

BIT 7 — BIT 0

pmeter_result_b(54:48)							
0	0	0	0	0	0	0	0

pmeter_result_a(54:48) : Most Significant 7 bits of the 55-bit UMTS or CDMA mode A channel power measurement.

pmeter_result_b(54:48) : Most Significant 7 bits of the 55-bit CDMA mode B channel power measurement.

4 GC5018 PINS

4.1 Digital Receive Section Signals

Signal Name	Ball	Type	Description
rxclk	U8	input	receive digital section clock input
adcclk_a	B10	input	rxin_a_x input clock
adcclk_b	A10	input	rxin_b_x input clock
adcclk_c	F2	input	rxin_c_x input clock
adcclk_d	E4	input	rxin_d_x input clock
rxin_a_ovr	B16	input	adc overflow/overrange bit for rxin_a
rxin_b_ovr	C9	input	adc overflow/overrange bit for rxin_b
rxin_c_ovr	A3	input	adc overflow/overrange bit for rxin_c
rxin_d_ovr	E1	input	adc overflow/overrange bit for rxin_d
dvga_a_5	A17	output	Digital VGA control output for ADC0 MSB
dvga_a_4	B17	output	Digital VGA control output for ADC0
dvga_a_3	C16	output	Digital VGA control output for ADC0
dvga_a_2	C17	output	Digital VGA control output for ADC0
dvga_a_1	D16	output	Digital VGA control output for ADC0
dvga_a_0	D17	output	Digital VGA control output for ADC0 LSB
dvga_b_5	B18	output	Digital VGA control output for ADC1 MSB
dvga_b_4	E16	output	Digital VGA control output for ADC1
dvga_b_3	E17	output	Digital VGA control output for ADC1
dvga_b_2	C18	output	Digital VGA control output for ADC1
dvga_b_1	D15	output	Digital VGA control output for ADC1
dvga_b_0	D18	output	Digital VGA control output for ADC1 LSB
dvga_c_5	M1	output	Digital VGA control output for rxin_c MSB, test bus bit 1
dvga_c_4	L2	output	Digital VGA control output for rxin_c, test bus bit 0
dvga_c_3	L3	output	Digital VGA control output for rxin_c, test bus bit 19
dvga_c_2	M4	output	Digital VGA control output for rxin_c, test bus bit 18
dvga_c_1	N4	output	Digital VGA control output for rxin_c, test bus CLK
dvga_c_0	M2	output	Digital VGA control output for rxin_c LSB, test bus SYNC
dvga_d_5	M3	output	Digital VGA control output for rxin_d MSB, test bus AFLAG
dvga_d_4	P1	output	Digital VGA control output for rxin_d
dvga_d_3	P4	output	Digital VGA control output for rxin_d
dvga_d_2	N2	output	Digital VGA control output for rxin_d
dvga_d_1	R1	output	Digital VGA control output for rxin_d
dvga_d_0	N3	output	Digital VGA control output for rxin_d LSB
rxin_a_15	C15	input	receive input data bus a bit 15 (MSB)
rxin_a_14	B15	input	receive input data bus a
rxin_a_13	C14	input	receive input data bus a
rxin_a_12	B14	input	receive input data bus a
rxin_a_11	A16	input	receive input data bus a
rxin_a_10	A15	input	receive input data bus a
rxin_a_9	C13	input	receive input data bus a
rxin_a_8	B13	input	receive input data bus a
rxin_a_7	A14	input	receive input data bus a
rxin_a_6	C12	input	receive input data bus a

Signal Name	Ball	Type	Description
rxin_a_5	B12	input	receive input data bus a
rxin_a_4	A12	input	receive input data bus a
rxin_a_3	C11	input	receive input data bus a
rxin_a_2	B11	input	receive input data bus a
rxin_a_1	D11	input	receive input data bus a
rxin_a_0	C10	input	receive input data bus a bit 0 (LSB)
rxin_b_15	B9	input	receive input data bus b bit 15 (MSB)
rxin_b_14	D9	input	receive input data bus b
rxin_b_13	A9	input	receive input data bus b
rxin_b_12	C8	input	receive input data bus b
rxin_b_11	B8	input	receive input data bus b
rxin_b_10	D8	input	receive input data bus b
rxin_b_9	C7	input	receive input data bus b
rxin_b_8	B7	input	receive input data bus b
rxin_b_7	A7	input	receive input data bus b
rxin_b_6	B6	input	receive input data bus b
rxin_b_5	C6	input	receive input data bus b
rxin_b_4	A5	input	receive input data bus b
rxin_b_3	B5	input	receive input data bus b
rxin_b_2	C5	input	receive input data bus b
rxin_b_1	A4	input	receive input data bus b
rxin_b_0	B4	input	receive input data bus b bit 0 (LSB)
rxin_c_15	A2	input/output	receive input data bus c bit 15 (MSB), test bus bit 17
rxin_c_14	B3	input/output	receive input data bus c bit 14, test bus bit 16
rxin_c_13	B2	input/output	receive input data bus c bit 13, test bus bit 15
rxin_c_12	C3	input/output	receive input data bus c bit 12, test bus bit 14
rxin_c_11	C2	input/output	receive input data bus c bit 11, test bus bit 13
rxin_c_10	A1	input/output	receive input data bus c bit 10, test bus bit 12
rxin_c_9	D3	input/output	receive input data bus c bit 9, test bus bit 11
rxin_c_8	D2	input/output	receive input data bus c bit 8, test bus bit 10
rxin_c_7	B1	input/output	receive input data bus c bit 7, test bus bit 9
rxin_c_6	C4	input/output	receive input data bus c bit 6, test bus bit 8
rxin_c_5	E3	input/output	receive input data bus c bit 5, test bus bit 7
rxin_c_4	C1	input/output	receive input data bus c bit 4, test bus bit 6
rxin_c_3	E2	input/output	receive input data bus c bit 3, test bus bit 5
rxin_c_2	D4	input/output	receive input data bus c bit 2, test bus bit 4
rxin_c_1	D1	input/output	receive input data bus c bit 1, test bus bit 3
rxin_c_0	F3	input/output	receive input data bus c bit 0 (LSB), test bus bit 2
rxin_d_15	G3	input/output	receive input data bus d bit 15 (MSB), test bus bit 35
rxin_d_14	G2	input/output	receive input data bus d bit 14, test bus bit 34
rxin_d_13	F4	input/output	receive input data bus d bit 13, test bus bit 33
rxin_d_12	G4	input/output	receive input data bus d bit 12, test bus bit 32
rxin_d_11	G1	input/output	receive input data bus d bit 11, test bus bit 31
rxin_d_10	H3	input/output	receive input data bus d bit 10, test bus bit 30
rxin_d_9	H2	input/output	receive input data bus d bit 9, test bus bit 29
rxin_d_8	H4	input/output	receive input data bus d bit 8, test bus bit 28
rxin_d_7	J3	input/output	receive input data bus d bit 7, test bus bit 27
rxin_d_6	J2	input/output	receive input data bus d bit 6, test bus bit 26

Signal Name	Ball	Type	Description
rxin_d_5	J1	input/output	receive input data bus d bit 5, test bus bit 25
rxin_d_4	K1	input/output	receive input data bus d bit 4, test bus bit 24
rxin_d_3	K2	input/output	receive input data bus d bit 3, test bus bit 23
rxin_d_2	K3	input/output	receive input data bus d bit 2, test bus bit 22
rxin_d_1	K4	input/output	receive input data bus d bit 1, test bus bit 21
rxin_d_0	L4	input/output	receive input data bus d bit 0 (LSB), test bus bit 20
rx_synca	T8	input	receive sync input
rx_syncb	V10	input	receive sync input
rx_syncc	R10	input	receive sync input
rx_syncd	U9	input	receive sync input
rx_sync_out	U16	output	receive general purpose output sync
rxclk_out	U17	output	receive clock output
rx_sync_out_7	U15	output	receive serial interface frame strobe for rxout_7_x
rx_sync_out_6	T18	output	receive serial interface frame strobe for rxout_6_x, frame strobe (rx_sync_out signal) for parallel interface.
rx_sync_out_5	P15	output	receive serial interface frame strobe for rxout_5_x
rx_sync_out_4	M15	output	receive serial interface frame strobe for rxout_4_x
rx_sync_out_3	K16	output	receive serial interface frame strobe for rxout_3_x
rx_sync_out_2	J16	output	receive serial interface frame strobe for rxout_2_x
rx_sync_out_1	G15	output	receive serial interface frame strobe for rxout_1_x
rx_sync_out_0	E15	output	receive serial interface frame strobe for rxout_0_x
rxout_7_a	R16	output	DDC 7 serial out data. CDMA A: I data UMTS: Imsb DDC Parallel Interface I(12)
rxout_7_b	R17	output	DDC 7 serial out data. CDMA B: I data UMTS: Imsb – 1 DDC Parallel Interface I(13)
rxout_7_c	U18	output	DDC 7 serial out data. CDMA A: Q data UMTS: Qmsb DDC Parallel Interface I(14)
rxout_7_d	P16	output	DDC 7 serial out data. CDMA B: Q data UMTS: Qmsb –1 DDC Parallel Interface I(15)
rxout_6_a	P17	output	DDC 6 serial out data. CDMA A: I data UMTS: Imsb DDC Parallel Interface I(8)
rxout_6_b	T15	output	DDC 6 serial out data. CDMA B: I data UMTS: Imsb – 1 DDC Parallel Interface I(9)
rxout_6_c	R15	output	DDC 6 serial out data. CDMA A: Q data UMTS: Qmsb DDC Parallel Interface I(10)
rxout_6_d	N16	output	DDC 6 serial out data. CDMA B: Q data UMTS: Qmsb –1 DDC Parallel Interface I(11)
rxout_5_a	N17	output	DDC 5 serial out data. CDMA A: I data UMTS: Imsb Parallel Interface I(4)
rxout_5_b	R18	output	DDC 5 serial out data. CDMA B: I data UMTS: Imsb – 1 Parallel Interface I(5)
rxout_5_c	P18	output	DDC 5 serial out data. CDMA A: Q data UMTS: Qmsb Parallel Interface I(6)
rxout_5_d	M16	output	DDC 5 serial out data. CDMA B: Q data UMTS: Qmsb –1 Parallel Interface I(7)
rxout_4_a	M17	output	DDC 4 serial out data. CDMA A: I data UMTS: Imsb Parallel Interface I(0)
rxout_4_b	N15	output	DDC 4 serial out data. CDMA B: I data UMTS: Imsb – 1 Parallel Interface I(1)
rxout_4_c	L16	output	DDC 4 serial out data. CDMA A: Q data UMTS: Qmsb Parallel Interface I(2)
rxout_4_d	L17	output	DDC 4 serial out data. CDMA B: Q data UMTS: Qmsb –1 Parallel Interface I(3)
rxout_3_a	M18	output	DDC 3 serial out data. CDMA A: I data UMTS: Imsb Parallel Interface Q(12)
rxout_3_b	L15	output	DDC 3 serial out data. CDMA B: I data UMTS: Imsb – 1 Parallel Interface Q(13)
rxout_3_c	K17	output	DDC 3 serial out data. CDMA A: Q data UMTS: Qmsb Parallel Interface Q(14)
rxout_3_d	K18	output	DDC 3 serial out data. CDMA B: Q data UMTS: Qmsb –1 Parallel Interface Q(15)
rxout_2_a	J18	output	DDC 2 serial out data. CDMA A: I data UMTS: Imsb Parallel Interface Q(8)
rxout_2_b	J17	output	DDC 2 serial out data. CDMA B: I data UMTS: Imsb – 1 Parallel Interface Q(9)
rxout_2_c	H15	output	DDC 2 serial out data. CDMA A: Q data UMTS: Qmsb Parallel Interface Q(10)
rxout_2_d	G18	output	DDC 2 serial out data. CDMA B: Q data UMTS: Qmsb –1 Parallel Interface Q(11)
rxout_1_a	H17	output	DDC 1 serial out data. CDMA A: I data UMTS: Imsb Parallel Interface Q(4)

Signal Name	Ball	Type	Description
rxout_1_b	H16	output	DDC 1 serial out data. CDMA B: I data UMTS: Imsb – 1 Parallel Interface Q(5)
rxout_1_c	F15	output	DDC 1 serial out data. CDMA A: Q data UMTS: Qmsb Parallel Interface Q(6)
rxout_1_d	G17	output	DDC 1 serial out data. CDMA B: Q data UMTS: Qmsb –1 Parallel Interface Q(7)
rxout_0_a	G16	output	DDC 0 serial out data. CDMA A: I data UMTS: Imsb Parallel Interface Q(0)
rxout_0_b	E18	output	DDC 0 serial out data. CDMA B: I data UMTS: Imsb – 1 Parallel Interface Q(1)
rxout_0_c	F17	output	DDC 0 serial out data. CDMA A: Q data UMTS: Qmsb Parallel Interface Q(2)
rxout_0_d	F16	output	DDC 0 serial out data. CDMA B: Q data UMTS: Qmsb –1 Parallel Interface Q(3)

4.2 Microprocessor Signals

Signal Name	Ball	Type	Description
d0	V3	input/output	MPU register interface data bus bit 0 (LSB)
d1	U3	input/output	MPU register interface data bus
d2	V2	input/output	MPU register interface data bus
d3	U2	input/output	MPU register interface data bus
d4	T3	input/output	MPU register interface data bus
d5	T2	input/output	MPU register interface data bus
d6	V4	input/output	MPU register interface data bus
d7	R3	input/output	MPU register interface data bus
d8	U4	input/output	MPU register interface data bus
d9	R2	input/output	MPU register interface data bus
d10	U1	input/output	MPU register interface data bus
d11	P3	input/output	MPU register interface data bus
d12	T4	input/output	MPU register interface data bus
d13	T1	input/output	MPU register interface data bus
d14	P2	input/output	MPU register interface data bus
d15	R4	input/output	MPU register interface data bus bit 15 (MSB)
a0	V7	input	MPU register interface address bus bit 0 (LSB)
a1	T6	input	MPU register interface address bus
a2	U6	input	MPU register interface address bus
a3	V5	input	MPU register interface address bus
a4	T5	input	MPU register interface address bus
a5	U5	input	MPU register interface address bus bit 5 (MSB)
rd_n	T7	input	MPU register interface read – active low
wr_n	V9	input	MPU register interface write – active low
ce_n	U7	input	MPU register interface chip enable – active low
reset_n	R9	input	chip reset – active low
interrupt	T9	output	chip interrupt

4.3 JTAG Signals

Signal Name	Ball	Type	Description
tdi	U11	input	JTAG test data in
tms	T11	input	JTAG test mode select
trst_n	U12	input	JTAG test reset (same as trst; the “_n” is for consistency - being active low) *Note: the trst_n pin should be asserted low after power up to insure the JTAG logic is properly initialized.*
tck	V12	input	JTAG test clock
tdo	U10	output	JTAG test data out

4.4 Factory Test and No Connect Signals

Signal Name	Ball	Type	Description
testmode0	U14	input	Do not connect; internal pull down
testmode1	T13	input	Do not connect; internal pull down
scanen	U13	input	Do not connect; internal pull down
fa002_scan	V14	input	Do not connect; internal pull down
fa002_clk	V15	input	Do not connect; internal pull down
fa002_out	T12	output	Do not connect
zero	T14	input	Do not connect; internal pull down
fuse_out	V16	output	Do not connect

4.5 Power and Ground Signals

Signal Name	Ball	Description
VDDS	A6, D5, D6, D10, D13, D14, E5, E13, E14, F1, F5, F14, F18, J4, J15, K15, L5, M5, N5, N14, P5, P13, P14, R5, R6, R7, R8, R12, R13, R14, T16, V6, V17	Digital I/O Power (3.3 V), also called Vpad
DVDD	A13, D7, D12, E6, E7, E8, E10, E11, E12, G5, G14, H5, H14, J5, J14, L14, M14, N1, N18, P6, P7, P8, P10, P11, P12, V13	Digital Core Power (1.5 V), also called Vcore
DVSS	A8, A11, E9, F6, F7, F8, F9, F10, F11, F12, F13, G6, G7, G8, G9, G10, G11, G12, G13, H1, H6, H7, H12, H13, H18 J6, J7, J12, J13, K5, K6, K7, K12, K13, K14, L1, L6, L7, L12, L13, L18, M6, M7, M8, M9, M10, M11, M12, M13, N6, N7, N8, N9, N10, N11, N12, N13, T17, P9, V8, V11	Digital Ground

4.6 Digital Supply Monitoring

Signal Name	Ball	Description
dvddmon	T10	It is recommended that this pin be brought to a probe point for monitoring and debugging purposes.
dvssmon	R11	It is recommended that this pin be brought to a probe point for monitoring and debugging purposes.

4.7 JTAG

The JTAG standard for boundary scan testing will be implemented for board testing purposes. Internal scan test will not be supported. Five device pins are dedicated for JTAG support: tdi, tdo, tms, tck, and trst_n. The JTAG bsdl configuration file is available at www.ti.com.

NOTE

The trst_n pin should be asserted after power up to ensure the JTAG logic is properly initialized.

5 SPECIFICATIONS

5.1 ABSOLUTE MAXIMUM RATINGS[1]

		UNIT
VDDS	Pad ring supply voltage	–0.3 V to 3.7 V
DVDD	Core supply voltage	–0.3 V to 1.8 V
	Digital input voltage	–0.3 V to VDDS+0.3 V
	Clamp current for an input or output	–20 mA to +20 mA
T_{STG}	Storage temperature	–65°C to 140°C
T_J	Junction temperature	105°C
	Lead soldering temperature (10 seconds)	300°C
Class 2	ESD classification (tested to EIA/JESD22-A114-B)	
Class 2	Moisture sensitivity	

(1) Stresses beyond those listed under *absolute maximum ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *recommended operating conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

5.2 RECOMMENDED OPERATING CONDITIONS

		MIN	NOM	MAX	UNIT
VDDS	Digital chip, I/O ring supply voltage	3		3.6	V
DVDD	Digital chip, core supply voltage	1.425		1.575	V
	Digital chip, supply voltage difference, VDDS – DVDD			2.0	V
T_A [1]	Temperature ambient, no air flow	–40		85	°C
T_J [2]	Junction temperature			105	°C

(1) Chips specifications in Section 5.5 and Section 5.6 are production tested to 90°C case temperature. QA tests are performed at 85°C.
(2) Thermal management will be required for full rate operation, See Section 5.3. The circuit is designed for junction temperatures up to 125°C. Sustained operation at elevated temperatures will reduce long-term reliability. Lifetime calculations based on maximum junction temperature of 105°C.

5.3 THERMAL CHARACTERISTICS[1]

THERMAL CONDUCTIVITY		MIN	TYP	MAX	UNIT
θ_{JA}	Theta Junction to Ambient (still air)		15.3		°C/W
θ_{JA2m}	Theta Junction to Ambient (2m/s estimated)		12.4		°C/W
θ_{JC}	Theta Junction to Case		4.5		°C/W

(1) Air flow will reduce θ_{JA} and is highly recommended.

5.4 POWER CONSUMPTION

The maximum power consumption is largely a function of the operating mode of the chip.

IDVDD = proportional to filter lengths, supply, frequency, and number of channels active.

Current consumption on the pad supply is primarily due to the external loads and follows C×V×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 rx_sync_out_X frame strobes consume negligible power due to the low transition frequency. In general,

$$IVDDS = \Sigma\ DataPad/4 \times C \times F \times V + \Sigma\ ClockPad \times C \times F \times V$$

5.5 DC CHARACTERISTICS[1] [2] [3]

–40°C to 85°C case (unless otherwise noted)

PARAMETER		VDDS=3 V to 3.6 V		UNIT
		MIN	MAX	
V_{IL}	Voltage input low		0.8	V
V_{IH}	Voltage input high	2.0		V
V_{OL}	Voltage output low (I_{OL} = 2 mA)[4]		0.5	V
V_{OH}	Voltage output high (I_{OH} = –2 mA)[4]	2.4	VDDS	V
$\|I_{PU}\|$	Pullup current (V_{IN} = 0 V) (**tdi, tms, trst_n, ce_n, wr_n, rd_n, reset_n**) (nominal 20 µA)[4]	5	35	µA
$\|I_{PD}\|$	Pulldown current (V_{IN} = VDDS) (all other inputs and bidirectionals) (nominal 20 µA)[4]	5	35	µA
$\|I_{IN}\|$	Leakage current (V_{IN} = 0V or VDDS), Outputs in 3-state condition [4]		20	µA
I_{CCQ}	Quiescent supply current, IDVDD or IVDDS (V_{IN} = 0 for pads with pulldowns, V_{IN} = VDDS for inputs with pullups) [4]		8	mA
C_{IN}	Capacitance for inputs[5]	Typical 5	Typical 5	pF
C_{BI}	Capacitance for bidirectionals[5]	Typical 5	Typical 5	pF

[1] Voltages are measured at low speed. Output voltages are measured with the indicated current load.
[2] Currents are measured at nominal voltages, high temperature (90°C for production test, 85°C for QA).
[3] reset_n and interrupt have no timing specifications since they are asynchronous signals.
die_id pins fa002_out, fa002_clk and fa002_scan will not be specified and are for factory use only.
fuse pin fuse_out will not be specified and is for factory use only.
test pins zero, scanen, testmode0 and testmode1 will not be specified and are for factory use only.
[4] Each part is tested at 90°C case temperature for the given specification. Lots are sample tested at -40°C.
[5] Controlled by design and process and not directly tested.

5.6 AC TIMING CHARACTERISTICS[1]

–40°C to 85°C case supplies across recommended range (unless otherwise noted)

PARAMETER		MIN	TYP	MAX	UNIT
F_{CK}	Clock frequency (**adcclk_a/b/c/d, rxclk**)[2]			160	MHz
t_{CKL}	Clock low period (below V_{IL}) (**adcclk_a/b/c/d, rxclk**)[2]	2			ns
t_{CKH}	Clock high period (above V_{IH}) (**adcclk_a/b/c/d, rxclk**)[2]	2			ns
t_{RF}	Clock rise and fall times (V_{IL} to V_{IH}) (**adcclk_a/b/c/d, rxclk**)[3]			2	ns
t_{SU}	Input setup (**rxsync_a/b/c/d)** before **rxclk** rises[2]	2			ns
	Input setup (**rxin_a/b/c/d_[0-15]**) before **rxclk** rises (adc fifo blocks bypassed)[2]	2			
	Input setup (**rxin_a/b/c/d_[0-15]**) before **adcclk_a/b/c/d** rises (adc fifo blocks enabled)[2]	2			
t_{HD}	Input hold (**rxsync_a/b/c/d**) after **rxclk** rises[2]	1			ns
	Input hold (**rxin_a/b/c/d_[0-15]**) after **rxclk** rises (adc fifo blocks bypassed)[2]	2.5			
	Input hold (**rxin_ a/b/c/d_[0-15]**) after **adcclk_a/b/c/d** rises (adc fifo blocks enabled)[2]	1			
t_{DLY}	Data output delay (**rx_sync_out_[0-7], rxout_[0-7]_a/b/c/d, rxclk_out, rx_sync_out, dvga_[a-d]_[5-0]**) after **rxclk** rises.[2]			7	ns
t_{OHD}	Data output hold **(rx_sync_out_[0-7], rxout_[0-7]_a/b/c/d, rxclk_out, rx_sync_out, dvga_[a-d]_[5-0]**) after **rxclk** rises.[2]	0.5			ns
F_{JCK}	JTAG Clock frequency (**tck**)[2]			40	MHz
t_{JCKL}	JTAG Clock low period (below V_{IL}) (**tck**)[2]	10			ns
t_{JCKH}	JTAG Clock high period (above V_{IH}) (tck)[2]	10			ns
t_{JSU}	JTAG Input (**tdi** or **tms**) setup before **tck** goes high[2]	2			ns
t_{JHD}	JTAG Input (**tdi** or **tms**) hold time after **tck** goes high[2]	10			ns
t_{JDLY}	JTAG output (**tdo**) delay from falling edge of **tck**.[2]			10	
t_{CSU}	Control setup during reads or writes 3 pin mode: **a[5:0]** valid before **rd_n, wr_n or ce_n** falling edge 2 pin mode: **a[5:0]** and **wr_n** valid before **ce_n** falling edge [2]	6			ns
t_{EWCSU}	Control setup during writes 3 pin mode: **d[15:0]** valid before **wr_n** and **ce_n** rising edge 2 pin mode: **d[15:0]** valid before **ce_n** rising edge [2]	10			ns
t_{CHD}	Control hold during writes. 3 pin mode: **a[5:0]** and **d[15:0]** valid after **wr_n** and **ce_n** rise 2 pin mode: **a[5:0], d[15:0]** and **wr_n** valid after **ce_n** rise [2]	6			ns
t_{CSPW}	Control strobe (**ce_n** and **wr_n** low) pulse width during write. [2]	25			ns
t_{CDLY}	Control output delay **ce_n** and **rd_n** low and **a[5:0]** stable to **d[15:0]** during read. [2]			25	ns
t_{REC}	Control recovery time between reads or writes. [2]			6	ns
t_{HIZ}	Control end of read to Hi-Z. **rd_n** and **ce_n** rise to **d[15:0]** 3-state [4]			10	ns
t_{COH}	Control read **d[15:0]** output hold time[4]	1			ns
I_{CDYN}	Core dynamic supply current ,nominal voltages, 160 MHz, (specific conditions, typical app with chip busy within capability of the tester, high temperature.)[4]		1700		mA

[1] Timing is measured from the respective clock at VDDS/2 to input or output at VDDS/2. Output loading is a 50 Ω transmission line whose delay is calibrated out.
[2] Each part is tested at 90°C case temperature for the given specification. Lots are sample tested at –40°C.
[3] Recommended practice.
[4] Controlled by design and process and not directly tested. Verified on initial part evaluation.

6 Revision History

6.1 Revision History Table

DATE	REV	PAGE	SECTION	DESCRIPTION
6 AUG 05	A		DC Characterisitcs	Changed Note 1
			AC Timing Characterisitcs	Changed note references at section header
				t_{HD} Input hold (rxin_a/b/c/d_[0-15]) after rxclk rises (adc fifo blocks bypassed) changed from 2.0 to 2.5 ns min
				t_{DLY} changed from 6.5 ns to 7.0 ns
				t_{JSU} JTAG input (tdi or tms) setup from 1.0 ns to 2.0 ns min
				t_{CSPW} control strobe pulse width during write from 15 ns to 25 ns min
				t_{HIZ} control end of read Hi-Z needs note (4) instead of note (1)
				t_{COH} control read d(15:0) hold time from 3 ns to 1 ns, and note changed from (1) to (4)
				I_{CDYN} needs to be 1700 mA typical (not a maximum, but a typical value) and note changed from (1) to (4)
25 MAR 05	*	–	–	Original version

Changes from Revision A (November 2005) to Revision B **Page**

- Changed unused to cfir_gain in section 4.4.5.2 99

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
GC5018IZDL	NRND	BGA	ZDL	305	84	Green (RoHS & no Sb/Br)	SNAGCU	Level-3-260C-168 HR	-40 to 85	GC5018IZDL	

[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.

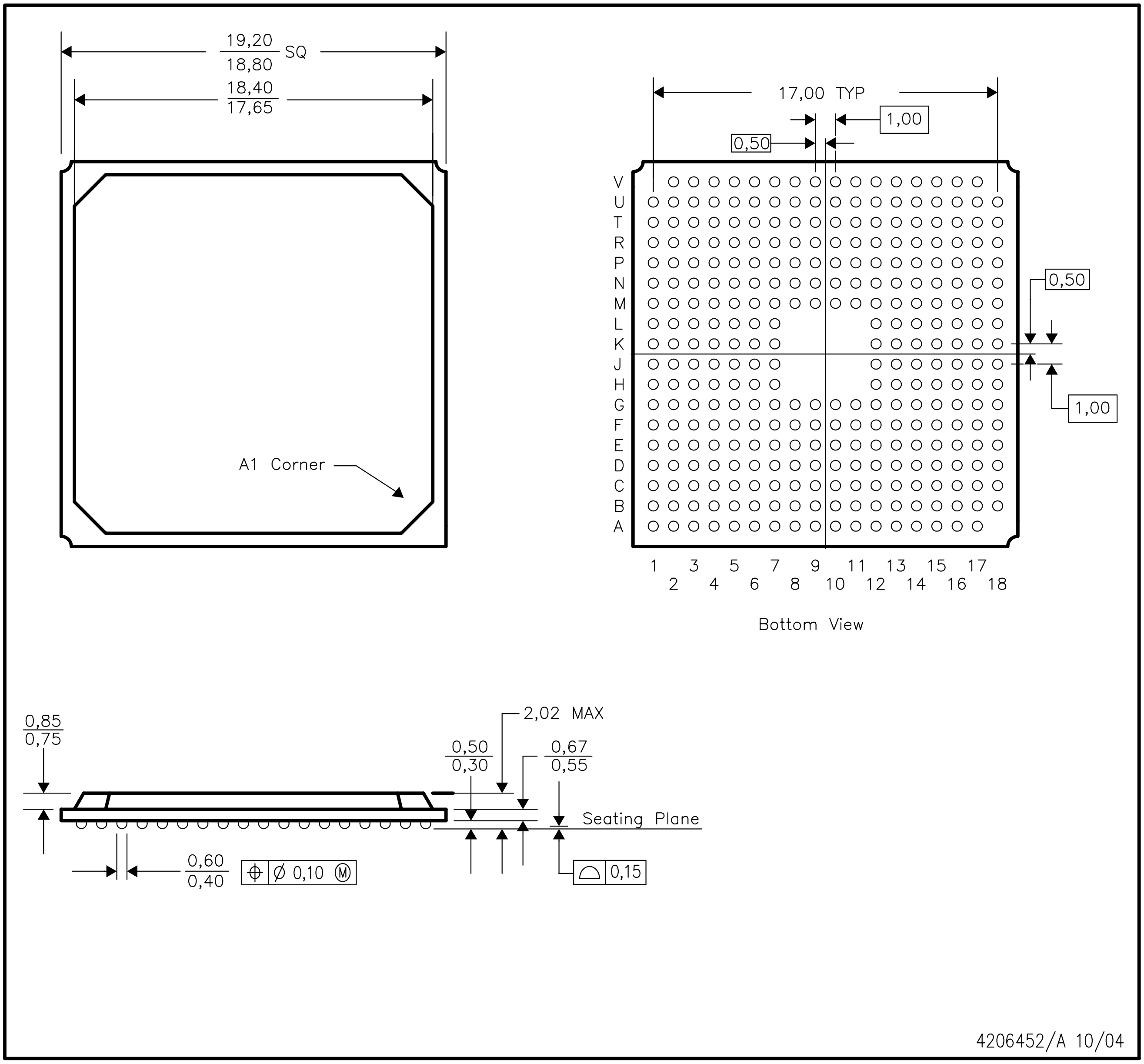
[4] There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

[5] Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

ZDL (S-PBGA-N305) PLASTIC BALL GRID ARRAY



NOTES:
- A. All linear dimensions are in millimeters.
- B. This drawing is subject to change without notice.
- C. This package is lead-free.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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