150 MSPS, Wideband, Digital Downconverter (DDC)

AD6636

FEATURES

- 4/6 independent wideband processing channels
- Processes 6 wideband carriers (UMTS, CDMA2000)
- 4 single-ended or 2 LVDS parallel input ports
  (16 linear bit plus 3-bit exponent) running at 150 MHz
- Supports 300 MSPS input using external interface logic
- Three 16-bit parallel output ports operating up to 200 MHz
- Real or complex input ports
- Quadrature correction and dc correction for complex inputs
- Supports output rate up to 34 MSPS per channel
- RMS/peak power monitoring of input ports
- Programmable attenuator control for external gain ranging
- 3 programmable coefficient FIR filters per channel
- 2 decimating half-band filters per channel
- 6 programmable digital AGC loops with 96 dB range
- Synchronous serial I/O operation (SPI®-, SPORT-compatible)
- Supports 8-bit or 16-bit microport modes
- 3.3 V I/O, 1.8 V CMOS core
- User-configurable, built-in, self-test (BIST) capability
- JTAG boundary scan

APPLICATIONS

- Multicarrier, multimode digital receivers
- GSM, EDGE, PHS, UMTS, WCDMA, CDMA2000, TD-SCDMA, WiMAX
- Micro and pico cell systems, software radios
- Broadband data applications
- Instrumentation and test equipment
- Wireless local loops
- In-building wireless telephony

FUNCTIONAL BLOCK DIAGRAM

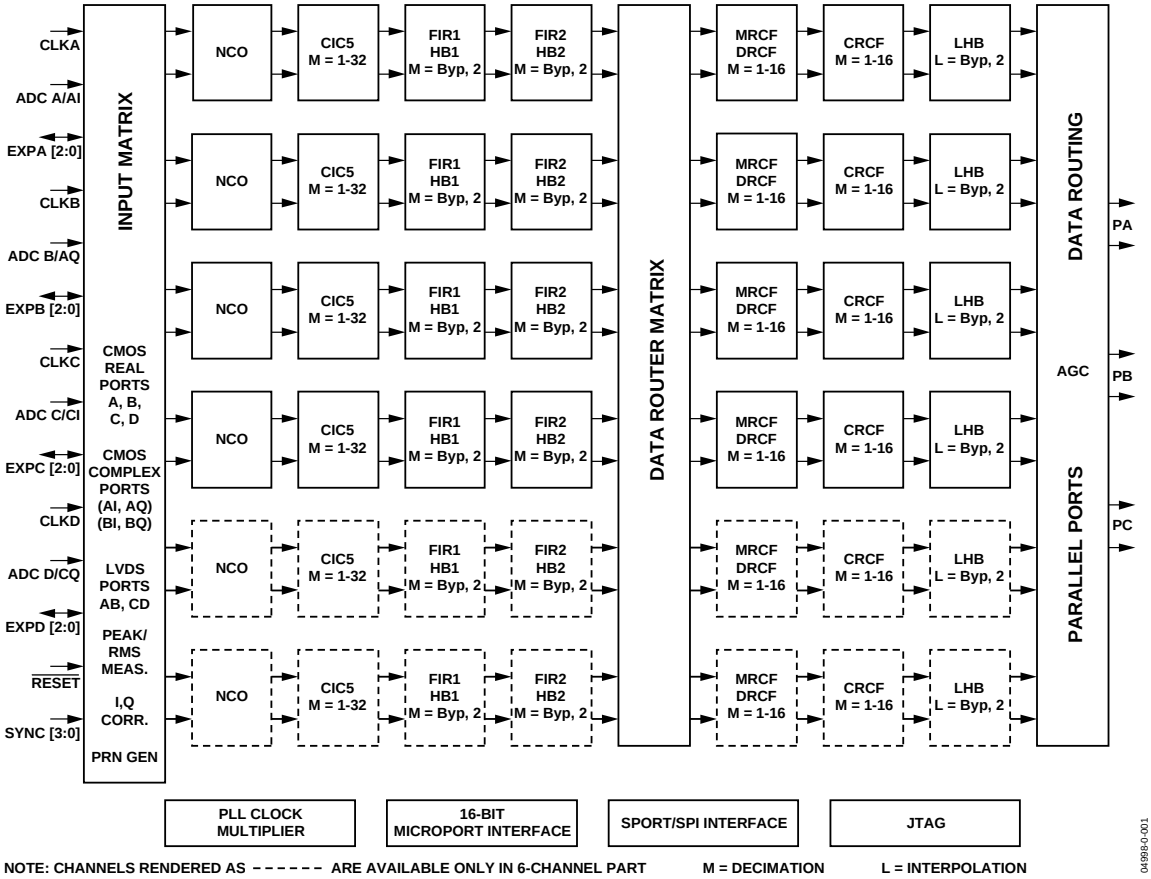


Figure 1.

Rev. A

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113

© 2005 Analog Devices, Inc. All rights reserved.

TABLE OF CONTENTS

General Description 4
Specifications 6
Recommended Operating Conditions 6
Electrical Characteristics 6
General Timing Characteristics[1] 7
Microport Timing Characteristics[1] 8
Serial Port Timing Characteristics[1, 2] 9
Explanation of Test Levels for Specifications 9
Absolute Maximum Ratings 10
Thermal Characteristics 10
ESD Caution 10
Pin Configuration and Function Descriptions 11
Pin Listing for Power, Ground, Data, and Address Buses 13
Timing Diagrams 14
Theory of Operation 20
ADC Input Port 20
PLL Clock Multiplier 21
ADC Gain Control 22
ADC Input Port Monitor Function 23
Quadrature I/Q Correction Block 25
Input Crossbar Matrix 27
Numerically Controlled Oscillator (NCO) 27
Fifth-Order CIC Filter 29
FIR Half-Band Block 30
Intermediate Data Router 33
MonoRate RAM Coefficient Filter (MRCF) 33
Decimating RAM Coefficient Filter (DRCF) 34
Channel RAM Coefficient Filter (CRCF) 36
Interpolating Half-Band Filter 38
Output Data Router 38
Automatic Gain Control 40
Parallel Port Output 44
User-Configurable, Built-In Self-Test (BIST) 48
Chip Synchronization 48
Serial Port Control 49
Microport 58
Memory Map 60
Reading the Memory Map Table 60
Global Register Map 62
Input Port Register Map 65
Channel Register Map 68
Output Port Register Map 73
Design Notes 77
Outline Dimensions 79
Ordering Guide 79

REVISION HISTORY

6/05—Rev. 0 to Rev. A

Changes to Format ..... Universal
Changes to Figure 1 1
Changes to Applications 1
Changes to General Description 4
Changes to Table 3 7
Changes to Table 5 9
Changes to Table 8 11
Changes to Figure 17 18
Changes to Figure 18 and Figure 19 19
Changes to Figure 25 23
Changes to Mean Power Mode (Control Bits 01) Section 24
Changes to NCO Frequency Section 27
Changes to Figure 30 28
Changes to 6-Tap Fixed Coefficient Filter (FIR2) Section 32
Changes to Decimate-by-2, Half-Band Filter (HB2) Section 32
Changes to Table 17 32
Changes to Clock Rate Section 34
Changes to Programming DRCF Register for a Symmetric Filter Section 35
Changes to Channel RAM Coefficient Filter (CRCF) Section 36
Changes to Programming CRCF Register for a Symmetrical Filter Section 37
Changes to Desired Signal Level Mode Section 41
Changes to Figure 41 45
Changes to Figure 42 and Figure 43 46
Changes to Start with Soft Sync Section 48
Changes to Hop with Soft Sync Section 49
Changes to Hop with Pin Sync Section 49
Replaced Serial Control Port Section 49
Changes to Intel (INM) Mode Section 58
Changes to Motorola (MNM) Mode Section 59
Changes to Table 30 61
Changes to Channel Register Map Section 68
Changes to AGC Control Register <10:0> Section 71
Changes to BIST Control <15:0> Section 73
Changes to Parallel Port Output Control <23:0> 73
Changes to Table 44 74
Changes to Design Notes 77
Changes to Figure 59 77

8/04—Revision 0: Initial Version

GENERAL DESCRIPTION

The AD6636 is a digital downconverter intended for IF sampling or oversampled baseband radios requiring wide bandwidth input signals. The AD6636 has been optimized for the demanding filtering requirements of wideband standards, such as CDMA2000, UMTS, and TD-SCDMA, but is flexible enough to support wider standards such as WiMAX. The AD6636 is designed for radio systems that use either an IF sampling ADC or a baseband sampling ADC.

The AD6636 channels have the following signal processing stages: a frequency translator, a fifth-order cascaded integrated comb filter, two sets of cascaded fixed-coefficient FIR and half-band filters, three cascaded programmable coefficient sum-of-product FIR filters, an interpolating half-band filter (IHB), and a digital automatic gain control (AGC) block. Multiple modes are supported for clocking data into and out of the chip and provide flexibility for interfacing to a wide variety of digitizers. Programming and control are accomplished via serial or microport interfaces.

Input ports can take input data at up to 150 MSPS. Up to 300 MSPS input data can be supported using two input ports (some external interface logic is required) and two internal channels processing in tandem. Biphase filtering in the output data router is selected to complete the combined filtering mode. The four input ports can operate in CMOS mode, or two ports can be combined for LVDS input mode. The maximum input data rate for each input port is 150 MHz.

Frequency translation is accomplished with a 32-bit complex numerically controlled oscillator (NCO). It has greater than 110 dBc SFDR. This stage translates either a real or complex input signal from intermediate frequency (IF) to a baseband complex digital output. Phase and amplitude dither can be enabled on-chip to improve spurious performance of the NCO. A 16-bit phase-offset word is available to create a known phase relationship between multiple AD6636 chips or channels. The NCO can also be bypassed so that baseband I and Q inputs can be provided directly from baseband sampling ADCs through input ports.

Following frequency translation is a fifth-order CIC filter with a programmable decimation between 1 and 32. This filter is used to lower the sample rate efficiently, while providing sufficient alias rejection at frequencies with higher frequency offsets from the signal of interest.

Following the CIC5 are two sets of filters. Each set has a non-decimating FIR filter and a decimate-by-2 half-band filter. The FIR1 filter provides about 30 dB of rejection, while the HB1 filter provides about 77 dB of rejection. They can be used together to achieve a 107 dB stop band alias rejection, or they can be individually bypassed to save power. The FIR2 filter provides about 30 dB of rejection, while the HB2 filter provides about 65 dB of rejection. The filters can be used either together to achieve more than 95 dB stop band alias rejection, or can be individually bypassed to save power. FIR1 and HB1 filters can run with a maximum input rate of 150 MSPS. In contrast, FIR2 and HB2 can run with a maximum input rate of 75 MSPS (input rate to FIR2 and HB2 filters).

The programmable filtering is divided into three cascaded RAM coefficient filters (RCFs) for flexible and power efficient filtering. The first filter in the cascade is the MRCF, consisting of a programmable nondecimating FIR. It is followed by programmable FIR filters (DRCF) with decimation from 1 to 16. They can be used either together to provide high rejection filters, or independently to save power. The maximum input rate to the MRCF is one-fourth of the PLL clock rate.

The channel RCF (CRCF) is the last programmable FIR filter with programmable decimation from 1 to 16. It typically is used to meet the spectral mask requirements for the air standard of interest. This could be an RRC, antialiasing filter or any other real data filter. Decimation in preceding blocks is used to keep the input rate of this stage as low as possible for the best filter performance.

The last filter stage in the chain is an interpolate-by-2 half-band filter, which is used to up-sample the CRCF output to produce higher output oversampling. Signal rejection requirements for this stage are relaxed because preceding filters have filtered the blockers and adjacent carriers already.

Each input port of the AD6636 has its own clock used for latching onto the input data, but the Input Port A clock (CLKA) is also used as the input for an on-board PLL clock multiplier. The output of the PLL clock is used for processing all filters and processing blocks beyond the data router following the CIC filter. The PLL clock can be programmed to have a maximum clock rate of 200 MHz.

A data routing block (DR) is used to distribute data from the CICs to the various channel filters. This block allows multiple back-end filter chains to work together to process high bandwidth signals or to make even sharper filter transitions than a single channel can perform. It can also allow complex filtering operations to be achieved in the programmable filters.

The digital AGC provides the user with scaled digital outputs based on the rms level of the signal present at the output of the digital filters. The user can set the requested level and time constant of the AGC loop for optimum performance of the postprocessor. This is a critical function in the base station for CDMA applications where the power level must be well controlled going into the RAKE receivers. It has programmable clipping and rounding control to provide different output resolutions.

The overall filter response for the AD6636 is the composite of all the combined filter stages. Each successive filter stage is capable of narrower transition bandwidths but requires a greater number of CLK cycles to calculate the output. More decimation in the first filter stage minimizes overall power consumption. Data from the device is interfaced to a DSP/FPGA/baseband processor via either high speed parallel ports (preferred) or a DSP-compatible microprocessor interface.

The AD6636 is available both in 4-channel and 6-channel versions. The data sheet primarily discusses the 6-channel part. The only difference between the 6-channel and 4-channel devices is that Channel 4 and Channel 5 are not available on the 4-channel version, (see Figure 1). The 4-channel device still has the same input ports, output ports, and memory map. The memory map section for Channel 4 and Channel 5 can be programmed and read back, but it serves no purpose.

PRODUCT HIGHLIGHTS

- Six independent digital filtering channels
- 101 dB SNR noise performance, 110 dB spurious performance
- Four input ports capable of 150 MSPS input data rates
- RMS/peak power monitoring of input ports and 96 dB range AGCs before the output ports
- Three programmable RAM coefficient filters, three half-band filters, two fixed coefficient filters, and one fifth-order CIC filter per channel
- Complex filtering and biphase filtering (300 MSPS ADC input) by combining filtering capability of multiple channels
- Three 16-bit parallel output ports operating at up to a 200 MHz clock
- Black*fin*®-compatible and *Tiger*SHARC®-compatible 16-bit microprocessor port
- Synchronous serial communications port is compatible with most serial interface standards, SPORT, SPI, and SSR

SPECIFICATIONS

RECOMMENDED OPERATING CONDITIONS

Table 1.

Parameter	Temp	Test Level	Min	Typ	Max	Unit
VDDCORE	Full	IV	1.7	1.8	1.9	V
VDDIO	Full	IV	3.0	3.3	3.6	V
$T_{AMBIENT}$	Full	IV	−40	+25	+85	°C

ELECTRICAL CHARACTERISTICS[1]

Table 2.

Parameter	Temp	Test Level	Min	Typ	Max	Unit
LOGIC INPUTS (NOT 5 V TOLERANT)						
Logic Compatibility	Full	IV	3.3			V CMOS
Logic 1 Voltage	Full	IV	2.0		3.6	V
Logic 0 Voltage	Full	IV	−0.3		+0.8	V
Logic 1 Current	Full	IV		1	10	µA
Logic 0 Current	Full	IV		1	10	µA
Input Capacitance	25°C	V		4		pF
LOGIC OUTPUTS						
Logic Compatibility	Full	IV	3.3			V CMOS
Logic 1 Voltage (I_{OH} = 0.25 mA)	Full	IV	2.0	VDDIO − 0.2		V
Logic 0 Voltage (I_{OL} = 0.25 mA)	Full	IV		0.2	0.4	V
SUPPLY CURRENTS						
WCDMA (61.44 MHz) Example[1]						
$I_{VDDCORE}$	25°C	V		450		mA
I_{VDDIO}	25°C	V		50		mA
CDMA 2000 (61.44 MHz) Example[1]						
$I_{VDDCORE}$	25°C	V		400		mA
I_{VDDIO}	25°C	V		25		mA
TDS-CDMA (76.8 MHz) Example[1, 2]						
$I_{VDDCORE}$	25°C	V		250		mA
I_{VDDIO}	25°C	V		15		mA
GSM (65 MHz) Example[1, 2]						
$I_{VDDCORE}$	25°C	V		175		mA
I_{VDDIO}	25°C	V		10		mA
TOTAL POWER DISSIPATION						
WCDMA (61.44 MHz)[1]	25°C	V		975		mW
CDMA2000 (61.44 MHz)[1]	25°C	V		800		mW
TD-SCDMA (76.8 MHz)[1, 2]	25°C	V		500		mW
GSM (65 MHz)[1, 2]	25°C	V		350		mW

[1] One input port, all six channels, and the relevant signal processing blocks are active.
[2] PLL is turned off for power savings.

GENERAL TIMING CHARACTERISTICS[1, 2]

Table 3.

Parameter		Temp	Test Level	Min	Typ	Max	Unit
CLK TIMING REQUIREMENTS							
t_{CLK}	CLKx Period (x = A, B, C, D)	Full	I	6.66			ns
t_{CLKL}	CLKx Width Low (x = A, B, C, D)	Full	IV	1.71	$0.5 \times t_{CLK}$		ns
t_{CLKH}	CLKx Width High (x = A, B, C, D)	Full	IV	1.70	$0.5 \times t_{CLK}$		ns
$t_{CLKSKEW}$	CLKA to CLKx Skew (x = B, C, D)	Full	IV	$t_{CLK} - 1.3$			ns
INPUT WIDEBAND DATA TIMING REQUIREMENTS		Full	IV				
t_{SI}	INx [15:0] to ↑CLKx Setup Time (x = A, B, C, D)	Full	IV	0.75			ns
t_{HI}	INx [15:0] to ↑CLKx Hold Time (x = A, B, C, D)	Full	IV	1.13			ns
t_{SEXP}	EXPx [2:0] to ↑CLKx Setup Time (x = A, B, C, D)	Full	IV	3.37			ns
t_{HEXP}	EXPx [2:0] to ↑CLKx Hold Time (x = A, B, C, D)	Full	IV	1.11			ns
t_{DEXP}	↑CLKx to EXPx[2:0] Delay (x = A, B, C, D)	Full	IV	5.98		10.74	ns
PARALLEL OUTPUT PORT TIMING REQUIREMENTS (MASTER)							
t_{DPREQ}	↑PCLK to ↑Px REQ Delay (x = A, B, C)	Full	IV	1.77		3.86	ns
t_{DPP}	↑PCLK to Px [15:0] Delay (x = A, B, C)	Full	IV	2.07		5.29	ns
t_{DPIQ}	↑PCLK to Px IQ Delay (x = A, B, C)	Full	IV	0.48		5.49	ns
t_{DPCH}	↑PCLK to Px CH[2:0] Delay (x = A, B, C)	Full	IV	0.38		5.35	ns
t_{DPGAIN}	↑PCLK to Px Gain Delay (x = A, B, C)	Full	IV	0.23		4.95	ns
t_{SPA}	Px ACK to ↑PCLK Setup Time (x = A, B, C)	Full	IV	4.59			ns
t_{HPA}	Px ACK to ↑PCLK Hold Time (x = A, B, C)	Full	IV	0.90			ns
PARALLEL OUTPUT PORT TIMING REQUIREMENTS (SLAVE)							
t_{PCLK}	PCLK Period	Full	IV	5.0			ns
t_{PCLKL}	PCLK Low Period	Full	IV	1.7	$0.5 \times t_{PCLK}$		ns
t_{PCLKH}	PCLK High Period	Full	IV	0.7	$0.5 \times t_{PCLK}$		ns
t_{DPREQ}	↑PCLK to ↑Px REQ Delay (x = A, B, C)	Full	IV	4.72		8.87	ns
t_{DPP}	↑PCLK to Px [15:0] Delay (x = A, B, C)	Full	IV	4.8		8.48	ns
t_{DPIQ}	↑PCLK to Px IQ Delay (x = A, B, C)	Full	IV	4.83		10.94	ns
t_{DPCH}	↑PCLK to Px CH[2:0] Delay (x = A, B, C)	Full	IV	4.88		10.09	ns
t_{DPGAIN}	↑PCLK to Px Gain Delay (x = A, B, C)	Full	IV	5.08		11.49	ns
t_{SPA}	Px ACK to ↓PCLK Setup Time (x = A, B, C)	Full	IV	6.09			ns
t_{HPA}	Px ACK to ↓PCLK Hold Time (x = A, B, C)	Full	IV	1.0			ns
MISC PINS TIMING REQUIREMENTS							
t_{RESET}	$\overline{\text{RESET}}$ Width Low	Full	IV	30			ns
t_{DIRP}	CPUCLK/SCLK to $\overline{\text{IRP}}$ Delay	Full	V		7.5		ns
t_{SSYNC}	SYNC(0, 1, 2, 3) to ↑CLKA Setup Time	Full	IV	0.87			ns
t_{HSYNC}	SYNC(0, 1, 2, 3) to ↑CLKA Hold Time	Full	IV	0.67			ns

[1] All timing specifications are valid over the VDDCORE range of 1.7 V to 1.9 V and the VDDIO range of 3.0 V to 3.6 V.
[2] C_{LOAD} = 40 pF on all outputs, unless otherwise noted.

MICROPORT TIMING CHARACTERISTICS[1, 2]

Table 4.

Parameter		Temp	Test Level	Min	Typ	Max	Unit
MICROPORT CLOCK TIMING REQUIREMENTS							
t_{CPUCLK}	CPUCLK Period	Full	IV	10.0			ns
$t_{CPUCLKL}$	CPUCLK Low Time	Full	IV	1.53	0.5 × t_{CPUCLK}		ns
$t_{CPUCLKH}$	CPUCLK High Time	Full	IV	1.70	0.5 × t_{CPUCLK}		ns
INM MODE WRITE TIMING (MODE = 0)							
t_{SC}	Control[3] to ↑CPUCLK Setup Time	Full	IV	0.80			ns
t_{HC}	Control[3] to ↑CPUCLK Hold Time	Full	IV	0.09			ns
t_{SAM}	Address/Data to ↑CPUCLK Setup Time	Full	IV	0.76			ns
t_{HAM}	Address/Data to ↑CPUCLK Hold Time	Full	IV	0.20			ns
t_{DRDY}	↑CPUCLK to RDY ($\overline{DTACK}$) Delay	Full	IV	3.51		6.72	ns
t_{ACC}	Write Access Time	Full	IV	3 × t_{CPUCLK}		9 × t_{CPUCLK}	ns
INM MODE READ TIMING (MODE = 0)							
t_{SC}	Control[3] to ↑CPUCLK Setup Time	Full	IV	1.00			ns
t_{HC}	Control[3] to ↑CPUCLK Hold Time	Full	IV	0.03			ns
t_{SAM}	Address to ↑CPUCLK Setup Time	Full	IV	0.80			ns
t_{HAM}	Address to ↑CPUCLK Hold Time	Full	IV	0.20			ns
t_{DD}	↑CPUCLK to Data Delay	Full	V		5.0		ns
t_{DRDY}	↑CPUCLK to RDY ($\overline{DTACK}$) Delay	Full	IV	4.50		6.72	ns
t_{ACC}	Read Access Time	Full	IV	3 × t_{CPUCLK}		9 × t_{CPUCLK}	ns
MNM MODE WRITE TIMING (MODE = 1)							
t_{SC}	Control[3] to ↑CPUCLK Setup Time	Full	IV	1.00			ns
t_{HC}	Control[3] to ↑CPUCLK Hold Time	Full	IV	0.00			ns
t_{SAM}	Address/Data to ↑CPUCLK Setup Time	Full	IV	0.00			ns
t_{HAM}	Address/Data to ↑CPUCLK Hold Time	Full	IV	0.57			ns
t_{DDTACK}	↑CPUCLK to $\overline{DTACK}$ (RDY) Delay	Full	IV	4.10		5.72	ns
t_{ACC}	Write Access Time	Full	IV	3 × t_{CPUCLK}		9 × t_{CPUCLK}	ns
MNM MODE READ TIMING (MODE = 1)							
t_{SC}	Control[3] to ↑CPUCLK Setup Time	Full	IV	1.00			ns
t_{HC}	Control[3] to ↑CPUCLK Hold Time	Full	IV	0.00			ns
t_{SAM}	Address to ↑CPUCLK Setup Time	Full	IV	0.00			ns
t_{HAM}	Address to ↑CPUCLK Hold Time	Full	IV	0.57			ns
t_{DD}	CPUCLK to Data Delay	Full	V		5.0		ns
t_{DDTACK}	↑CPUCLK to $\overline{DTACK}$ (RDY) Delay	Full	IV	4.20		6.03	ns
t_{ACC}	Read Access Time	Full	IV	3 × t_{CPUCLK}		9 × t_{CPUCLK}	ns

[1] All timing specifications are valid over the VDDCORE range of 1.7 V to 1.9 V and the VDDIO range of 3.0 V to 3.6 V.
[2] C_{LOAD} = 40 pF on all outputs, unless otherwise noted.
[3] Specification pertains to control signals: R/$\overline{W}$ ($\overline{WR}$), $\overline{DS}$ ($\overline{RD}$), and $\overline{CS}$.

SERIAL PORT TIMING CHARACTERISTICS[1, 2, 3]

Table 5.

Parameter		Temp	Test Level	Min	Typ	Max	Unit
SERIAL PORT CLOCK TIMING REQUIREMENTS							
t_{SCLK}	SCLK Period	Full	IV	10.0			ns
t_{SCLKL}	SCLK Low Time	Full	IV	1.60	$0.5 \times t_{SCLK}$		ns
t_{SCLKH}	SCLK High Time	Full	IV	1.60	$0.5 \times t_{SCLK}$		ns
SPI PORT CONTROL TIMING REQUIREMENTS (MODE = 0)							
t_{SSDI}	SDI to ↑SCLK Setup Time	Full	IV	1.30			ns
t_{HSDI}	SDI to ↑SCLK Hold Time	Full	IV	0.40			ns
t_{SSCS}	$\overline{SCS}$ to ↑SCLK Setup Time	Full	IV	4.12			ns
t_{HSCS}	$\overline{SCS}$ to ↑SCLK Hold Time	Full	IV	−2.78			ns
t_{DSDO}	↑SCLK to SDO Delay Time	Full	IV	4.28		7.96	ns
SPORT MODE CONTROL TIMING REQUIREMENTS (MODE = 1)							
t_{SSDI}	SDI to ↑SCLK Setup Time	Full	IV	0.80			ns
t_{HSDI}	SDI to ↑SCLK Hold Time	Full	IV	0.40			ns
t_{SSRFS}	SRFS to ↓SCLK Setup Time	Full	IV	1.60			ns
t_{HSRFS}	SRFS to ↓SCLK Hold Time	Full	IV	−0.13			ns
t_{SSTFS}	STFS to ↓SCLK Setup Time	Full	IV	1.60			ns
t_{HSTFS}	STFS to ↑SCLK Hold Time	Full	IV	−0.30			ns
t_{SSCS}	$\overline{SCS}$ to ↑SCLK Setup Time	Full	IV	4.12			ns
t_{HSCS}	$\overline{SCS}$ to ↑SCLK Hold Time	Full	IV	−2.76			ns
t_{DSDO}	↑SCLK to SDO Delay Time	Full	IV	4.29		7.95	ns

[1] All timing specifications are valid over the VDDCORE range of 1.7 V to 1.9 V and the VDDIO range of 3.0 V to 3.6 V.
[2] C_{LOAD} = 40 pF on all outputs, unless otherwise noted.
[3] SCLK rise/fall time should be 3 ns maximum.

EXPLANATION OF TEST LEVELS FOR SPECIFICATIONS

Table 6.

Test Level	Description
I	100% production tested.
II	100% production tested at 25°C, and sample tested at specified temperatures.
III	Sample tested only.
IV	Parameter guaranteed by design and analysis.
V	Parameter is typical value only.
VI	100% production tested at 25°C, and sampled tested at temperature extremes.

ABSOLUTE MAXIMUM RATINGS

Table 7.

Parameter	Rating
ELECTRICAL	
VDDCORE Supply Voltage (Core Supply)	2.2 V
VDDIO Supply Voltage (Ring or IO Supply)	4.0 V
Input Voltage	−0.3 to +3.6 V (Not 5 V Tolerant)
Output Voltage	−0.3 to VDDIO + 0.3 V
Load Capacitance	200 pF
ENVIRONMENTAL	
Operating Temperature Range (Ambient)	−40°C to +85°C
Maximum Junction Temperature Under Bias	125°C
Storage Temperature Range (Ambient)	−65°C to +150°C

Stresses above those listed under the Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTICS

256-ball CSP_BGA package:

θ_{JA} = 25.4°C /W, no airflow

θ_{JA} = 23.3°C /W, 0.5 m/s airflow

θ_{JA} = 22.6°C /W, 1.0 m/s airflow

θ_{JA} = 21.9°C /W, 2.0 m/s airflow

Thermal measurements made in the horizontal position on a 4-layer board with vias.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although this product features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.



PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
A	GND	INC3	IND4	IND7	CLKD	CLKC	IND11	GND	VDDCORE	IND14	IND15	SYNC1	TDO	PBGAIN	PB11	GND	A
B	IND0	VDDIO	INC2	IND5	IND6	IND8	IND10	IND12	IND13	INC14	SYNC3	SYNC0	$\overline{\text{TRST}}$	PBCH2	VDDIO	PB12	B
C	EXPA1	EXPD1	INC0	INC1	IND3	INC5	IND9	INC10	INC13	SYNC2	TMS	TCLK	PBCH0	PB8	PB15	PB10	C
D	EXPB0	EXPC2	EXPC1	EXPD0	IND2	INC4	INC7	INC9	INC12	TDI	PBCH1	PBIQ	PB14	PB9	PB13	PACH1	D
E	INA14	INA15	EXPA0	LVDS_ RSET	GND	IND1	INC6	INC8	INC11	INC15	PBREQ	PBACK	PB4	PB5	PB1	PCLK	E
F	INA12	INA13	EXPB1	EXPC0	EXPD2	GND	VDDIO	VDDIO	VDDIO	VDDIO	GND	PB6	PB0	PB7	PAREQ	PA0	F
G	INA11	INB13	INB15	EXPB2	EXPA2	VDDCORE	GND	GND	GND	GND	VDDCORE	PB3	PAGAIN	PB2	PACH0	PA2	G
H	VDDCORE	INA10	INB12	INB11	INB14	VDDCORE	GND	GND	GND	GND	VDDCORE	PACH2	PAIQ	PAACK	PA1	GND	H
J	GND	INA9	INB10	INB8	INB9	VDDCORE	GND	GND	GND	GND	VDDCORE	PA3	PA7	PA5	PA4	VDDCORE	J
K	CLKA	INA8	INA7	INB6	INB7	VDDCORE	GND	GND	GND	GND	VDDCORE	PA12	PA15	PA9	PA8	PA6	K
L	CLKB	INA6	INB4	INB1	INB3	GND	VDDIO	VDDIO	VDDIO	VDDIO	GND	PC3	PCACK	PCCH1	PA13	PA10	L
M	INA5	INB5	INB2	INB0	GND	$\overline{\text{DTACK}}$ (RDY, SDO)	D13	D15	D5	A5	PC12	PC7	PC2	PC0	PCCH0	PA11	M
N	INA4	INA3	INA0	R/$\overline{\text{W}}$ ($\overline{\text{WR}}$, STFS)	$\overline{\text{CS}}$ ($\overline{\text{SCS}}$)	CHIPID2	D12	D2	D1	A4	A0 (SDI)	PC15	PC5	PC1	PCCH2	PA14	N
P	INA2	INA1	$\overline{\text{RESET}}$	$\overline{\text{DS}}$ ($\overline{\text{RD}}$, SRFS)	SMODE	CHIPID3	GND	D9	D4	A6	A2	PC11	PC10	PC4	PCIQ	PCGAIN	P
R	CPUCLK (SCLK)	VDDIO	MSB_ FIRST	EXT_ FILTER	CHIPID1	D14	D10	D11	D6	D0	A3	A1	PC9	PC6	VDDIO	PCREQ	R
T	GND	$\overline{\text{IRP}}$	MODE	CHIPID0	D7	D8	D3	VDDCORE	GND	GND	A7	PC14	PC13	PC8	GND	GND	T
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

= VDDCORE = VDDIO = GROUND

04998-0-002

Figure 2. CSP_BGA Pin Configuration

Table 8. Pin Function Descriptions

Mnemonic	Type	Pin No.	Function
POWER SUPPLY			
VDDCORE	Power	See Table 9	1.8 V Digital Core Supply.
VDDIO	Power	See Table 9	3.3 V Digital I/O Supply.
GND	Ground	See Table 9	Digital Core and I/O Ground.
INPUT (ADC) PORTS (CMOS/LVDS)			
CLKA	Input	K1	Clock for Input Port A. Used to clock INA[15:0] and EXPA[2:0] data. Additionally, this clock is used to drive internal circuitry and PLL clock multiplier.
CLKB	Input	L1	Clock for Input Port B. Used to clock INB[15:0] and EXPB[2:0] data.
CLKC	Input	A6	Clock for Input Port C. Used to clock INC[15:0] and EXPC[2:0] data.
CLKD	Input	A5	Clock for Input Port D. Used to clock IND[15:0] and EXPD[2:0] data.
INA[0:15]	Input	See Table 9	Input Port A (Parallel).
INB[0:15]	Input	See Table 9	Input Port B (Parallel).
INC[0:15]	Input	See Table 9	Input Port C (Parallel).
IND[0:15]	Input	See Table 9	Input Port D (Parallel).
EXPA[0:2]	Bidirectional	E3, C1, G5	Exponent Bus Input Port A. Gain control output.

Mnemonic	Type	Pin No.	Function
EXPB[0:2]	Bidirectional	D1, F3, G4	Exponent Bus Input Port B. Gain control output.
EXPC[0:2]	Bidirectional	F4, D3, D2	Exponent Bus Input Port C. Gain control output.
EXPD[0:2]	Bidirectional	D4, C2, F5	Exponent Bus Input Port D. Gain control output.
CLKA, CLKB	Input	K1, L1	LVDS Differential Clock for LVDS_A Input Port (LVDS_CLKA+, LVDS_CLKA–).
CLKC, CLKD	Input	A6, A5	LVDS Differential Clock for LVDS_C Input Port (LVDS_CLKC+, LVDS_CLKC–).
INA[0:15], INB[0:15]	LVDS Input	See Table 9	In LVDS input mode, INA[0:15] and INB[0:15] form a differential pair LVDS_A+[0:15] (positive node) and LVDS_A–[0:15] (negative node), respectively.
INC[0:15], IND[0:15]	LVDS Input	See Table 9	In LVDS input mode, INC[0:15] and IND[0:15] form a differential pair LVDS_C+[0:15] (positive node) and LVDS_C–[0:15] (negative node), respectively.
OUTPUT PORTS			
PCLK	Bidirectional	E16	Parallel Output Port Clock. Master mode output, and slave mode input.
PA[0:15]	Output	See Table 9	Parallel Output Port A Data Bus.
PACH[0:2]	Output	G15, D16, H12	Channel Indicator Output Port A.
PAIQ	Output	H13	Parallel Port A I/Q Data Indicator. Logic 1 indicates I data on data bus.
PAGAIN	Output	G13	Parallel Port A Gain Word Output Indicator. Logic 1 indicates gain word on data bus.
PAACK	Input	H14	Parallel Port A Acknowledge (Active High).
PAREQ	Output	F15	Parallel Port A Request (Active High).
PB[0:15]	Output	See Table 9	Parallel Output Port B Data Bus.
PBCH[0:2]	Output	C13, D11, B14	Channel Indicator Output Port B.
PBIQ	Output	D12	Parallel Port B I/Q Data Indicator. Logic 1 indicates I data on data bus.
PBGAIN	Output	A14	Parallel Port B Gain Word Output Indicator. Logic 1 indicates gain word on data bus.
PBACK	Input	E12	Parallel Port B Acknowledge (Active High).
PBREQ	Output	E11	Parallel Port B Request (Active High).
PC[0:15]	Output	See Table 9	Parallel Output Port C Data Bus.
PCCH[0:2]	Output	M15, L14, N15	Channel Indicator Output Port C.
PCIQ	Output	P15	Parallel Port C I/Q Data Indicator. Logic 1 indicates I data on data bus.
PCGAIN	Output	P16	Parallel Port C Gain Word Output Indicator. Logic 1 indicates gain word on data bus.
PCACK	Input	L13	Parallel Port C Acknowledge (Active High).
PCREQ	Output	R16	Parallel Port C Request (Active High).
MISC PINS			
$\overline{\text{RESET}}$	Input	P3	Master Reset (Active Low).
$\overline{\text{IRP}}$[1]	Output	T2	Interrupt Pin (Open Drain Output, Needs External Pull-Up Resistor 1 kΩ).
SYNC[0:3]	Input	B12, A12, C10, B11	Synchronization Inputs. SYNC pins are independent of channels or input ports and independent of each other.
LVDS_RSET	Input	E4	LVDS Resistor Set Pin (Analog Pin). See Design Notes.
EXT_FILTER	Input	R4	PLL Loop Filter (Analog Pin). See Design Notes.
MICROPORT CONTROL			
D[0:15]	Bidirectional	See Table 9	Bidirectional Microport Data. This bus is three-stated when $\overline{\text{CS}}$ is high.
A[0:7]	Input	See Table 9	Microport Address Bus.
$\overline{\text{DS}}$ ($\overline{\text{RD}}$)	Input	P4	Active Low Data Strobe when MODE = 1. Active low read strobe when MODE = 0.
$\overline{\text{DTACK}}$ (RDY)[1]	Output	M6	Active Low Data Acknowledge when MODE = 1. Microport status pin when MODE = 0. Open drain output, needs external pull-up resistor 1 kΩ.
R/$\overline{\text{W}}$ ($\overline{\text{WR}}$)	Input	N4	Read/Write Strobe when MODE = 1. Active low write strobe when MODE = 0.
MODE	Input	T3	Mode Select Pin. When SMODE = 0: Logic 0 = Intel mode; Logic 1 = Motorola mode. When SMODE = 1: Logic 0 = SPI mode; Logic 1 = SPORT mode.
$\overline{\text{CS}}$	Input	N5	Active Low Chip Select. Logic 1 three-states the microport data bus.
CPUCLK	Input	R1	Microport CLK Input (Input Only).
CHIPID[0:3]	Input	T4, R5, N6, P6	Chip ID Input Pins.

Mnemonic	Type	Pin No.	Function
SERIAL PORT CONTROL			
SCLK	Input	R1	Serial Clock.
SDO[1]	Output	M6	Serial Port Data Output (Open drain output, needs external pull-up resistor 1KΩ).
SDI[2]	Input	N11	Serial Port Data Input.
STFS	Input	N4	Serial Transmit Frame Sync.
SRFS	Input	P4	Serial Receive Frame Sync.
$\overline{SCS}$	Input	N5	Serial Chip Select.
MSB_FIRST	Input	R3	Select MSB First into SDI Pin and MSB First Out of SDO Pin. Logic 0 = MSB first; Logic 1 = LSB first.
SMODE	Input	P5	Serial Mode Select. Pull high when serial port is used and low when microport is used.
JTAG			
$\overline{TRST}$[1]	Input	B13	Test Reset Pin. Pull low when JTAG is not used.
TCLK[2]	Input	C12	Test Clock.
TMS[1]	Input	C11	Test Mode Select.
TDO	Output	A13	Test Data Output. Three-stated when JTAG is in reset.
TDI[1]	Input	D10	Test Data Input.

[1] Pin with a pull-up resistor of nominal 70 kΩ.
[2] Pin with a pull-down resistor of nominal 70 kΩ.

PIN LISTING FOR POWER, GROUND, DATA, AND ADDRESS BUSES

Table 9.

Mnemonic	Pin No.
VDDCORE	A9, G6, G11, H1, H6, H11, J6, J11, J16, K6, K11, T8
VDDIO	B2, B15, F7, F8, F9, F10, L7, L8, L9, L10, R2, R15
GND	A1, A8, A16, E5, F6, F11, G7, G8, G9, G10, H7, H8, H9, H10, H16, J1, J7, J8, J9, J10, K7, K8, K9, K10, L6, L11, M5, P7, T1, T9, T10, T15, T16
INA[0:15]	N3, P2, P1, N2, N1, M1, L2, K3, K2, J2, H2, G1, F1, F2, E1, E2
INB[0:15]	M4, L4, M3, L5, L3, M2, K4, K5, J4, J5, J3, H4, H3, G2, H5, G3
INC[0:15]	C3, C4, B3, A2, D6, C6, E7, D7, E8, D8, C8, E9, D9, C9, B10, E10
IND[0:15]	B1, E6, D5, C5, A3, B4, B5, A4, B6, C7, B7, A7, B8, B9, A10, A11
PA[0:15]	F16, H15, G16, J12, J15, J14, K16, J13, K15, K14, L16, M16, K12, L15, N16, K13
PB[0:15]	F13, E15, G14, G12, E13, E14, F12, F14, C14, D14, C16, A15, B16, D15, D13, C15
PC[0:15]	M14, N14, M13, L12, P14, N13, R14, M12, T14, R13, P13, P12, M11, T13, T12, N12
D[0:15]	R10, N9, N8, T7, P9, M9, R9, T5, T6, P8, R7, R8, N7, M7, R6, M8
A[0:7]	N11, R12, P11, R11, N10, M10, P10, T11

TIMING DIAGRAMS



Figure 3. Reset Timing Requirements

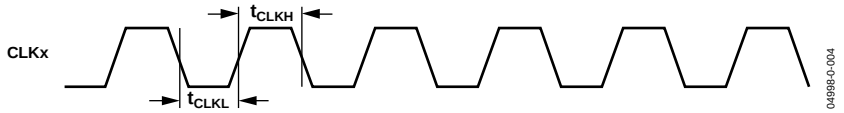


Figure 4. CLK Switching Characteristics
(x = A, B, C, D for Individual Input Ports)

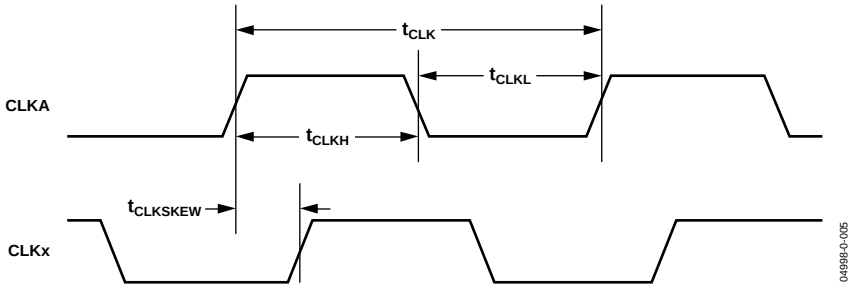


Figure 5. CLK Skew Characteristics
(x = B, C, D for Individual Input Ports)

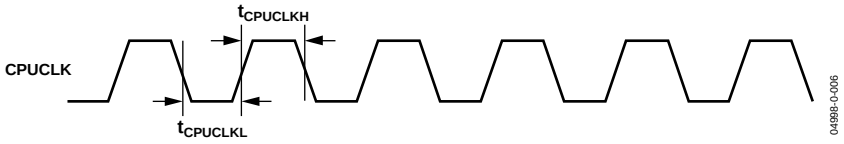


Figure 6. CPUCLK Switching Characteristics

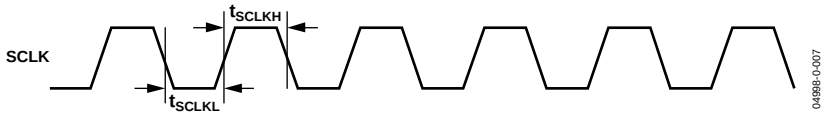


Figure 7. SCLK Switching Characteristics

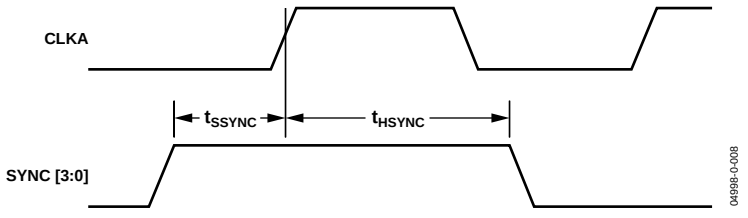


Figure 8. SYNC Timing Inputs

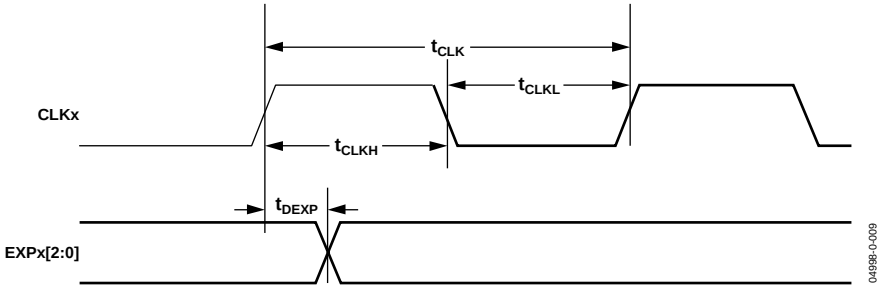


Figure 9. Gain Control Word Output Switching Characteristics
(x = A, B, C, D for Individual Input Ports)

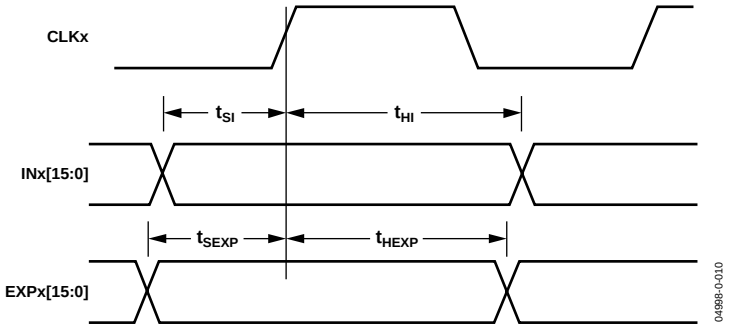


Figure 10. Input Port Timing for Data
(x = A, B, C, D for Individual Input Ports)

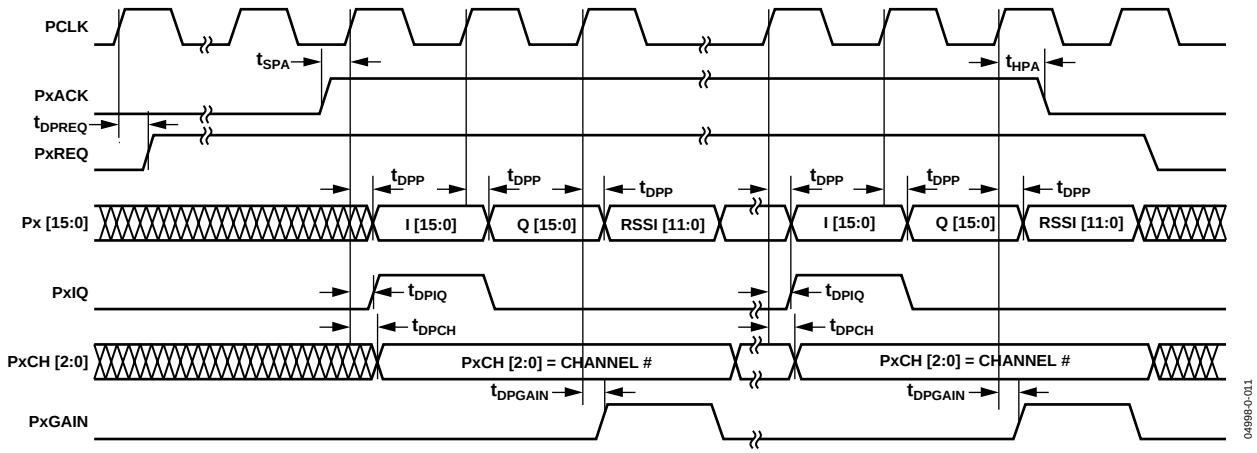


Figure 11. Master Mode PxACK to PCLK Switching Characteristics
(x = A, B, C, D for Individual Output Ports)

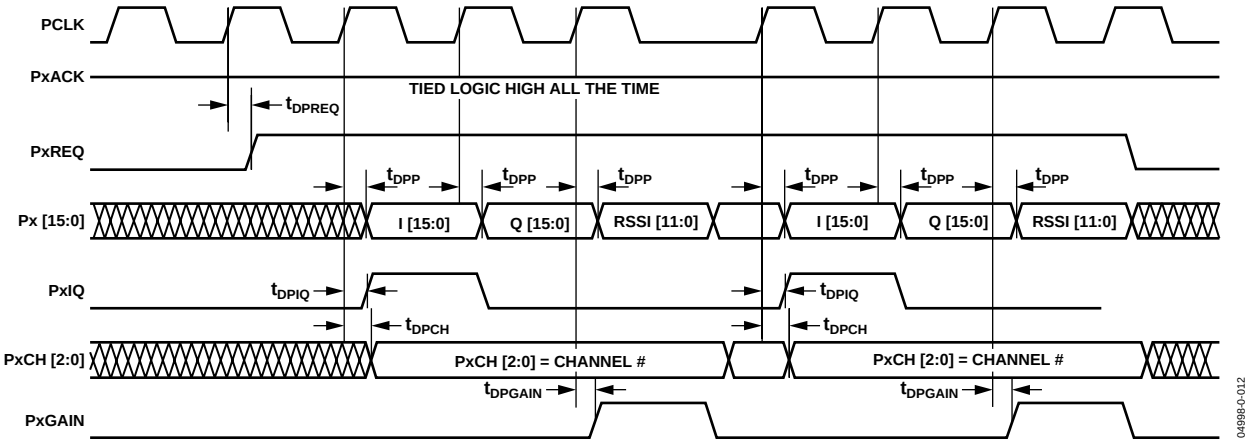
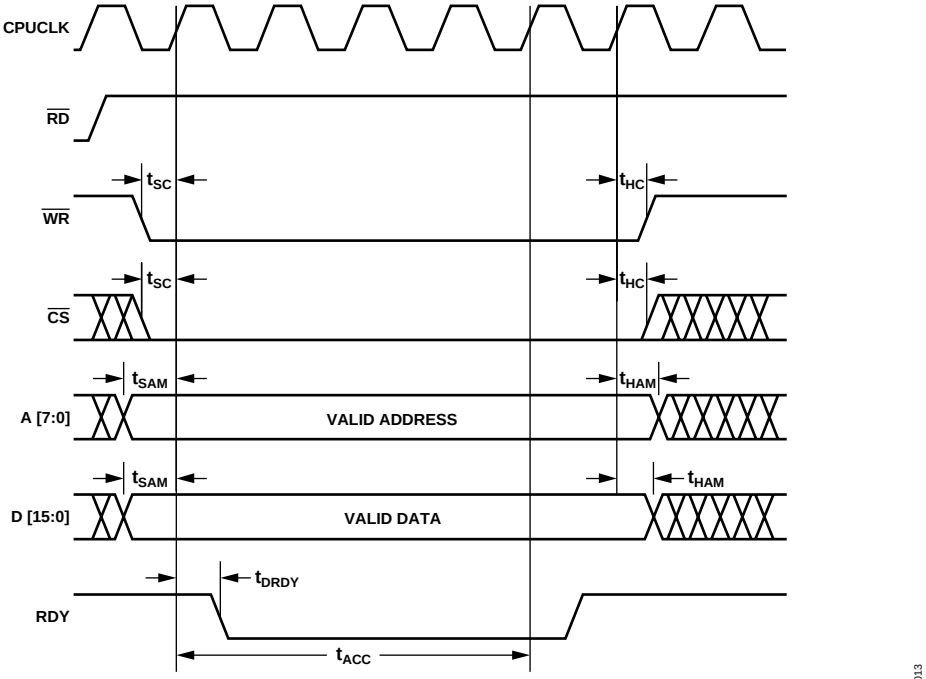
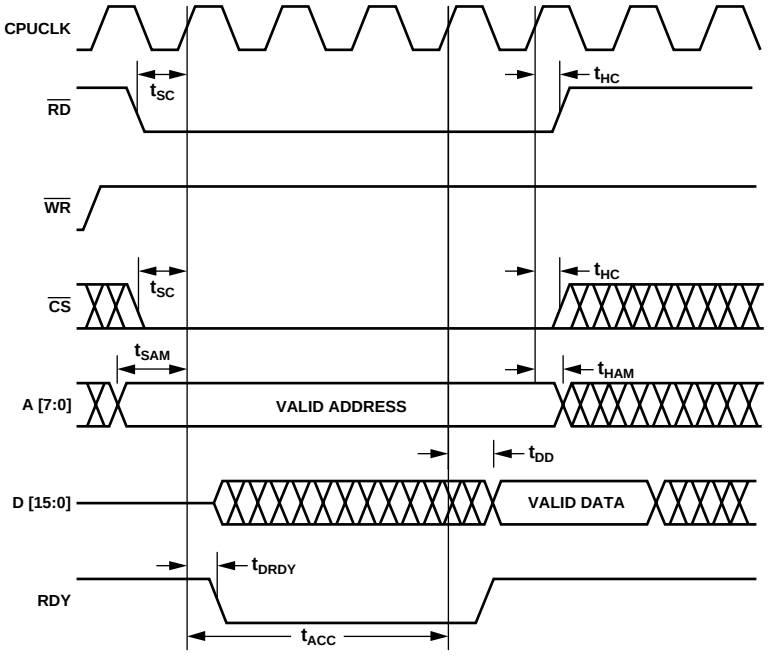


Figure 12. Master Mode PxREQ to PCLK Switching Characteristics



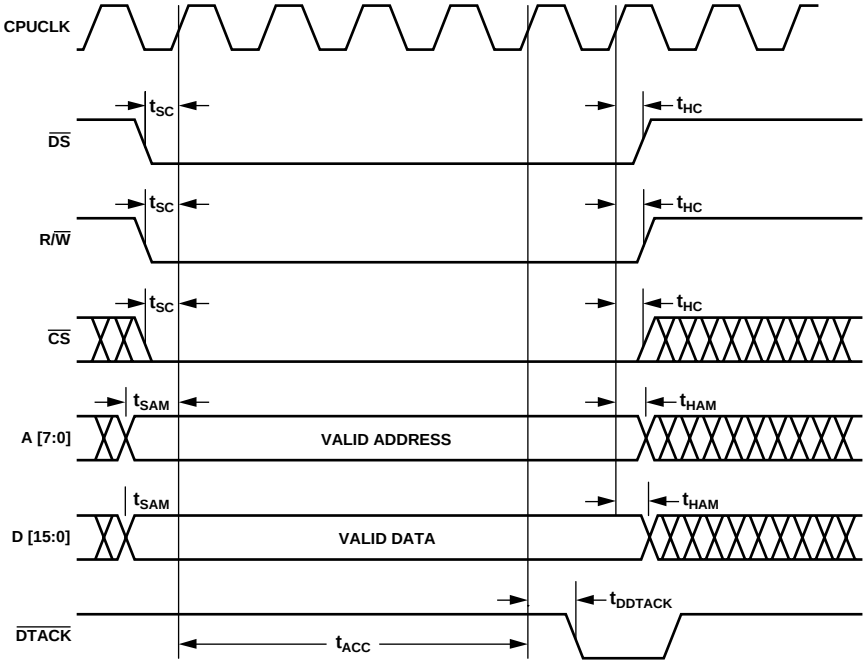
NOTE:
t_{ACC} ACCESS TIME DEPENDS ON THE ADDRESS ACCESSED. IT CAN VARY FROM 3 TO 9 CPUCLK CYCLES.

Figure 13. INM Microport Write Timing Requirements



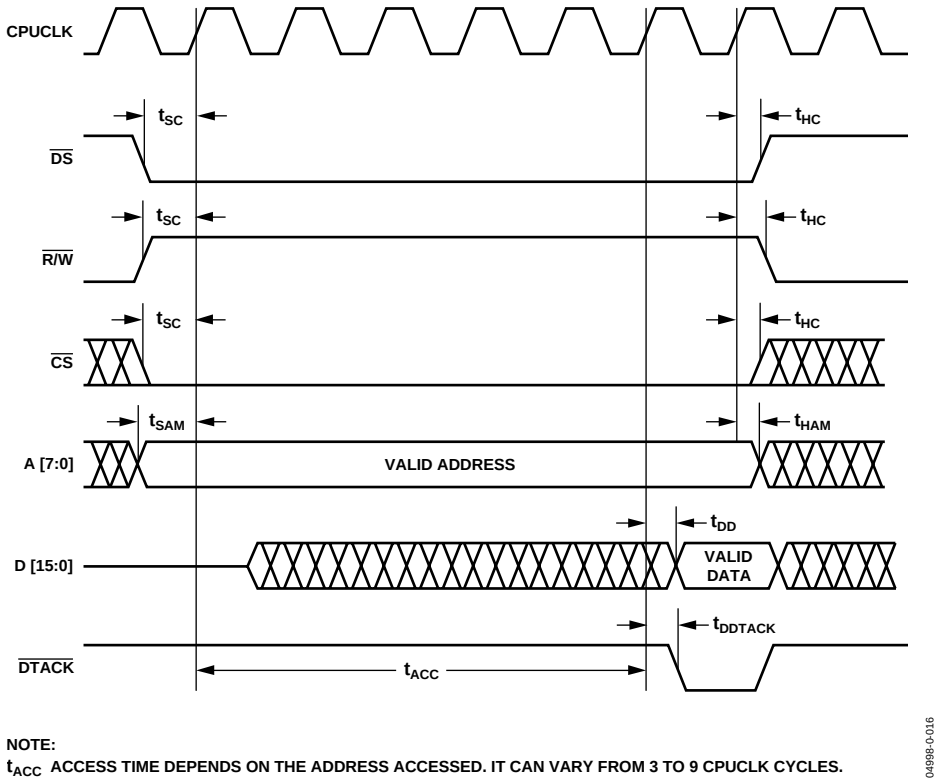
NOTE:
t_{ACC} ACCESS TIME DEPENDS ON THE ADDRESS ACCESSED. IT CAN VARY FROM 3 TO 9 CPUCLK CYCLES.

Figure 14. INM Microport Read Timing Requirements



NOTE:
t_{ACC} ACCESS TIME DEPENDS ON THE ADDRESS ACCESSED. IT CAN VARY FROM 3 TO 9 CPUCLK CYCLES.

Figure 15. MNM Microport Write Timing Requirements



NOTE:
t_{ACC} ACCESS TIME DEPENDS ON THE ADDRESS ACCESSED. IT CAN VARY FROM 3 TO 9 CPUCLK CYCLES.

Figure 16. MNM Microport Read Timing Requirements

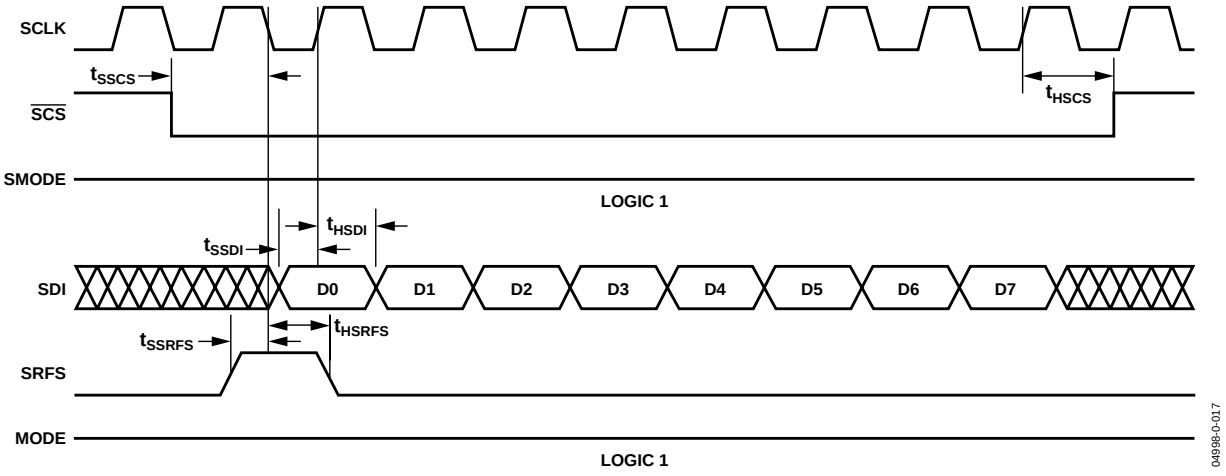


Figure 17. SPORT Mode Write Timing Characteristics

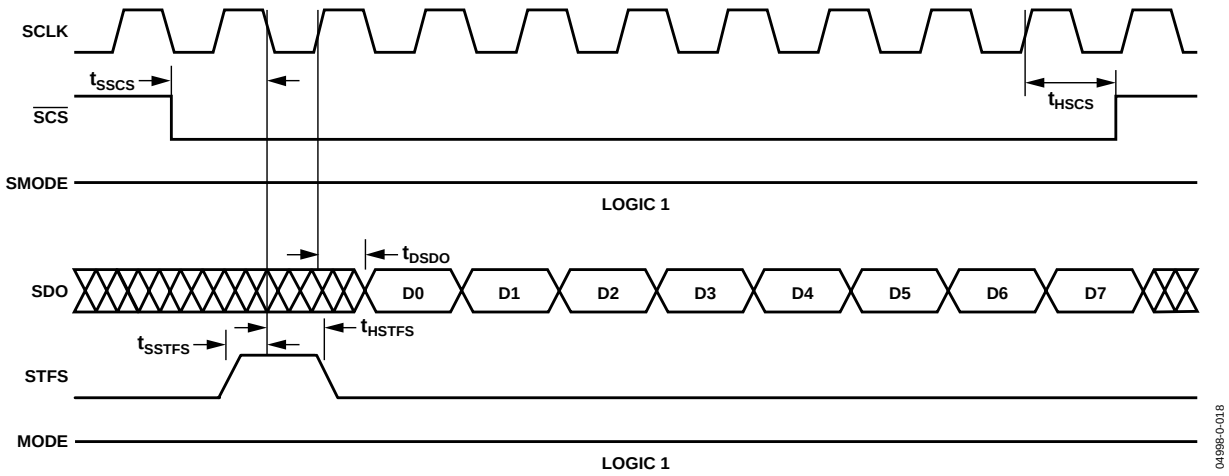


Figure 18. SPORT Mode Read Timing Characteristics

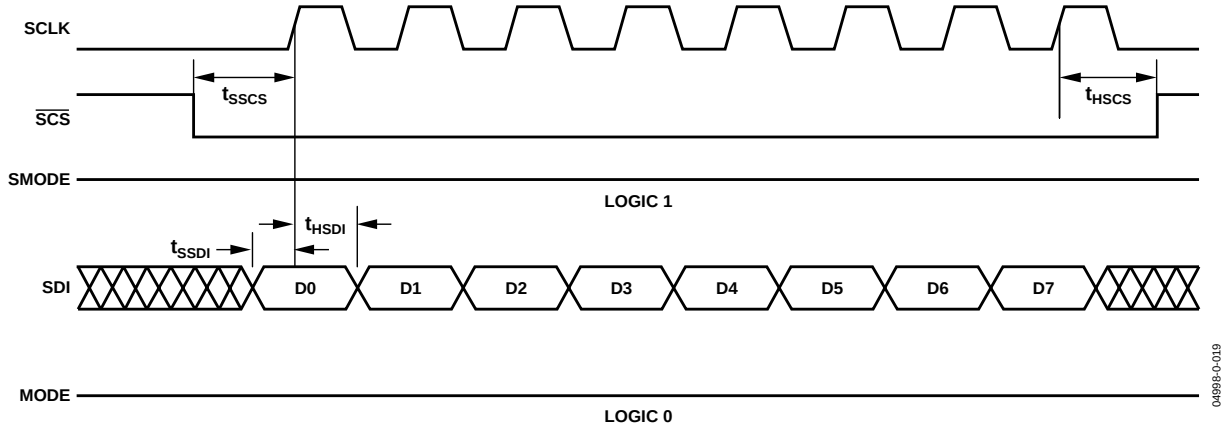


Figure 19. SPI Mode Write Timing Characteristics

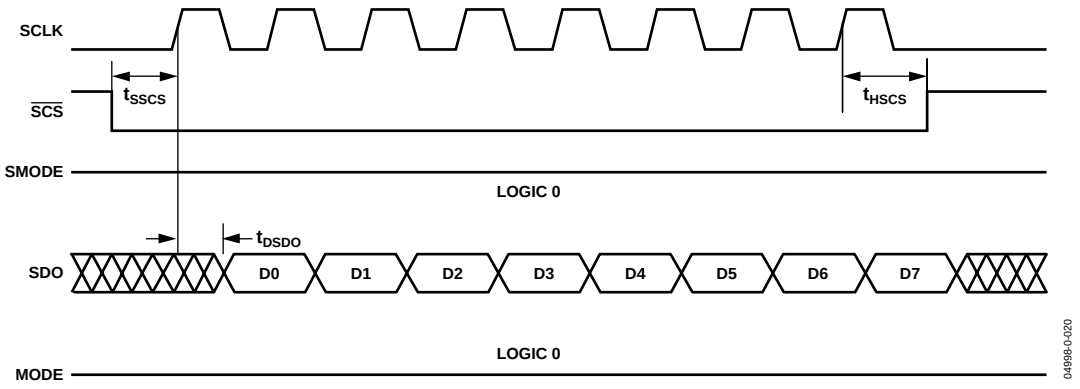


Figure 20. SPI Mode Read Timing Characteristics

THEORY OF OPERATION

ADC INPUT PORT

The AD6636 features four identical, independent high speed ADC input ports named A, B, C, and D. These input ports have the flexibility to allow independent inputs, diversity inputs, or complex I/Q inputs. Any of the ADC input ports can be routed to any of the six tuner channels; that is, any of the six. The AD6636 channels can receive input data from any of the input ports. Time-multiplexed inputs on a single port are not supported in the AD6636.

These four input ports can operate at up to 150 MSPS. Each input port has its own clock (CLKA, CLKB, CLKC, and CLKD) used for registering input data into the AD6636. To allow slow input rates while providing fast processing clock rates, the AD6636 contains an internal PLL clock multiplier that supplies the internal signal processing clock. CLKA is used as an input to the PLL clock multiplier. Additional programmability allows the input data to be clocked into the part either on the rising edge or the falling edge of the input clock.

In addition, the front end of the AD6636 contains circuitry that enables high speed signal-level detection, gain control, and quadrature I/Q correction. This is accomplished with a unique high speed level-detection circuit that offers minimal latency and maximum flexibility to control all four input signals (typically ADC inputs) individually. The input ports also provide input power-monitoring functions via various modes and magnitude and phase I/Q correction blocks. See the Quadrature I/Q Correction Block section for details.

Each individual processing channel can receive input data from any of the four input ports individually. This is controlled using 3-bit crossbar mux-select bit words in the ADC input control register. Each individual channel has a similar 3-bit selection. In addition to the four input ports, an internal test signal (PN—pseudorandom noise sequence) can also be selected. This internal test signal is discussed in the User-Configurable, Built-In Self-Test (BIST) section.

Input Data Format

Each input port consists of a 16-bit mantissa and a 3-bit exponent (16 + 3 floating-point input, or up to 16-bit fixed-point input). When interfacing to standard fixed-point ADCs, the exponent bit should either be connected to ground or be programmed as outputs for gain control output. If connected to a floating-point ADC (also called gain ranging ADC), the exponent bits from the ADC can be connected to the input exponent bits of the AD6636. The mantissa data format is twos complement, and the exponent is unsigned binary.

The 3-exponent bits are shared with the gain range control bits in the hardware. When floating-point ADCs are not used, these three pins on each ADC input port can be used as gain range control output bits.

Input Timing

The data from each high speed input port is latched either on the rising edge or the falling edge of the port's individual CLKx (where x stands for A, B, C, or D input ports). The ADC clock invert bit in ADC clock control register selects the edge of the clock (rising or falling) used to register input data into the AD6636.

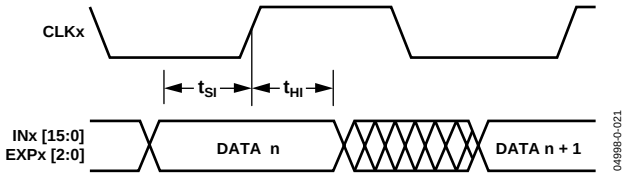


Figure 21. Input Data Timing Requirements (Rising Edge of Clock, x = A, B, C, or D for Four Input Ports)

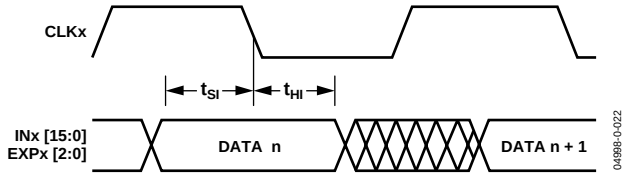


Figure 22. Input Data Timing Requirements (Falling Edge of Clock, x = A, B, C, or D for Four Input Ports)

The clock signals (CLKA, CLKB, CLKC, and CLKD) can operate at up to 150 MHz. In applications using high speed ADCs, the ADC sample clock, data valid, or data-ready strobe are typically used to clock the AD6636.

Connection to Fixed-Point ADC

For fixed-point ADCs, the AD6636 exponent inputs, EXP[2:0], are not typically used and should be tied low. Alternatively, because these pins are shared with gain range control bits, if the gain ranging block is used, these pins can be used as outputs of the gain range control block. The ADC outputs are tied directly to the AD6636 inputs, MSB justified. Therefore, for fixed-point ADCs, the exponents are typically static and no input scaling is used in the AD6636. Figure 23 shows a typical interconnection.

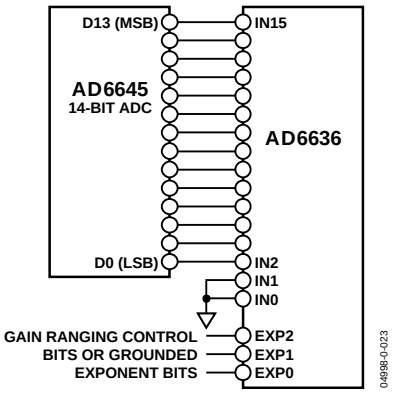


Figure 23. Typical Interconnection of the AD6645 Fixed-Point ADC and AD6636

Scaling with Floating-Point ADC

An example of the exponent control feature combines the AD6600 and the AD6636. The AD6600 is an 11-bit ADC with three bits of gain ranging. In effect, the 11-bit ADC provides the mantissa, and the three bits of the relative signal strength indicator (RSSI) are the exponent. Only five of the eight available steps are used by the AD6600. See the AD6600 data sheet for details.

Table 10. Weighting Factors for Different Exp[2:0] Values

ADC Input Level	AD6636 Exp[2:0]	Data Divide-By	Signal Attenuation (dB)
Largest	000 (0)	/1 (>> 0)	0
	001 (1)	/2 (>> 1)	6
	010 (2)	/4 (>> 2)	12
	011 (3)	/8 (>> 3)	18
	100 (4)	/16 (>> 4)	24
	101 (5)	/32 (>> 5)	30
	110 (6)	/64 (>> 6)	36
Smallest	111 (7)	/128 (>> 7)	42

Complex (I/Q) Input Ports

The four individual ADC input ports of the AD6636 can be configured to function as two complex input ports. Additionally, if required, only two input ports can be made to function as a complex port, while the remaining two input ports function as real individual input ports.

In complex mode, Input Port A is paired with Input Port B to receive I and Q data, respectively. Similarly, Input Port C can be paired with Input Port D to receive I and Q data, respectively. These two pairings are controlled individually using Bit 24 and Bit 25 of the ADC input control register.

As explained previously, each individual channel can receive input signals from any of the four input ports using the crossbar mux select bits in the ADC input control register. In addition to the three bits, a 1-bit selection is provided for choosing the complex input port option for any individual channel. For example, if Channel 0 needs to receive complex input from Input Port A and Input Port B, the mux select bits should indicate Input Port A, and the complex input bit should be selected.

When the input ports are paired for complex input operation, only one set of exponent bits is driven externally with gain control output. Therefore, when Input Port A and Input Port B form a complex input, EXPA[2:0] are output and, similarly, for Input Port C and Input Port D, EXPC[2:0] are output.

LVDS Input Ports

The AD6636 input ports can be configured in CMOS mode or LVDS mode. In CMOS input mode, the four input ports can be configured as two complex input ports. In LVDS mode, two CMOS input ports are each combined to form one LVDS input port.

CMOS Input Port INA[15:0] and CMOS Input Port INB[15:0] form the positive and negative differential nodes, LVDS_A+[15:0] and LVDS_A–[15:0], respectively. Similarly, INC[15:0] and IND[15:0] form the positive and negative differential nodes, LVDS_C+[15:0] and LVDS_C– [15:0], respectively. CLKA and CLKB form the differential pair, Pin LVDS_CLKA+ and Pin LVDS_CLKA–. Similarly, CLKC and CLKD form the differential pair Pin LVDS_CLKC+ and Pin LVDS_CLKC–.

By default, the AD6636 powers up in CMOS mode and can be programmed to CMOS mode by using the CMOS mode bit (Bit 10 of the LVDS control register). Writing Logic 1 to Bit 8 of the LVDS control register enables an autocalibrate routine that calibrates the impedance of the LVDS pads to match the output impedance of the LVDS signal source impedance. The LVDS pads in the AD6636 have an internal impedance of 100 Ω across the differential signals; therefore, an external resistor is not required.

PLL CLOCK MULTIPLIER

In the AD6636, the input clock rate must be the same as the input data rate. In a typical digital downconverter architecture, the clock rate is a limitation on the number of filter taps that can be calculated in the programmable RAM coefficient filters (MRCF, DRCF, and CRCF). For slower ADC clock rates (or for any clock rate), this limitation can be overcome by using a PLL clock multiplier to provide a higher clock rate to the RCF filters. Using this clock multiplier, the internal signal processing clock rate can be increased up to 200 MHz. The CLKA signal is used as an input to the PLL clock multiplier.

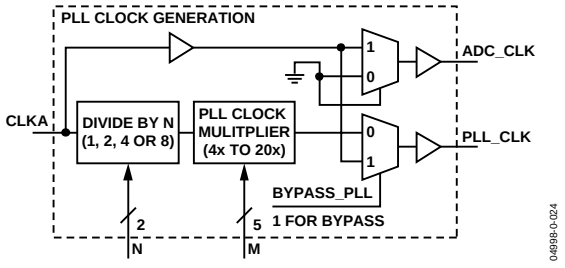


Figure 24. PLL Clock Generation

The PLL clock multiplier is programmable and uses input clock rates between 4 MHz and 150 MHz to give a system clock rate (output) of as high as 200 MHz.

The output clock rate is given by

$$PLL_CLK = \frac{CLKA \times M}{N}$$

where:

CLKA is the Input Port A clock rate.

M is a 5-bit programmable multiplication factor.

N is a predivide factor.

M is a 5-bit number between 4 and 20 (both values included). *N* (predivide) can be 1, 2, 4, or 8. The multiplication factor *M* is programmed using a 5-bit PLL clock multiplier word in the ADC clock control register. A value outside the valid range of 4 to 20 bypasses the PLL clock multiplier and, therefore, the PLL clock is the same as the input clock. The predivide factor *N* is programmed using a 2-bit ADC pre-PLL clock divider word in the ADC clock control register, as listed in Table 11.

Table 11. PLL Clock Generation Predivider Control

Predivide Word [1:0]	Divide-by Value for the Clock
00	Divide-by-1, bypass
01	Divide-by-2
10	Divide-by-4
11	Divide-by-8

For the best signal processing advantage, the user should program the clock multiplier to give a system clock output as close as possible to, but not exceeding, 200 MHz. The internal blocks of the AD6636 that run off of the PLL clock are rated to run at a maximum of 200 MHz. The default power-up state for the PLL clock multiplier is the bypass state, where CLKA is passed on as the PLL clock.

ADC GAIN CONTROL

Each ADC input port has individual, high speed, gain-control logic circuitry. Such gain-control circuitry is useful in applications that involve large dynamic range inputs or in which gain ranging ADCs are employed. The AD6636 gain-control logic allows programmable upper and lower thresholds and a programmable dwell-time counter for temporal hysteresis.

Each input port has a 3-bit output from the gain control block. These three output pins are shared with the 3-bit exponent input pins for each input port. The operation is controlled by the gain control enable bit in the gain control register of the individual input ports. Logic 1 in this bit programs the EXP[2:0] pins as gain-control outputs, and Logic 0 configures the pins as input exponent pins. To avoid bus contention, these pins are set, by default, as input exponent pins.

Function

The gain-control block features a programmable upper threshold register and a lower threshold register. The ADC input data is compared to both these registers. If ADC input data is larger than the upper threshold register, then the gain control output is decremented by 1. If ADC input data is smaller than the lower threshold register, then the gain control output is incremented by 1. When decrementing the gain control output, the change is immediate. But when incrementing the output, a dwell-time register is used to delay the change. If the ADC input is larger than the upper threshold register value, the gain-control output is decremented to prevent overflow immediately.

When the ADC input is lower than the lower threshold register, a dwell timer is loaded with the value in the programmable, 20-bit, dwell-time register. The counter decrements once every input clock cycle, as long as the input signal remains below the lower threshold register value. If the counter reaches 1, the gain control output is incremented by 1. If the signal goes above the lower threshold register value, the gain adjustment is not made, and the normal comparison to lower and upper threshold registers is initiated once again. Therefore, the dwell timer provides temporal hysteresis and prevents the gain from switching continuously.

In a typical application, if the ADC signal goes below the lower threshold for a time greater than the dwell time, then the gain control output is incremented by 1. Gain control bits control the gain ranging block, which appears before the ADC in the signal chain. With each increment of the gain control output, gain in the gain-ranging block is increased by 6.02 dB. This increases the dynamic range of the input signal into the ADC by 6.02 dB. This gain is compensated for in the AD6636 by relinearizing (see the Relinearization section). Therefore, the AD6636 can increase the dynamic range of the ADC by 42 dB, provided that the gain-ranging block can support it.

Relinearization

The gain in the gain-ranging block (external) is compensated for by relinearizing, using the exponent bits, EXP[2:0], of the input port. For this purpose, the gain control bits are connected to the EXP[2:0] bits, providing an attenuation of 6.02 dB for every increase in the gain control output. After the gain in the external gain-ranging block and the attenuation in the AD6636 (using EXP bits), the signal gain is essentially unchanged. The only change is the increase in the dynamic range of the ADC.

External gain-ranging blocks or gain-ranging ADCs have a delay associated with changing the gain of the signal. Typically, these delays can be up to 14 clock cycles. The gain change in the AD6636 (via EXP[2:0]) must be synchronized with the gain change in the gain-ranging block (external). This is allowed in the AD6636 by providing a flexible delay, programmable 6-bit word in the gain control register. The value in this 6-bit word

gives the delay in input clock cycles. A programmable pipeline delay given by the 6-bit value (maximum delay of 63 clock cycles) is placed between the gain control output and the EXP[2:0] input. Therefore, the external gain-ranging block's settling delays are compensated for in the AD6636.

Note that any gain changes that are initiated during the relinearization period are ignored. For example, if the AD6636 detects that a gain adjustment is required during the relinearization period of a previous gain adjustment, then the new adjustment is ignored.

Setting Up the Gain Control Block

To set up the gain control block for individual input ports, the individual upper threshold registers and lower threshold registers should be written with appropriate values. The 10-bit values written into upper and lower threshold registers are compared to the 10 MSB bits of the absolute magnitude calculated using the input port data. The 20-bit dwell timer register should have the appropriate number of clock cycles to provide temporal hysteresis.

A 6-bit relinearization pipeline delay word is set to synchronize with the settling delay in the external gain ranging circuitry. Finally, the gain control enable bit is written with Logic 1 to activate the gain control block. On enabling, the gain control output bits are made 000 (output on EXP[2:0] pins), which represent the minimum gain for the external gain-ranging circuitry and corresponding minimum attenuation during relinearization. The normal functioning takes over, as explained previously in this section.

Complex Inputs

For complex inputs (formed by pairing two input ports), only one set of EXP[2:0] pins should be used as the gain control output. For the pair of Input Port A and Input Port B, gain control circuitry for Input Port A is active, and EXPA[2:0] should be connected externally as the gain control output. The gain control circuitry for Input Port B is not activated (shut down), and EXPB[2:0] is forced to be equal to EXP[2:0].

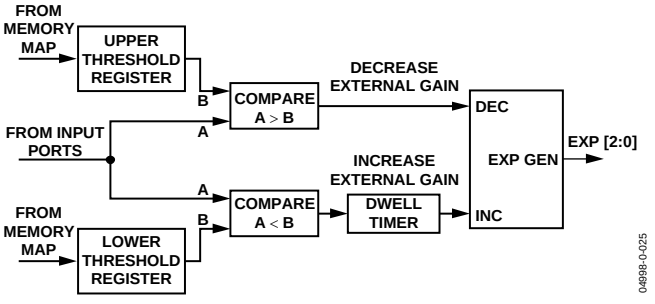


Figure 25. AD6636 Gain Control Block Diagram

ADC INPUT PORT MONITOR FUNCTION

The AD6636 provides a power-monitor function that can monitor and gather statistics about the received signal in a signal chain. Each input port is equipped with an individual power-monitor function that can operate both in real and complex modes of the input port. This function block can operate in one of three modes, which measure the following over a programmable period of time:

- Peak power
- Mean power
- Number of samples crossing a threshold

These functions are controlled via the 2-bit power-monitor function select bits of the power monitor control register for each individual input port. The input ports can be set for different modes, but only one function can be active at a time for any given input port.

The three modes of operation can function continuously over a programmable time period. This time period is programmed as the number of input clock cycles in a 24-bit ADC monitor period register (AMPR). This register is separate for each input port. An internal magnitude storage register (MSR) is used to monitor, accumulate, or count, depending on the mode of operation.

Peak Detector Mode (Control Bits 00)

The magnitude of the input port signal is monitored over a programmable time period (given by AMPR) to give the peak value detected. This mode is set by programming Logic 0 in the power-monitor function select bits of the power-monitor control register for each individual input port. The 24-bit AMPR must be programmed before activating this mode.

After enabling this mode, the value in the AMPR is loaded into a monitor period timer and the countdown is started. The magnitude of the input signal is compared to the MSR, and the greater of the two is updated back into the MSR. The initial value of the MSR is set to the current ADC input signal magnitude. This comparison continues until the monitor period timer reaches a count of 1.

When the monitor period timer reaches a count of 1, the value in the MSR is transferred to the power-monitor holding register, which can be read through the microport or the serial port. The monitor period timer is reloaded with the value in the AMPR, and the countdown is started. Also, the first input sample's magnitude is updated in the MSR, and the comparison and update procedure, as explained above, continues. If the interrupt is enabled, an interrupt is generated, and the interrupt status register is updated when the AMPR reaches a count of 1.

Figure 26 is a block diagram of the peak detector logic. The MSR contains the absolute magnitude of the peak detected by the peak detector logic.

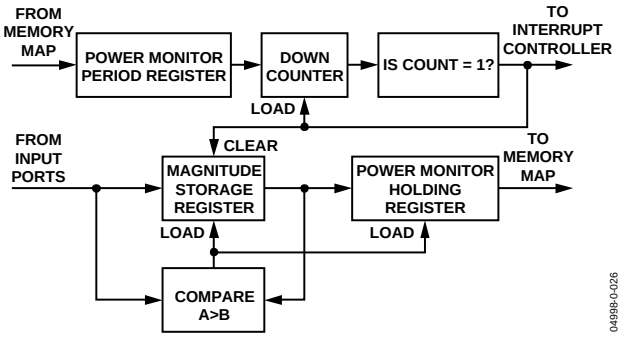


Figure 26. ADC Input Peak Detector Block Diagram

Mean Power Mode (Control Bits 01)

In this mode, the mean power of the input port signal is integrated (by adding an accumulator) over a programmable time period (given by AMPR) to give the mean power of the input signal. This mode is set by programming Logic 1 in the power monitor function select bits of the power monitor control register for each individual input port. The 24-bit AMPR, representing the period over which integration is performed, must be programmed before activating this mode.

After enabling this mode, the value in the AMPR is loaded into a monitor period timer, and the countdown is started immediately. The 15-bit mean power of input signal is right-shifted by nine bits to give 6-bit data. This 6-bit data is added to the contents of a 24-bit holding register, thus performing an accumulation. The integration continues until the monitor period timer reaches a count of 1.

When the monitor period timer reaches a count of 1, the value in the MSR is transferred to the power-monitor holding register (after some formatting), which can be read through the microport or the serial port. The monitor period timer is reloaded with the value in the AMPR, and the countdown is started. Also, the first input sample signal power is updated in the MSR, and the accumulation continues with the subsequent input samples. If the interrupt is enabled, an interrupt is generated, and the interrupt status register is updated when the AMPR reaches a count of 1. Figure 27 illustrates the mean power-monitoring logic.

The value in the MSR is a floating-point number with 4 MSBs and 20 LSBs. If the 4 MSBs are EXP and the 20 LSBs are MAG, the value in dBFS can be decoded by

$$Mean\ Power = 10 \log \left[\left(\frac{MAG}{2^{20}} \right) 2^{-(EXP-1)} \right]$$

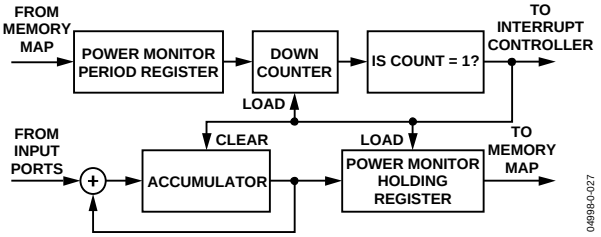


Figure 27. ADC Input Mean Power-Monitoring Block Diagram

Threshold Crossing Mode (Control Bits 10)

In this mode of operation, the magnitude of the input port signal is monitored over a programmable time period (given by AMPR) to count the number of times it crosses a certain programmable threshold value. This mode is set by programming Logic 1x (where x is a don't care bit) in the power-monitor function select bits of the power monitor control register for each individual input port. Before activating this mode, the user needs to program the 24-bit AMPR and the 10-bit upper threshold register for each individual input port. The same upper threshold register is used for both power monitoring and gain control (see the ADC Gain Control section).

After entering this mode, the value in the AMPR is loaded into a monitor period timer, and the countdown is started. The magnitude of the input signal is compared to the upper threshold register (programmed previously) on each input clock cycle. If the input signal has magnitude greater than the upper threshold register, then the MSR register is incremented by 1. The initial value of the MSR is set to 0. This comparison and increment of the MSR register continues until the monitor period timer reaches a count of 1.

When the monitor period timer reaches a count of 1, the value in the MSR is transferred to the power monitor holding register, which can be read through the microport or the serial port. The monitor period timer is reloaded with the value in the AMPR, and the countdown is started. The MSR register is also cleared to a value of 0. If interrupts are enabled, an interrupt is generated, and the interrupt status register is updated when the AMPR reaches a count of 1. Figure 28 illustrates the threshold crossing logic. The value in the MSR is the number of samples that have an amplitude greater than the threshold register.

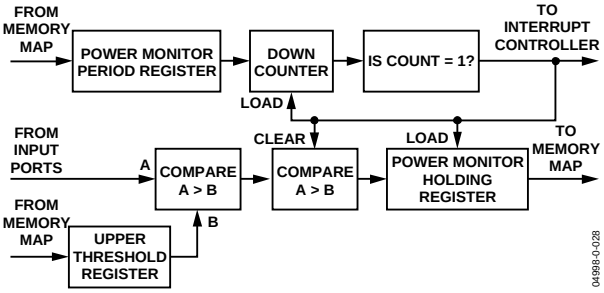


Figure 28. ADC Input Threshold Crossing Block Diagram

Additional Control Bits

For additional flexibility in the power monitoring process, two control bits are provided in the power-monitor control register. They are the disable monitor period timer bit and the clear-on-read bit. These options have the same function in all three modes of operation.

Disable Monitor Period Timer Bit

When the disable monitor period timer bit is written with Logic 1, the timer continues to run but does not cause the contents of the MSR to be transferred to the holding register when the count reaches 1. This function of transferring the MSR to the power monitor holding register and resetting the MSR is now controlled by a read operation on the microport or serial port.

When a microport or serial port read is performed on the power monitor holding register, the MSR value is transferred to the holding register. After the read operation, the timer is reloaded with the AMPR value. If the timer reaches 1 before the microport or serial port read, the MSR value is not transferred to the holding register, as in normal operation. The timer still generates an interrupt on the AD6636 interrupt pin and updates the interrupt status register. An interrupt appears on the $\overline{\text{IRP}}$ pin, if interrupts are enabled in the interrupt enable register.

Clear-on-Read Bit

This control bit is valid when the disable monitor period timer bit is Logic 1 only. When both of these bits are set, a read operation to either the microport or the serial port reads the MSR value, and the monitor period timer is reloaded with the AMPR value. The MSR is cleared (written with current input signal magnitude in peak power and mean power mode; written with a 0 in threshold crossing mode), and normal operation continues.

When the monitor period timer is disabled and the clear-on-read bit is set, a read operation to the power monitor holding register clears the contents of the MSR and, therefore, the power monitor loop restarts.

If the clear-on-read bit is Logic 0, the read operation to the microport or serial port does not clear the MSR value after it is transferred into the holding register. The value from the previous monitor time period persists, and it continues to be compared, accumulated, or incremented, based on new input signal magnitude values.

QUADRATURE I/Q CORRECTION BLOCK

When the I and Q paths are digitized using separate ADCs, as in quadrature IF down-conversion, a mismatch often occurs between I and Q due to variations in the ADCs from the manufacturing process. The AD6636 is equipped with two quadrature correction blocks that can be used to correct I/Q mismatch errors in a complex baseband input stream. These I/Q mismatches can result in spectral distortions and removing them is useful.

Two such blocks are present, one each for the I/Q signal formed by combining the A and B inputs and the C and D inputs, respectively. The I/Q correction block can be enabled when the Port A (or Port C) complex data active bit is enabled in the ADC input control register. This block is bypassed when real input data is present on the ADC input ports because there is no possibility of I/Q mismatch in real data.

The I/Q or quadrature correction block consists of three independent subblocks: dc correction, phase correction, and amplitude correction. Three individual bits in the AB (or CD) correction control registers can be used to enable or disable each of these subblocks independently. Figure 29 shows the contents and definitions of the registers related to the quadrature correction block.

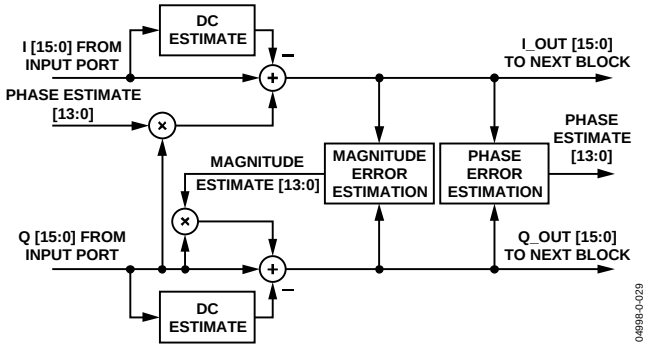


Figure 29. Quadrature Correction Block Diagram

Table 12. Correction Control Registers

Register	Bits	Description
I/Q Correction Control	15 to 12	Amplitude Loop BW
	11 to 8	Phase Loop BW
	7 to 4	DC Loop BW
	3	Reserved (Logic 0)
	2	Amplitude Correction Enable
	1	Phase Correction Enable
	0	DC Correction Enable
DC Offset Correction I	31 to 16	DC Offset Q
DC Offset Correction Q	15 to 0	DC Offset I
Amplitude Offset Correction	31 to 16	Amplitude Correction
Phase Offset Correction	15 to 0	Phase Correction

DC Correction

All ADCs have a nominal dc offset related to them. If the ADCs in the I and Q path have different dc offsets due to variations in the manufacturing process, the dc correction circuit can be used to compensate for these dc offsets. Writing Logic 1 into the dc correction enable bit of the AB (or CD) correction control register enables the dc correction block. Two dc estimation blocks are used, one each for the I and Q paths. The estimated dc value is subtracted from the I and Q paths. Therefore, the dc signal is removed independently from the I and Q path signals.

A cascade of two low-pass decimating filters estimates the dc offset in the feedback loop. A decimating first-order CIC filter is followed by an interpolating second-order CIC filter. The decimation and interpolation values of the CIC filters are the same and are programmable between 2^{12} and 2^{24} in powers of 2. The 4-bit dc loop BW word in the I/Q correction control AB (or CD) register is used to program this decimation (interpolation) value. When the dc loop BW is a 0, decimation is 2^{12}, and when the dc loop BW is 11, decimation is 2^{24}.

When the dc correction circuit is enabled, the dc correction values are estimated. The values, which are estimated independently in the I and Q paths, are subtracted independently from their respective datapaths. These dc correction values are also available for output continuously through the dc correction I and dc correction Q registers. These registers contain 16-bit dc offset values whose MSB-justified values are subtracted directly from MSB-justified ADC inputs for the I and Q paths.

When the dc correction circuit is disabled, the value in the dc correction register is used for continuously subtracting the dc offset from I and Q datapaths. This method can be used to manually set the dc offset instead of using the automatic dc correction circuit.

Phase Correction

When using complex ADC input, the I and Q datapaths typically have phase offset, caused mainly by the local oscillator and demodulator IC. The AD6636 phase-offset correction circuit can be used to compensate for this phase offset.

When the phase correction enable bit is Logic 1, the phase error between I and Q is estimated (ideally, the phase should be 90°). The phase mismatch is estimated over a period of time determined by the integrator loop bandwidth. This integrator is implemented as a first-order CIC decimating filter, whose decimation value can vary between 2^{12} and 2^{24} in powers of 2. Phase loop BW (Bits [11:8]) of the I/Q correction control register determine this decimation value. When phase loop BW equals 0, the decimation value is 2^{12}, and when phase loop BW is 11, the decimation value is 2^{24}.

While the phase offset correction circuit is enabled, the tan(phase_mismatch) is estimated continuously. This value is multiplied with Q path data and added to I path data continuously. The estimated value is also updated in the phase offset correction register. The tan(phase_mismatch) can be ±0.125 with a 14-bit resolution. This converts to a phase mismatch of about ±7.125°.

When the phase offset correction circuit is disabled, the value in the phase correction register is multiplied by the Q path data and added to the I path data continuously. This method can be used to manually set the phase offset instead of using the automatic phase offset correction circuit.

Amplitude Correction

When using complex ADC input, the I and Q datapaths typically have amplitude offset, caused mainly by the local oscillator and the demodulator IC. The AD6636 amplitude offset correction circuit can be used to compensate for this amplitude offset.

When the amplitude correction enable bit is Logic 1, the amplitude error between the I and Q datapaths is estimated. The amplitude mismatch is estimated over a period of time determined by the integrator loop bandwidth. This integrator is implemented as a first-order CIC decimating filter, whose decimation value can vary between 2^{12} and 2^{24} in powers of 2. Phase loop BW (Bits [11:8]) of the I/Q correction control register determines this decimation value. When the phase loop BW equals 0, the decimation value is 2^{12}, and when phase loop BW is 11, the decimation value is 2^{24}.

While the amplitude offset correction circuit is enabled, the difference (MAG(Q) – MAG(I)) is estimated continuously. This value is multiplied with the Q path data and added to the Q path data continuously. The estimated value is also updated in the phase offset correction register. The difference (MAG(Q) –

MAG(I)) can be between 1.125 and 0.875 with a 14-bit resolution.

When the amplitude offset correction circuit is disabled, the value in the amplitude offset correction register is multiplied by the Q path data and added to the Q path data continuously. This method can be used to manually set the amplitude offset instead of using the automatic amplitude offset correction circuit.

INPUT CROSSBAR MATRIX

The AD6636 has four ADC input ports and six channels. Two input ports can be paired to support complex input ports. Crossbar mux selection allows each channel to select its input signal from the following sources: four real input ports, two complex input ports, and internally generated pseudorandom sequence (referred to as a PN sequence, which can be either real or complex). Each channel has an input crossbar matrix to select from the above-listed input signal choices.

The selection of the input signal for a particular channel is made using a 3-bit crossbar mux select word and a 1-bit complex data input bit selection in the ADC input control register. Each channel has a separate selection for individual control. Table 13 lists the valid combinations of the crossbar mux select word, the complex data input bit values, and the corresponding input signal selections.

NUMERICALLY CONTROLLED OSCILLATOR (NCO)

Each channel consists of an independent complex NCO and a complex mixer. This processing stage has a digital tuner consisting of three multipliers and a 32-bit complex NCO. The NCO serves as a quadrature local oscillator capable of producing an NCO frequency of between −CLK/2 and +CLK/2 with a resolution of $CLK/2^{32}$ in complex mode, where CLK is the input clock frequency.

The frequency word used for generating the NCO is a 32-bit word. This word is used to generate a 20-bit phase word. A 16-bit phase offset word is added to this phase word. Eighteen bits of this phase word are used to generate the sine and cosine of the required NCO frequency.

The amplitude of the sine and cosine are represented using 17 bits. The worst-case spurious signal from the NCO is better than −100 dBc for all output frequencies.

Because the filtering in the AD6636 is low-pass filtering, the carrier of interest is tuned down to dc (frequency = 0 Hz). This is illustrated in Figure 30. Once the signal of interest is tuned down to dc, the unwanted adjacent carriers can be rejected using the low-pass filtering that follows.

NCO Frequency

The NCO frequency value is given by the 32-bit twos complement number entered in the NCO frequency register. Frequencies between −CLK/2 and CLK/2 (CLK/2 excluded) are represented using this frequency word:

0x8000 0000 represents a frequency given by −CLK/2.

0x0000 0000 represents dc (frequency is 0 Hz).

0x7FFF FFFF represents $CLK/2 - CLK/2^{32}$.

The NCO frequency word can be calculated by

$$NCO_FREQ = 2^{32}\,\frac{\mathrm{mod}\left(f_{ch}\,,\,f_{clk}\right)}{f_{clk}}$$

where:

NCO_FREQ is the 32-bit twos complement number representing the NCO frequency register.

f_{ch} is the desired carrier frequency.

f_{clk} is the clock rate for the channel under consideration.

mod( ) is a remainder function. For example, mod(110, 100) = 10 and, for negative numbers, mod(−32, 10) = −2.

Note that this equation applies to the aliasing of signals in the digital domain (that is, aliasing introduced when digitizing analog signals).

Table 13. Crossbar Mux Selection for Channel Input Signal

Complex Input Bit	Crossbar Mux Select Bit	Input Signal Selection
0	000	Input Port A magnitude and exponent pins drive the channel.
0	001	Input Port B magnitude and exponent pins drive the channel.
0	010	Input Port C magnitude and exponent pins drive the channel.
0	011	Input Port D magnitude and exponent pins drive the channel.
0	100	Internal PN sequence's magnitude and exponent bits drive the channel.
1	000	Input Ports A and B form a pair to drive I and Q paths of the channel, respectively. Input Port A exponent pins drive the channel exponent bits.
1	001	Input Ports C and D form a pair to drive I and Q paths of the channel, respectively. Input Port C exponent pins drive the channel exponent bits.
1	010	Internal PN sequence's magnitude and exponent bits drive the channel.

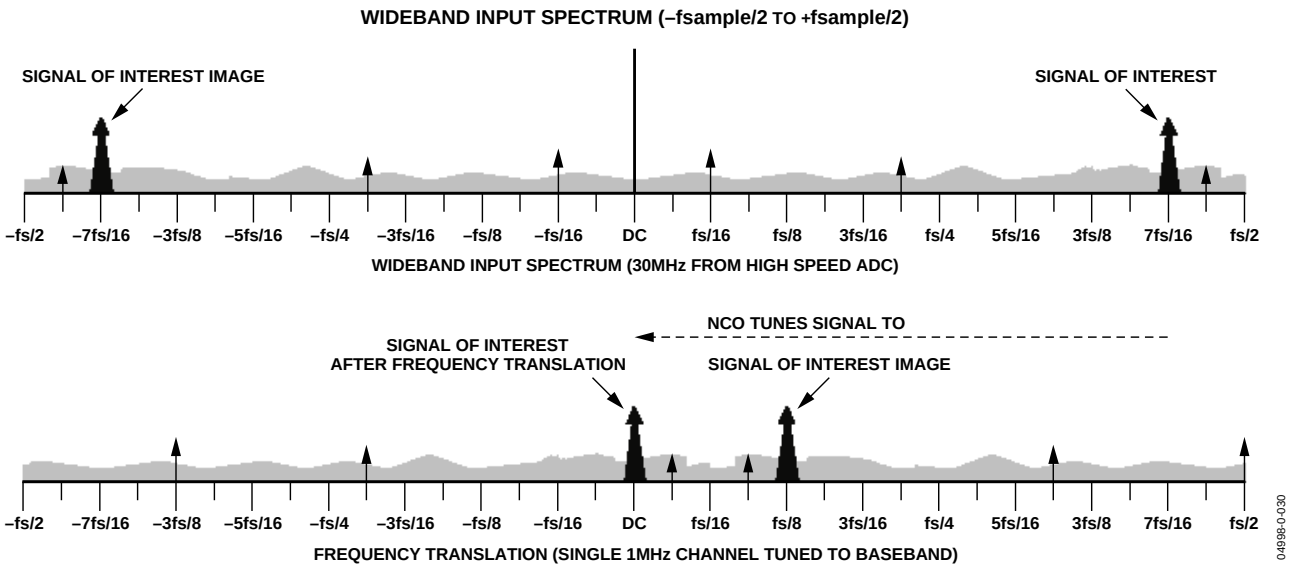


Figure 30. Frequency Translation Principle Using the NCO and Mixer

For example, if the carrier frequency is 100 MHz and the clock frequency is 80 MHz,

$$\frac{\text{mod}(f_{ch}, f_{clk})}{f_{clk}} = \frac{20}{80} = 0.25$$

This, in turn, converts to 0x4000 0000 in the 32-bit twos complement representation for *NCO_FREQ*.

If the carrier frequency is 70 MHz and the clock frequency is 80 MHz,

$$\frac{\text{mod}(f_{ch}, f_{clk})}{f_{clk}} = \frac{10}{80} = 0.125$$

This, in turn, converts to 0xE000 0000 in the twos complement 32-bit representation.

Mixer

The NCO is accompanied by a mixer. Its operation is similar to an analog mixer. It does the down-conversion of input signals (real or complex) by using the NCO frequency as a local oscillator. For real input signals, this mixer performs a real mixer operation (with two multipliers). For complex input signals, the mixer performs a complex mixer operation (with four multipliers). The mixer adjusts its operation based on the input signal (real or complex) provided to each individual channel.

Bypass

The NCO and the mixer can be bypassed individually in each channel by writing Logic 1 in the NCO bypass bit in the NCO control register of the channel under consideration. When bypassed, down-conversion is not performed and the AD6636 channel functions simply as a real filter on complex data. This is useful for baseband sampling applications in which the input Port A (or C) is connected to the I signal path within the filter and the Input Port B (or D) is connected to the Q signal path. This can be desired if the digitized signal has already been converted to baseband in prior analog stages or by other digital preprocessing.

Clear Phase Accumulator on Hop

When the clear NCO accumulator bit of the NCO control register is set (Logic 1), the NCO phase accumulator is cleared prior to a frequency hop. See the Chip Synchronization section for details on frequency hopping. This ensures a consistent phase of the NCO on each hop. The NCO phase offset is unaffected by this setting and is still in effect. If phase-continuous hopping is needed, this bit should be cleared (NCO accumulator is not cleared). The last phase in the NCO phase register is the initiating point for the new frequency.

Phase Dither

The AD6636 provides a phase dither option for improving the spurious performance of the NCO. Writing Logic 1 in the phase dither enable bit of the NCO control register of the individual channels enables phase dither. When phase dither is enabled, random phase is added to the LSBs of the phase accumulator of the NCO. When phase dither is enabled, spurs due to phase truncation in the NCO are randomized.

The energy from these spurs is spread into the noise floor and the spurious-free dynamic range is increased at the expense of a very slight decrease in the SNR. The choice of whether to use phase dither in a system is ultimately decided by the system goals. If lower spurs are desired at the expense of a slightly

raised noise floor, phase dither should be employed. If a low noise floor is desired and the higher spurs can be tolerated or filtered by subsequent stages, then phase dither is not needed.

Amplitude Dither

This can be used to improve spurious performance of the NCO. Amplitude dither is enabled by writing Logic 1 in the amplitude dither enable bit of the NCO control register of the channel under consideration. When this feature is enabled, random amplitude is added to the LSBs of the sine and cosine amplitudes. Amplitude dither improves performance by randomizing the amplitude quantization errors within the angular-to-Cartesian conversion of the NCO. This option can reduce spurs at the expense of a slightly raised noise floor. Amplitude and phase dither can be used together, separately, or not at all.

NCO Frequency Hold-Off Register

When the NCO frequency registers are written by the microport or serial port, data is passed to a shadow register. Data can be moved to the main registers when the channel comes out of sleep mode, or when a sync hop occurs. In either event, a counter can be loaded with the NCO frequency hold-off register value. The 16-bit unsigned integer counter starts counting down, clocked by the input port clock selected at the crossbar mux. When the counter reaches 0, the new frequency value in the shadow register is written to the NCO frequency register. Writing 1 in this hold-off register updates the NCO frequency register as soon as the start sync or hop sync occurs. See the Chip Synchronization section for details.

Phase Offset

The phase offset register can be written with a value that is added as an offset to the phase accumulator of the NCO. This 16-bit register is interpreted as a 16-bit unsigned integer. A 0x0000 in this register corresponds to a 0 radian offset and a 0xFFFF corresponds to an offset of $2\pi \times (1 - 1/2^{16})$ radians. This register allows multiple NCOs (multiple channels) to be synchronized to produce complex sinusoids with a known and steady phase difference.

Hop Sync

When the channel's NCO frequency needs to be changed from one frequency to a different frequency, a hop sync should be issued to the channel. This feature is discussed in detail in the Chip Synchronization section.

FIFTH-ORDER CIC FILTER

The signal processing stage immediately after the NCO is a CIC filter stage. This stage implements a fixed-coefficient, decimating, cascade integrated comb filter. The input rate to this filter is the same as the data rate at the input port; the output rate from this stage is dependent on the decimation factor.

$$f_{CIC} = \frac{f_{in}}{M_{CIC}}$$

The decimation ratio, M_{CIC}, can be programmed from 2 to 32 (only integer values). The 5-bit word in the CIC decimation register is used to set the CIC decimation factor. A binary value of one less than the decimation factor is written into this register. The decimation ratio of 1 can be achieved by bypassing the CIC filter stage. The frequency response of the filter is given by the following equations. The gain and pass-band droop of the CIC should be calculated by these equations. Both parameters can be offset in the RCF stage.

$$H(z) = \frac{1}{2^{(S_{CIC}+5)}} \times \left(\frac{1 - Z^{-M_{CIC}}}{1 - Z^{-1}} \right)^5$$

$$H(f) = \frac{1}{2^{(S_{CIC}+5)}} \times \left(\frac{SIN\left(\frac{M_{CIC} \times f}{f_{in}} \right)}{SIN\left(\pi \frac{f}{f_{in}} \right)} \right)^5$$

where:

f_{in} is the data input rate to the channel under consideration.

S_{CIC}, the scale factor, is a programmable unsigned integer between 0 and 20.

The attenuation of the data into the CIC stage should be controlled in 6 dB increments. For the best dynamic range, S_{CIC} should be set to the smallest value possible (lowest attenuation possible) without creating an overflow condition. This can be accomplished safely using the following equation, where input_level is the largest possible fraction of the full-scale value at the input port. This value is output from the NCO stage and pipelined into the CIC filter.

$$S_{CIC} = ceil\left(\log_2\left(M_{CIC}{}^5 \times input_level\right)\right) - 5$$

$$OL_{CIC} = \frac{\left(M_{CIC}{}^5\right)}{2^{S_{CIC}+5}} \times input_level$$

Bypass

The fifth-order CIC filter can be bypassed when no decimation is required of it. When it is bypassed, the scaling operation is not performed. In bypass mode, the output of the CIC filter is the same as the input of the CIC filter.

CIC Rejection

Table 14 illustrates the amount of bandwidth as a percentage of the data rate into the CIC stage, which can be protected with various decimation rates and alias rejection specifications. The maximum input rate into the CIC is 150 MHz (the same as the maximum input port data rate). The data may be scaled to any other allowable sample rate.

Table 14 can be used to decide the minimum decimation required in the CIC stage to preserve a certain bandwidth. The CIC5 stage can protect a much wider bandwidth to any given rejection, when a decimation ratio lower than that identified in the table is used. The table helps to calculate an upper boundary on decimation, M_{CIC}, given the desired filter characteristics.

Table 14. SSB CIC5 Alias Rejection Table (f_{in} = 1)

M_{CIC}^5	−60 dB	−70 dB	−80 dB	−90 dB	−100 dB
2	8.078	6.393	5.066	4.008	3.183
3	6.367	5.11	4.107	3.297	2.642
4	5.022	4.057	3.271	2.636	2.121
5	4.107	3.326	2.687	2.17	1.748
6	3.463	2.808	2.27	1.836	1.48
7	2.989	2.425	1.962	1.588	1.281
8	2.627	2.133	1.726	1.397	1.128
9	2.342	1.902	1.54	1.247	1.007
10	2.113	1.716	1.39	1.125	0.909
11	1.924	1.563	1.266	1.025	0.828
12	1.765	1.435	1.162	0.941	0.76
13	1.631	1.326	1.074	0.87	0.703
14	1.516	1.232	0.998	0.809	0.653
15	1.416	1.151	0.932	0.755	0.61
16	1.328	1.079	0.874	0.708	0.572
17	1.25	1.016	0.823	0.667	0.539
18	1.181	0.96	0.778	0.63	0.509
19	1.119	0.91	0.737	0.597	0.483
20	1.064	0.865	0.701	0.568	0.459
21	1.013	0.824	0.667	0.541	0.437
22	0.967	0.786	0.637	0.516	0.417
23	0.925	0.752	0.61	0.494	0.399
24	0.887	0.721	0.584	0.474	0.383
25	0.852	0.692	0.561	0.455	0.367
26	0.819	0.666	0.54	0.437	0.353
27	0.789	0.641	0.52	0.421	0.34
28	0.761	0.618	0.501	0.406	0.328
29	0.734	0.597	0.484	0.392	0.317
30	0.71	0.577	0.468	0.379	0.306
31	0.687	0.559	0.453	0.367	0.297
32	0.666	0.541	0.439	0.355	0.287

Example Calculations

Goal: Implement a filter with an input sample rate of 100 MHz requiring 100 dB of alias rejection for a ± 1.4 MHz pass band.

Solution: First determine the percentage of the sample rate that is represented by the pass band.

$$BW_{fraction} = 100 \times \frac{1.4\ \text{MHz}}{100\ \text{MHz}} = 1.4$$

In the −100 dB column in Table 14, find the value greater than or equal to the pass-band percentage of the clock rate. Then find the corresponding rate decimation factor (M_{CIC}). For an M_{CIC} of 6, the frequency that has −100 dB of alias rejection is 1.48%, which is slightly larger than the 1.4% calculated. Therefore, for this example, the maximum bound on the CIC decimation rate is 6. A higher M_{CIC} means less alias rejection than the 100 dB required.

FIR HALF-BAND BLOCK

The output of the CIC filter is pipelined into the FIR HB (half-band) block. Each channel has two sets of cascading fixed-coefficient FIR and fixed-coefficient half-band filters. The half-band filters decimate by 2. Each of these filters (FIR1, HB1, FIR2, HB2) are described in the following sections.

3-Tap Fixed-Coefficient Filter (FIR1)

The 3-tap FIR filter is useful in certain filter configurations in which extra alias protection is needed for the decimating HB1 filter. It is a simple sum-of-products FIR filter with three filter taps and 2-bit fixed coefficients. Note that this filter does not decimate. The coefficients of this symmetric filter are {1, 2, 1}. The normalized coefficients used in the implementation are {0.25, 0.5, 0.25}.

The user can either use or bypass this filter. Writing Logic 0 to the FIR1 enable bit in the FIR-HB control register bypasses this fixed-coefficient filter. The filter is useful in certain filter configurations only and bypassing it for other applications results in power savings.

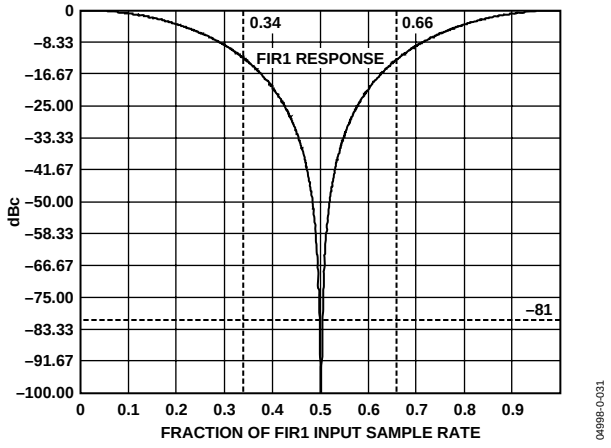


Figure 31. FIR1 Filter Response to the Input Rate of the Filter

This filter runs at the same sample rate as the CIC filter output rate and is given by

$$f_{FIR1} = f_{in}/M_{CIC}$$

where:

f_{in} is the input rate in to the channel.

M_{CIC} is the decimation ratio in the CIC filter stage.

The maximum input and output rates for this filter are 150 MHz.

Decimate-by-2, Half-Band Filter (HB1)

The next stage of the FIR-HB block is a decimate-by-2, half-band filter. The 11-tap, symmetrical, fixed-coefficient HB1 filter has low power consumption due to its polyphase implementation. The filter has 22 bits of input and output data with 10-bit coefficients. Table 15 lists the coefficients of the half-band filter. The normalized coefficients used in the implementation and the 10-bit decimal equivalent value of the coefficients are also listed. Other coefficients are 0s.

Table 15. Fixed Coefficients for HB1 Filter

Coefficient Number	Normalized Coefficient	Decimal Coefficient (10-Bit)
C1, C11	+0.013671875	+7
C3, C9	−0.103515625	−53
C5, C7	+0.58984375	+302
C6	+1	+512

Similar to the FIR1 filter, this filter can be used or bypassed. Writing Logic 0 to the HB1 enable bit in the FIR-HB control register bypasses this fixed-coefficient HB filter. The filter is useful in certain filter configurations only and bypassing it for other applications results in power savings. For example, it is useful in narrow-band and wideband output applications in which more filtering is required as compared to very wide bandwidth applications in which a higher output rate may prohibit the use of a decimating filter. The response of the filter is shown in Figure 32.

The input sample rate of this filter is the same as the CIC filter output rate and is given by

$$f_{HB1} = f_{in}/M_{CIC}$$

where:

f_{in} is the input rate in to the channel.

M_{CIC} is the decimation ratio in the CIC filter stage.

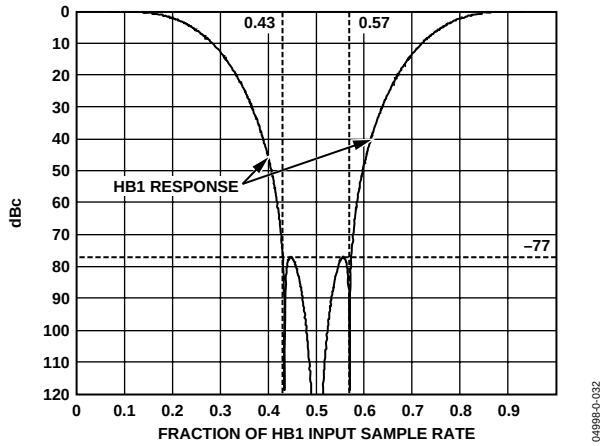


Figure 32. HB1 Filter Response to the Input Rate of the Filter

The filter has a maximum input sample rate of 150 MHz and, when the filter is not bypassed, the maximum output rate is 75 MHz.

The filter has a ripple of 0.0012 dB and rejection of 77 dB. For an alias rejection of 77 dB, the alias-protected bandwidth is 14% of the filter input sample rate. The bandwidth of the filter for a ripple of 0.00075 dB is also the same as the alias-protected bandwidth, due to the nature of half-band filters. The 3 dB bandwidth of this filter is 44% of the filter input sample rate. For example, if the sample rate into the filter is 50 MHz, then the alias-protected bandwidth of the HB1 filter is 7 MHz. If the bandwidth of the required carrier is greater than 7 MHz, then HB1 may not be useful.

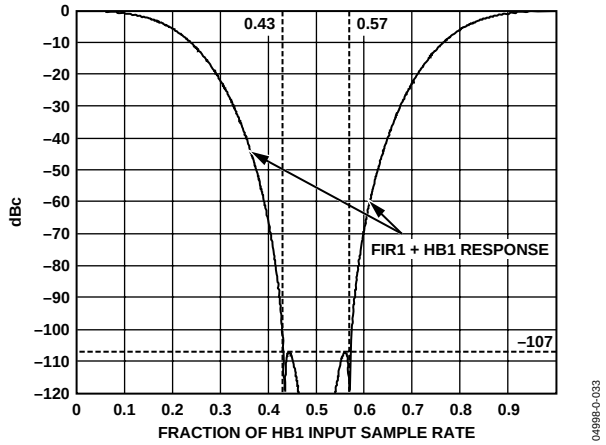


Figure 33. Composite Response of FIR1 and HB1 Filters to their Input Rate

6-Tap Fixed Coefficient Filter (FIR2)

Following the first cascade of the FIR1 and HB1 filters is the second cascade of the FIR2 and HB2 filters. The 6-tap, fixed-coefficient FIR2 filter is useful in providing extra alias protection for the decimating HB2 filter in certain filter configurations. It is a simple sum-of-products FIR filter with six filter taps and 5-bit fixed coefficients. Note that this filter does not decimate. The normalized coefficients used in the implementation and the 5-bit decimal equivalent value of the coefficients are listed in Table 16.

Table 16. 6-Tap FIR2 Filter Coefficients

Coefficient Number	Normalized Coefficient	Decimal Coefficient (5-Bit)
C0, C5	−0.125	−2
C1, C4	+0.1875	+3
C2, C3	+0.9375	+15

The user can either use or bypass this filter. Writing Logic 0 to the FIR2 enable bit in the FIR-HB control register bypasses this fixed-coefficient filter. The filter is useful in certain filter configurations only and bypassing it for other applications results in power savings. The filter is especially useful in increasing the stop-band attenuation of the HB2 filter that follows. Therefore, it is optimal to use both FIR2 and HB2 in a configuration.

This filter runs at a sample rate given by one of the following equations:

If HB1 is bypassed,

$$f_{FIR2} = f_{HB1}$$

If HB1 is not bypassed,

$$f_{FIR2} = f_{HB1}/2,$$

where:

f_{HB1} is the input rate of the HB1 filter.

f_{FIR2} is the input rate of the FIR2 filter.

The maximum input and output rate for this filter is 75 MHz. The response of the FIR2 filter is shown in Figure 34.

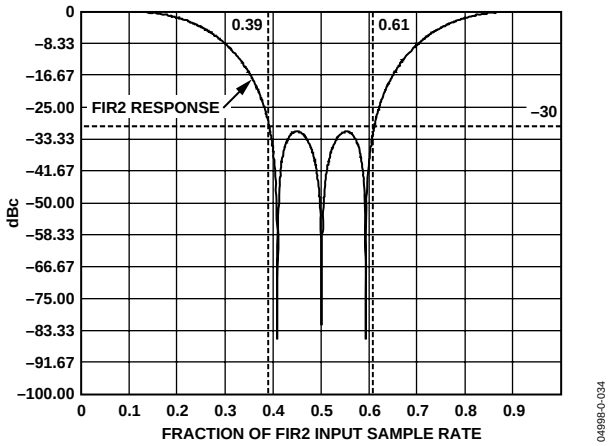


Figure 34. FIR2 Filter Response to the Input Rate of the Filter

Decimate-by-2, Half-Band Filter (HB2)

The second stage of the second cascade of the FIR-HB block is a decimate-by-2, half-band filter. The 27-tap, symmetric, fixed-coefficient HB2 filter has low power consumption due to its polyphase implementation. The filter has 20 bits of input and output data with 12-bit coefficients. The normalized coefficients used in the implementation and the 10-bit decimal equivalent value of the coefficients are listed in Table 17. Other coefficients are 0s.

Table 17. HB2 Filter Fixed Coefficients

Coefficient Number	Normalized Coefficient	Decimal Coefficient (12-Bit)
C1, C27	+0.00097656	+2
C3, C25	−0.00537109	−11
C5, C23	+0.015	+32
C7, C21	−0.0380859	−78
C9, C19	+0.0825195	+169
C11, C17	−0.1821289	−373
C13, C15	+0.6259766	+1282
C14	+1	+2048

Similar to the HB1 filter, the user can either use or bypass this filter. Writing Logic 0 to the HB1 enable bit in the FIR-HB control register bypasses this fixed-coefficient HB filter. The filter is useful in certain filter configurations only and bypassing it for other applications results in power savings. For example, the filter is useful in narrow-band applications in which more filtering is required, as compared to wide-band applications, in which a higher output rate may prohibit the use of a decimating filter. The response of the HB2 filter is shown in Figure 35.

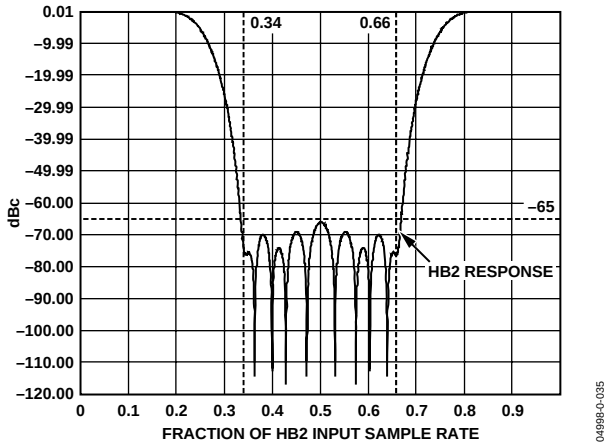


Figure 35. HB2 Filter Response to the Input Rate of the Filter

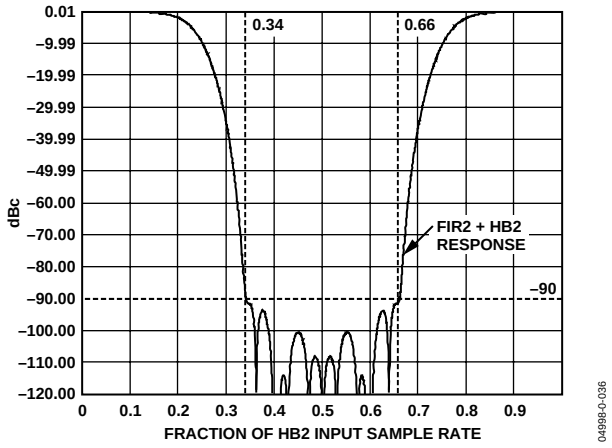


Figure 36. Composite Response of FIR2 and HB2 filters to their Input Rates

The filter input sample rate is the same as the FIR2 filter output rate and is given by one of the following equations:

If HB1 is bypassed,

$$f_{HB2} = f_{FIR2} = f_{HB1}$$

If HB1 is not bypassed,

$$f_{HB2} = f_{FIR2} = \frac{f_{HB1}}{2}$$

where:

f_{HB1} is the input rate of the HB1 filter.

f_{FIR2} is the input rate of the FIR2 filter.

f_{HB2} is the input rate of the HB2 filter.

The input to the filter has a maximum of 75 MHz. When not bypassed, the maximum output rate is 37.5 MHz.

The filter has a ripple of 0.00075 dB and rejection of 81 dB. For an alias rejection of 81 dB, the alias-protected bandwidth is 33% of the filter input sample rate. The bandwidth of the filter for a ripple of 0.00075 dB is the same as alias-protected bandwidth, due to the nature of half-band filters. The 3 dB bandwidth of this filter is 47% of the filter input sample rate. For example, if the sample rate into the filter is 25 MHz, then the alias-protected bandwidth of the HB2 filter is 8.25 MHz (33% of 25 MHz). If the bandwidth of the required carrier is greater than 8.25 MHz, then HB2 may not be useful.

INTERMEDIATE DATA ROUTER

Following the FIR-HB cascade filters is the intermediate data router. This data router consists of muxes that allow the I and Q data from any channel front end (input port + NCO + CIC + FIR-HB) to be processed by any channel back end (MRCF + DRCF + CRCF). The choice of channel front end is made by programming a 3-bit MRCF data select word in the MRCF control register. The valid values for this word and their corresponding settings are listed in Table 18.

Table 18. Data Router Select Settings

MRCF Data Select [2:0]	Data Source
000	Channel 0
001	Channel 1
010	Channel 2
011	Channel 3
1x0	Channel 4
1x1	Channel 5

Allowing different channel back ends to select different channel front ends is useful in the polyphase implementation of filters. When multiple AD6636 channels are used to process a single carrier, a single-channel front end feeds more than one channel back end. After processing through the channel back ends (RCF filters), the data is interleaved back from the polyphased channels.

MONORATE RAM COEFFICIENT FILTER (MRCF)

The MRCF is a programmable sum-of-products FIR filter. This filter block comes after the first data router and before the DRCF and CRCF programmable filters. It consists of a maximum of eight taps with 6-bit programmable coefficients. Note that this block does not decimate and is used as a helper filter for the DRCF and CRCF filters that follow in the signal chain.

The number of filter taps that are to be calculated is programmable using the 3-bit number-of-taps word in the MRCF control register of the channel under consideration. The 3-bit word programmed is one less than the number of filter taps. The coefficients themselves are programmed in eight MRCF

coefficient memory registers for individual channels. The input and output data to the block are both 20 bit.

Symmetry

Though the MRCF filter does not require symmetrical filters, if the filter is symmetrical, the symmetry bit in the MRCF control register should be set. When this bit is set, only half of the impulse response needs to be programmed into the MRCF coefficient memory registers. For example, if the number of filter taps is equal to five or six and the filter is symmetrical, only three coefficients need to be written into the coefficient memory. For both symmetrical and asymmetrical filters, the number of filter taps is limited to eight.

Clock Rate

The MRCF filter runs on an internal, high speed PLL clock. This clock rate can be as high as 200 MHz. If the half clock rate bit in the MRCF control register is set, only half the PLL clock rate is used (maximum of 100 MHz). This results in power savings but can only be used if certain conditions are met.

Because this filter is nondecimating, the input and output rates are both the same and equal to one of the following:

If HB2 is bypassed,

$$f_{MRCF} = f_{HB2}$$

If HB2 is not bypassed,

$$f_{MRCF} = \frac{f_{HB2}}{2}$$

If f_{PLLCLK} is the PLL clock and if $f_{MRCF} \times N_{TAPS} \leq \frac{f_{PLLCLK}}{2}$, then half of the PLL clock can be used for processing (power savings). Otherwise, the PLL clock should be used.

Bypass

The MRCF filter can be used in normal operation or bypassed using the MRCF bypass bit in the MRCF control register. When the filter is bypassed, the output of the filter is the same as the input of the filter. Bypassing the MRCF filter when it is not required results in power savings.

Scaling

The output of the MRCF filter can be scaled by using the 2-bit MRCF scaling word in the MRCF control register. Table 19 shows the valid values for the 2-bit word and their corresponding settings.

Table 19. MRCF Scaling Factor Settings

MRCF Scale Word [1:0]	Scaling Factor
00	18.06 dB attenuation
01	12.04 dB attenuation
10	6.02 dB attenuation
11	No scaling, 0 dB

DECIMATING RAM COEFFICIENT FILTER (DRCF)

Following the MRCF is the programmable DRCF FIR filter. This filter can calculate up to 64 asymmetrical filter taps or up to 128 symmetrical filter taps. The filter is also capable of a programmable decimation rate of from 1 to 16. A flexible coefficient offset feature allows loading multiple filters into the coefficient RAM and changing the filters on the fly. The decimation phase feature allows a polyphase implementation, where multiple AD6636 channels are used for processing a single carrier.

The DRCF filter has 20-bit input and output data and 14-bit coefficient data. The number of filter taps to calculate is programmable and is set in the DRCF taps register. The value of the number of taps minus one is written to this register. For example, a value of 19 in the register corresponds to 20 filter taps.

The decimation rate is programmable using the 4-bit DRCF decimation rate word in the DRCF control register. Again, the value written is the decimation rate minus one.

Bypass

The DRCF filter can be used in normal operation or bypassed using the DRCF bypass bit in the DRCF control register. When the DRCF filter is bypassed, no scaling is applied and the output of the filter is the same as the input to the DRCF filter.

Scaling

The output of the DRCF filter can be scaled using the 2-bit DRCF scaling word in the DRCF control register. Table 20 lists the valid values for the 2-bit word and their corresponding settings.

Table 20. DRCF Scaling Factor Settings

DRCF Scale Word [1:0]	Scaling Factor
00	18.06 dB attenuation
01	12.04 dB attenuation
10	6.02 dB attenuation
11	No scaling, 0 dB

Symmetry

The DRCF filter does not require symmetrical filters. However, if the filter is symmetrical, the symmetry bit in the DRCF control register should be set. When this bit is set, only half of the impulse response needs to be programmed into the DRCF coefficient memory registers. For example, if the number of filter taps is equal to 15 or 16 and the filter is symmetrical, only eight coefficients need to be written into the coefficient memory. Because a total of 64 taps can be written into the memory registers, the DRCF can perform 64 asymmetrical filter taps or 128 symmetrical filter taps.

Coefficient Offset

More than one set of filter coefficients can be loaded into the coefficient RAM at any given time (given sufficient RAM space). The coefficient offset can be used in this case to access the two or more different filters. By changing the coefficient offset, the filter coefficients being accessed can be changed on the fly. This decimal offset value is programmed in the DRCF coefficient offset register. When this value is changed during the calculation of a particular output data sample, the sample calculation is completed using the old coefficients, and the new coefficient offset from the next data sample calculation is used.

Decimation Phase

When more than one channel of AD6636 is used to process one carrier, polyphase implementation of corresponding channels' DRCF or CRCF is possible using the decimation phase feature. This feature can only be used under certain conditions. The decimation phase is programmed using the 4-bit DRCF decimation phase word of the DRCF control register.

Maximum Number of Taps Calculated

The output rate of the DRCF filter is given by

$$f_{DRCF} = \frac{f_{MRCF}}{M_{DRCF}}$$

where:

f_{MRCF} is the data rate out of the MRCF filter and into the DRCF filter.

M_{DRCF} is the decimation rate in the DRCF filter.

The DRCF filter consists of two multipliers (one each for the I and Q paths). Each multiplier, working at the high speed clock rate (PLL clock), can do one multiply (or one tap) per high speed clock cycle. Therefore, the maximum number of filter taps that can be calculated (symmetrical or asymmetrical filter) is given by

$$Maximum\,Number\ of\ Taps = ceil\left(\frac{f_{PLLCLK}}{f_{DRCF}}\right) - 1$$

where:

f_{PLLCLK} is the high speed internal processing clock generated by the PLL clock multiplier.

f_{DRCF} is the output rate of the DRCF filter calculated above.

Programming DRCF Registers for an Asymmetrical Filter

To program the DRCF registers for an asymmetrical filter:

1. Write *NTAPS* – 1 in the DRCF taps register, where *NTAPS* is the number of filter taps. The absolute maximum value for NTAPS is 64 in asymmetrical filter mode.
2. Write 0 to the DRCF coefficient offset register.
3. Write 0 to the symmetrical filter bit in the DRCF control register.
4. Write the start address for the coefficient RAM, typically equal to the coefficient offset register, in the DRCF start address register.
5. In the DRCF stop address register, write the stop address for the coefficient RAM, typically equal to

   Coefficient Offset + *NTAPS* − 1
6. Write all coefficients in reverse order (start with last coefficient) to the DRCF coefficient memory register. If in 8-bit microport mode or serial port mode, write the lower byte of the memory register first and then the higher byte.
7. After each write access to the DRCF coefficient memory register, the internal RAM address is incremented starting with the start address and ending with the stop address.

Note that each write or read access increments the internal RAM address. Therefore, all coefficients should be read first before reading them back. Also, for debugging purposes, each RAM address can be written individually by making the start address and stop address the same. Therefore, to program one RAM location, the user writes the address of the RAM location to both the start and stop address registers, and then writes the coefficient memory register.

Programming DRCF Registers for a Symmetric Filter

To program the DRCF registers for a symmetrical filter:

1. Write *NTAPS* – 1 in the DRCF taps register, where *NTAPS* is the number of filter taps. The absolute maximum value for NTAPS is 128 in symmetric filter mode.
2. Write *ceil*(64 – *NTAPS*/2) to the DRCF coefficient offset register, where the *ceil* function takes the closest integer greater than or equal to the argument.
3. Write 1 to the symmetrical filter bit in the DRCF control register.
4. Write the start address for the coefficient RAM, typically equal to coefficient offset register, in the DRCF start address register.

5. Write the stop address for the coefficient RAM, typically equal to ceil(*NTAPS*/2) – 1, in the DRCF stop address register.

6. Write all coefficients to the DRCF coefficient memory register, starting with the middle of the filter and working towards the end of the filter. When coefficients are numbered 0 to *NTAPS* – 1, the middle coefficient is given by the coefficient number ceil(*NTAPS*/2). If in 8-bit microport mode or serial port mode, write the lower byte of the memory register first and then the higher byte. After each write access to the DRCF coefficient memory register, the internal RAM address is incremented starting with the start address and ending with stop address.

Note that each write or read access increments the internal RAM address. Therefore, all coefficients should be written first before reading them back. Also, for debugging purposes, each RAM address can be written individually by making the start and stop addresses the same. Therefore, to program one RAM location, the user writes the address of the RAM location to both the start and stop address registers, and then writes to the coefficient memory register.

CHANNEL RAM COEFFICIENT FILTER (CRCF)

Following the DRCF is the programmable decimating CRCF FIR filter. The only difference between the DRCF and CRCF filters is the coefficient bit width. The DRCF has 14-bit coefficients, while the CRCF has 20-bit coefficients.

This filter can calculate up to 64 asymmetrical filter taps or up to 128 symmetrical filter taps. The filter is capable of a programmable decimation rate from 1 to 16. The flexible coefficient offset feature allows loading multiple filters into the coefficient RAM and changing the filters on the fly. The decimation phase feature allows for a polyphase implementation in which multiple AD6636 channels are used to process a single carrier.

The CRCF filter has 20-bit input and output data and 20-bit coefficient data. The number of filter taps to calculate is programmable and is set in the CRCF taps register. The value of the number of taps minus one is written to this register. For example, a value of 19 in the register corresponds to 20 filter taps. The decimation rate is programmable using the 4-bit CRCF decimation rate word in the CRCF control register. Again, the value written is the decimation rate minus one.

Bypass

The CRCF filter can be used in normal operation or bypassed using the CRCF bypass bit in the CRCF control register. When the CRCF filter is bypassed, no scaling is applied and the output of the filter is the same as the input to the CRCF filter.

Scaling

The output of the CRCF filter can be scaled using the 2-bit CRCF scaling word in the CRCF control register. Table 21 shows the valid values for the 2-bit word and the corresponding settings. | ∑COEFF | is the sum of all coefficients (in normalized form) used to calculate the FIR filter.

Table 21. CRCF Scaling Factor Settings

CRCF Scale Word [1:0]	Scaling Factor
00	18.06 dB attenuation
01	12.04 dB attenuation
10	6.02 dB attenuation
11	No scaling, 0 dB

Symmetry

The CRCF filter does not require symmetrical filters. However, if the filter is symmetrical, the symmetry bit in the CRCF control register should be set. When this bit is set, only half the impulse response needs to be programmed into the CRCF coefficient memory registers. For example, if the number of filter taps is equal to 15 or 16 and the filter is symmetric, then only eight coefficients need to be written into the coefficient memory. Because a total of 64 taps can be written into the memory registers, the CRCF can perform 64 asymmetrical filter taps or 128 symmetrical filter taps.

Coefficient Offset

More than one set of filter coefficients can be loaded into the coefficient RAM at any time (given sufficient RAM space). The coefficient offset can be used in this case to access the two or more different filters. By changing the coefficient offset, the filter coefficients being accessed can be changed on the fly. This decimal offset value is programmed in the CRCF coefficient offset register. When this value is changed during the calculation of a particular output data sample, the sample calculation is completed using the old coefficients, and the new coefficient offset is brought into effect from the next data sample calculation.

Decimation Phase

When more than one channel of the AD6636 is used to process one carrier, polyphase implementation of the corresponding channels' DRCF or CRCF is possible using the decimation phase feature. This feature can only be used under certain conditions. The decimation phase is programmed using the 4-bit CRCF decimation phase word of the CRCF control register.

Maximum Number of Taps Calculated

The output rate of the CRCF filter is given by

$$f_{CRCF} = \frac{f_{DRCF}}{M_{CRCF}}$$

where:

f_{DRCF} is the data rate out of the DRCF filter and into the CRCF filter.

M_{CRCF} is the decimation rate in the CRCF filter.

The CRCF filter consists of two multipliers (one each for the I and Q paths). Each multiplier, working at the high speed clock rate (PLL clock), can multiply (or tap once). Therefore, the maximum number of filter taps that can be calculated (symmetrical or asymmetrical filter) is given by

$$Maximum\,Number\,of\,Taps = ceil\left(\frac{f_{PLLCLK}}{f_{CRCF}}\right) - 1$$

where:

f_{PLLCLK} is the high speed internal processing clock generated by the PLL clock multiplier.

f_{CRCF} is the output rate of the CRCF filter as calculated previously.

Programming CRCF Registers for an Asymmetrical Filter

To program the CRCF registers for an asymmetrical filter:

1. Write *NTAPS* – 1 in the CRCF taps register, where *NTAPS* is the number of filter taps. The absolute maximum value for NTAPS is 64 in asymmetrical filter mode.
2. Write 0 in the CRCF coefficient offset register.
3. Write 0 in the symmetrical filter bit in the CRCF control register.
4. In the CRCF start address register, write the start address for the coefficient RAM, typically equal to the coefficient offset register.
5. In the CRCF stop address register, write the stop address for the coefficient RAM, typically equal to

   Coefficient Offset + *NTAPS* – 1
6. Write all coefficients in reverse order (start with last coefficient) to the CRCF coefficient memory register. In 8-bit microport mode or serial port mode, write the lower byte of the memory register first and then the higher byte. In 16-bit microport mode, write the lower 16-bits of the CRCF memory register first and then the high four bits. After each write access to the CRCF coefficient memory register, the internal RAM address is incremented starting with the start address and ending with the stop address.

Note that each write or read access increments the internal RAM address. Therefore, all coefficients should be read first before reading them back. Also, for debugging purposes, each RAM address can be written individually by making the start and stop addresses the same. Therefore, to program one RAM location, the user writes the address of the RAM location to both the start and stop address registers, and then writes the coefficient memory register.

Programming CRCF Registers for a Symmetrical Filter

To program the CRCF registers for a symmetrical filter:

1. Write *NTAPS* – 1 in the CRCF taps register, where *NTAPS* is the number of filter taps. The absolute maximum value for NTAPS is 128 in symmetrical filter mode.
2. Write *ceil*(64 – *NTAPS*/2) to the CRCF coefficient offset register, where the *ceil* function takes the closest integer greater than or equal to the argument.
3. Write 1 to the symmetrical filter bit in the CRCF control register.
4. In the CRCF start address register, write the start address for the coefficient RAM, typically equal to the coefficient offset register.
5. In the CRCF stop address register, write the stop address for the coefficient RAM, typically equal to

   ceil(*NTAPS*/2) – 1
6. Write all coefficients to the CRCF coefficient memory register, starting with middle of the filter and working towards the end of the filter. When coefficients are numbered 0 to *NTAPS* – 1, the middle coefficient is given by the coefficient number *ceil*(*NTAPS*/2). In 8-bit microport mode or serial port mode, write the lower byte of the memory register first and then the higher byte. In 16-bit microport mode, write the lower 16-bits of the CRCF memory register first and then the high four bits. After each write access to the CRCF coefficient memory register, the internal RAM address is incremented starting with the start address and ending with the stop address.

Note that each write or read access increments the internal RAM address. Therefore, all coefficients should be written first before reading them back. Also, for debugging purposes, each RAM address can be written individually by making the start and stop addresses the same. Therefore, to program one RAM location, the user writes the address of the RAM location to both the start and stop address registers, and then writes the coefficient memory register.

INTERPOLATING HALF-BAND FILTER

The AD6636 has interpolating half-band FIR filters that immediately follow the CRCF programmable FIR filters and precede the second data router. Each interpolating half-band filter takes 22-bit I and 22-bit Q data from the preceding CRCF and outputs rounded 22-bit I and 22-bit Q data to the second data router. A 10-tap fixed-coefficient filter is implemented in this stage.

The maximum input rate into this block is 17 MHz. Consequently, the maximum output is constrained to 34 MHz. The normalized coefficients used in the implementation and the 10-bit decimal equivalent value of the coefficients are listed in Table 22. Other coefficients are 0.

Table 22. Interpolating HB Filter Fixed Coefficients

Coefficient Number	Normalized Coefficient	Decimal Coefficient (10-Bit)
C1, C11	+0.02734375	+14
C3, C9	−0.12890625	−66
C5, C7	+0.603515625	+309
C6	+1	+512

The half-band filters interpolate the incoming data by 2×. For a channel running at 2× the chip rate, the half-band can be used to output channel data at 4× the chip rate. The interpolation operation creates an image of the baseband signal, which is filtered out by the half-band filter.

The image rejection of this filter is about 55 dB, but is still sufficient, because the image is from the desired signal, not an interfering signal. Note that the interpolating half-band filter can be enabled by writing a Logic 1 to Bit 9 of the MRCF control registers.

The frequency response of the interpolating half-band FIR is shown in Figure 37 with respect to the chip rate. The input rate to this filter is 2× the chip rate, and the output rate is 4× the chip rate.

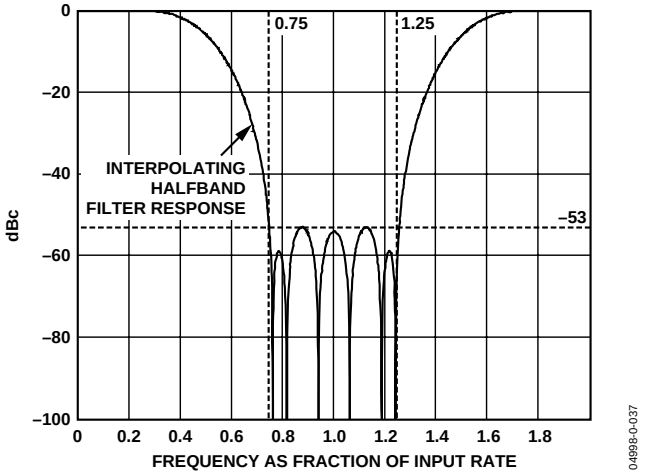


Figure 37. Interpolating Half-Band Frequency Response

OUTPUT DATA ROUTER

The output data router circuit precedes the six AGCs of the final output block and immediately follows the interpolating half-band filters. This block consists of two subblocks. The first subblock is responsible for combining (interleaving) data from more than one channel into a single stream of data.

The second subblock can perform two special functions, either complex filter completion or biphase filtering. The combined data is passed on to the AGCs.

Interleaving Data

In some cases, filtering using a single channel is insufficient. For such setups, it is advantageous to combine the filtering resources of more than one channel.

Multiple channels can be set up to work on the ADC input port data with the same NCO and filter setups. The decimation phase values in one of the RCF filters are set such that the channel filters are exactly out of phase with each other. In the data router, these multiple channels are interleaved (combined) to form a single stream of data. Because each individual channel is decimated more than it would be if a single channel were filtering, a larger number of filter taps can be calculated.

For example, two channels need to work together to produce a filter at an output rate of 10 MHz when the input rate is 100 MHz. Each channel is decimated by a factor of 20 (total decimation) to achieve the desired output rate of 5 MHz each. This compares to a decimation of 10, if a single channel were filtering.

The same coefficients are programmed in both channels' RCF filters, and the decimation phases are set to 0 and 1. The decimation phases can be set to 0 for one channel and 1 for the second channel in the pair. This causes the first channel to produce the even outputs of the filter, and the second to produce the odd outputs of the filter. The streams can then be recombined (interleaved) to produce the desired 10 MHz output rate. The benefit is that now each channel's RCF has time to calculate twice as many taps because it has a lower output rate.

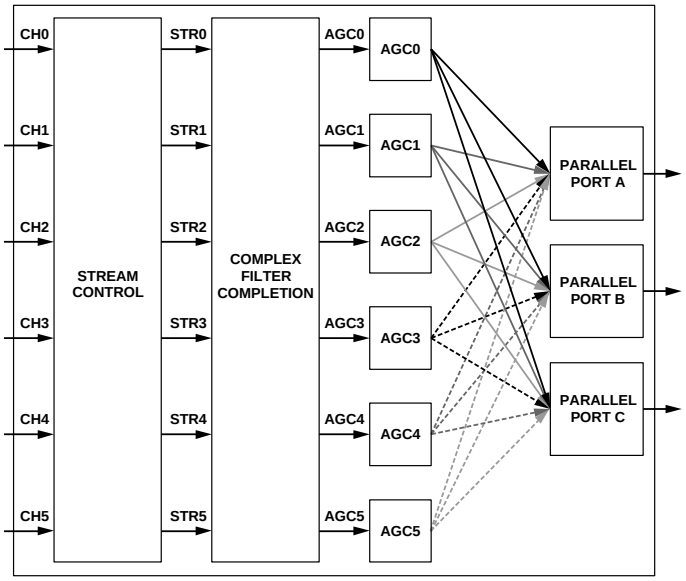


Figure 38. Output Data Router Block Diagram

The interleaving function is a simple time-multiplexing function, with a lower data rate on the input side and a higher data rate on the output side. The output data rate is the sum of all input stream data rates that are combined.

The channels that need to be combined are programmable with sufficient flexibility. Table 23 gives the combinations that are possible using a 4-bit word (stream control bits) in the Parallel Port Control 2 register.

After interleaving of data (see the Output Data Router section), the data is passed to the second subblock, in which either complex filter completion or biphase filtering can be performed.

Complex Filter Completion

In normal operation, each individual channel's filter performs real coefficient, complex data filtering.

Two channels are used to perform complex coefficient data filtering. One channel is loaded with the real part (in-phase) of the coefficients; the other channel is loaded with the imaginary part (quadrature) of the coefficients.

The calculated terms include:

- (ICi, QCi) from first channel
- (Icq, QCq) from the second channel

Using these terms, the complex filter is completed by applying

$$(I + jQ)\ (Ci + jCq) = (ICi - QCq) + j(ICq + QCi)$$

The channels to be combined can be programmed using a 3-bit complex control word in the Parallel Output Control 2 register. The values for the 3-bit control word and the corresponding settings are listed in Table 24.

These outputs go to the six available AGCs. Not all AGCs need to be used in the different applications, so unused AGCs can be bypassed and the output data streams ignored by the parallel output ports. For example, if Stream 0 and Stream 1 are combined for a complex filter, AGC1 can be bypassed, because Stream 1 is already combined into Stream 0 and sent to AGC0.

Table 23. Stream Control Bit Combinations

Stream Control Bits	Output Streams	No. of Streams
0000	Ch 0/Ch 1 combined, Ch 2, Ch 3, Ch 4, Ch 5 independent	5
0001	Ch 0/Ch 1/Ch 2 combined, Ch 3, Ch 4, Ch5 independent	4
0010	Ch 0/Ch 1/Ch 2/Ch 3 combined; Ch 4, Ch 5 independent	3
0011	Ch 0/Ch 1/Ch 2/Ch 3/Ch 4 combined; Ch 5 independent	2
0100	Ch 0/Ch 1/Ch 2/Ch 3/Ch 4/Ch 5 combined	1
0101	Ch 0/Ch 1/Ch 2 combined, Ch 3/Ch 4/Ch 5 combined	2
0110	Ch 0/Ch 1 combined, Ch 2/Ch 3 combined, Ch 4/Ch 5 combined	3
0111	Ch 0/Ch 1 combined, Ch 2/Ch 3 combined, Ch 4, Ch 5 independent	3
1000	Ch 0/Ch 1/Ch 2 combined, Ch 3/Ch 4 combined, Ch 5 independent	3
1001	Ch 0/Ch 1/Ch 2/Ch 3 combined, Ch 4/Ch 5 combined.	2
Any other state	Independent channels	6

Table 24. Definitions for Complex Control Register Selections

Complex Control Word	Data Routing	Comments
000	No complex filters	Stream control register controls AGC usage.
001	Stream 0/Stream 1 combined	Allows Ch 0 and Ch 1 to form a complex filter.
010	Stream 0/Stream 1 combined, Stream 2/Stream 3 combined	Allows Ch 0 and Ch 1 to form a complex filter and Ch 2 and Ch 3 to form a complex filter.
011	Stream 0/Stream 1 combined, Stream 2/Stream 3 combined, Stream 4/Stream 5 combined	Allows Ch 0 and Ch 1 to form a complex filter, Ch 2 and Ch 3 to form a complex filter, and Ch 4 and Ch 5 to form a complex filter.
101	Stream 0/Stream 1 Combined	Allows Ch 0 and Ch 1 to form a biphase filter.
110	Stream 0/Stream 1 combined, Stream 2/Stream 3 combined	Allows Ch 0 and Ch 1 to form a biphase filter, and Ch 2 and Ch 3 to form a biphase filter.
111	Stream 0/Stream 1 combined, Stream 2/Stream 3 combined, Stream 4/Stream 5 combined	Allows Ch 0 and Ch 1 to form a biphase filter, Ch 2 and Ch 3 to form a biphase filter, and Ch 4 and Ch 5 to form a biphase filter.

Biphase Filtering Option

The second special function that can be performed by the second subblock of the output data router is called the biphase filtering option. With this option, the AD6636 can be used to process data from ADCs that run faster than the input clock frequency by using two channels or two streams to form a biphase filter.

For example, a 300 MHz ADC can be used with a clock rate of 150 MHz driving the ADC. The ADC data can be decimated by 2 to produce even and odd data streams of data. The even stream can be clocked into ADC Input Port A, and the odd stream can be clocked into ADC Input Port B. These input ports drive separate channels or separate groups of channels. The filters of the RCF can be designed to place a 300 MHz sample time difference (1/300 MHz = 3.3 ns) between the even and odd path filters.

After the channel-filter coefficients have appropriate delay, a complex addition of the odd and even sample channels can be performed to create a single filter. This equivalent filter looks like a single channel with a 300 MHz input rate, even though the clock rate of the chip runs at only 150 MHz.

A biphase filter summation is implemented by

$$Output = (Ie \times Ce + Io \times Co) + j(Qe \times Ce + Qo \times Co)$$

where:

$Ie \times Ce$, $Qe \times Ce$ are even in-phase and quadrature-phase samples from one stream.

$Io \times Co$ and $Qo \times Co$ are odd in-phase and quadrature-phase samples from the other stream.

Ce and Co are the even and odd coefficients, which differ by 1 high speed sample time (300 MHz in the previous example).

Users can program certain streams to be summed using the biphase filtering option. This option can be programmed using the same 3-bit complex control word in the Parallel Output Control 2 register. The values for the 3-bit control word and their corresponding settings are listed in Table 24.

AUTOMATIC GAIN CONTROL

The AD6636 is equipped with six independent automatic gain control (AGC) loops that directly follow the second data router and immediately precede the parallel output ports. Each AGC circuit has 96 dB of range. It is important that the decimating filters of the AD6636 preceding the AGC reject unwanted signals so that each AGC loop is operating on the carrier of interest only, and carriers at other frequencies do not affect the ranging of the loop.

The AGC compresses the 22-bit complex output from the second data router into a programmable word size of 4 bits to 8 bits, 10 bits, 12 bits, or 16 bits. Because the small signals from the lower bits are pushed in to higher bits by adding gain, the clipping of the lower bits does not compromise the SNR of the signal of interest.

The AGC maintains a constant mean power on the output despite the level of the signal of interest, allowing operation in environments where the dynamic range of the signal exceeds the dynamic range of the output resolution. The output width of the AGC is set by writing a 3-bit AGC word length word in the AGC control register of the individual channel's memory map.

The AGC can be bypassed, if needed, and, when bypassed, the 22-bit complex input word is still truncated to a 16-bit value that is output through the parallel port output. The six AGCs available on the AD6636 are programmable through the six channel memory maps. AGCs corresponding to individual channels can be bypassed by writing Logic 1 to AGC bypass bit in the AGC control register.

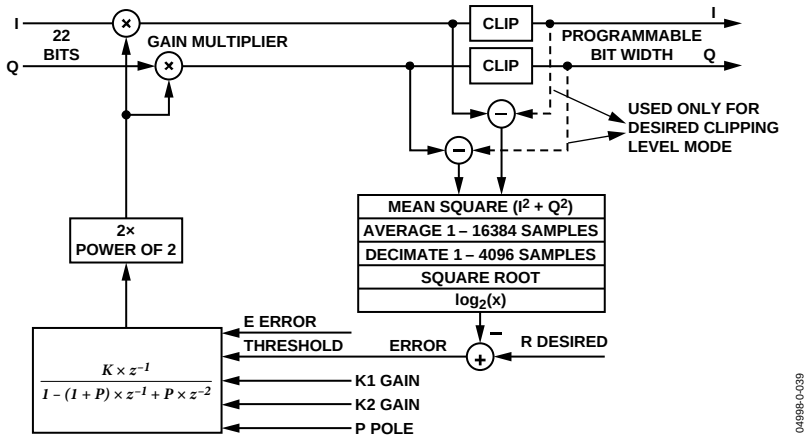


Figure 39. Block Diagram of the AGC

Three sources of error can be introduced by the AGC function: underflow, overflow, and modulation. Underflow is caused by truncation of bits below the output range. Overflow is caused by clipping errors when the output signal exceeds the output range. Modulation error occurs when the output gain varies while receiving data.

The desired signal level should be set based on the probability density function of the signal, so that the errors due to underflow and overflow are balanced. The gain and damping values of the loop filter should be set so that the AGC is fast enough to track long-term amplitude variations of the signal that may cause excessive underflow or overflow but slow enough to avoid excessive loss of amplitude information due to the modulation of the signal.

AGC Loop

The AGC loop is implemented using a log-linear architecture. It contains four basic operations: power calculation, error calculation, loop filtering, and gain multiplication.

The AGC can be configured to operate in either desired signal level mode or desired clipping level mode. The mode is set by the AGC clipping error bit of the AGC control register. The AGC adjusts the gain of the incoming data according to how far it is from a given desired signal level or desired clipping level, depending on the selected mode of operation.

Two datapaths to the AGC loop are provided: one before the clipping circuitry and one after the clipping circuitry, as shown in Figure 39. For the desired signal level mode, only the I/Q path from before the clipping is used. For the desired clipping level mode, the difference of the I/Q signals from before and after the clipping circuitry is used.

Desired Signal Level Mode

In this mode of operation, the AGC strives to maintain the output signal at a programmable set level. The desired signal level mode is selected by writing Logic 0 into the AGC mode bit of the AGC control register. The loop finds the square (or power) of the incoming complex data signal by squaring I and Q and adding them.

The AGC loop has an average and decimate block. This average and decimate operation takes place on power samples and before the square root operation. This block can be programmed to average from 1 to 16,384 power samples, and the decimate section can be programmed to update the AGC once every 1 to 4,096 samples. The limitation on the averaging operation is that the number of averaged power samples should be a multiple of the decimation value (1×, 2×, 3×, or 4×).

The averaging and decimation effectively means that the AGC can operate over averaged power of 1 to 16,384 output samples. Updating the AGC once every 1 to 4,096 samples and operating on average power facilitates the implementation of the loop filter with slow time constants, where the AGC error converges slowly and makes infrequent gain adjustments. It is also useful when the user wants to keep the gain scaling constant over a frame of data or a stream of symbols.

Due to the limitation that the number of average samples must be a multiple of the decimation value, only the multiple numbers 1, 2, 3, or 4 are programmed. This is set using the AGC average samples word in the AGC average sample register. These averaged samples are then decimated with decimation ratios programmable from 1 to 4,096. This decimation ratio is defined in the 12-bit AGC update decimation register.

The average and decimate operations are tied together and implemented using a first-order CIC filter and FIFO registers. Gain and bit growth are associated with CIC filters and depend on the decimation ratio. To compensate for the gain associated with these operations, attenuation scaling is provided before the CIC filter.

This scaling operation accounts for the division associated with the averaging operation as well as the traditional bit growth in CIC filters. Because this scaling is implemented as a bit-shift operation, only coarse scaling is possible. Fine scaling is implemented as an offset in the request level, as explained later in this section. The attenuation scaling S_{CIC} is programmable from 0 to 14 using a 4-bit CIC scale word in the AGC average samples register and is given by

$$S_{CIC} = ceil\ [\log_2(M_{CIC} \times N_{avg})]$$

where:

M_{CIC} is the decimation ratio (1 to 4,096).

N_{AVG} is the number of averaged samples programmed as a multiple of the decimation ratio (1, 2, 3, or 4).

For example, if a decimation ratio M_{CIC} is 1,000 and N_{avg} is 3 (decimation of 1,000 and averaging of 3,000 samples), then the actual gain due to averaging and decimation is 3,000 or 69.54 dB ($\log_2$ (3000)). Because attenuation is implemented as a bit-shift operation, only multiples of 6.02 dB attenuations are possible. S_{CIC} in this case is 12, corresponding to 72.24 dB. This way, S_{CIC} scaling always attenuates more than is sufficient to compensate for the gain in the average and decimate sections and, therefore, prevents overflows in the AGC loop. However, it is also evident that the S_{CIC} scaling induces a gain error (the difference between gain due to CIC and attenuation provided by scaling) of up to 6.02 dB. This error should be compensated for in the request signal level, as explained later in this section.

A Base 2 logarithm is applied to the output from the average and decimate section. These decimated power samples are converted to rms signal samples by applying a square root operation. This square root is implemented using a simple shift operation in the logarithmic domain. The rms samples obtained are subtracted from the request signal level R specified in the AGC desired level register, leaving an error term to be processed by the loop filter, G(z).

The user sets this programmable request signal level R according to the output signal level that is desired. The request signal level R is programmable from −0 dB to −23.99 dB in steps of 0.094 dB.

The request signal level should also compensate for errors, if any, due to the CIC scaling, as explained previously in this section. Therefore, the request signal level is offset by the amount of error induced in CIC, given by

$$Offset = 10 \times \log(M_{CIC} \times N_{avg}) - S_{CIC} \times 3.01\ \text{dB}$$

where *Offset* is in dB.

Continuing the previous example, this offset is given by

$$Offset = 72.24 - 69.54 = 2.7\ \text{dB}$$

So the request signal level is given by

$$R = -ceil\left[\frac{(DSL - Offset)}{0.094}\right] \times 0.094\ \text{dBFS}$$

where:

R is the request signal level.

DSL (desired signal level) is the output signal level that the user desires.

Therefore, in the previous example, if the desired signal level is −13.8 dB, the request level R is programmed to be −16.54 dB, compensating for the offset.

This request signal level is programmed in the 8-bit AGC desired level register. This register has a floating-point representation, where the 2 MSBs are exponent bits and the 6 LSBs are mantissa bits. The exponent is in steps of 6.02 dB, and the mantissa is in steps of 0.094 dB. For example, a 10'100101 value represents 2 × 6.02 + 37 × 0.094 = 15.518 dB.

The AGC provides a programmable second-order loop filter. The programmable parameters gain 1 (K_1), gain 2 (K_2), error threshold E, and pole P completely define the loop filter characteristics. The error term after subtracting the request signal level is processed by the loop filter, G(z). The open-loop poles of the second-order loop filter are 1 and P, respectively. The loop filter parameters, pole P and gain K, allow the adjustment of the filter time constant that determines the window for calculating the peak-to-average ratio.

Depending on the value of the error term that is obtained after subtracting the request signal level from the actual signal level, either gain value, K_1 or K_2, is used. If the error is less than the programmable threshold E, K_1, or K_2 is used. This allows a fast loop when the error term is high (large convergence steps required) and a slower loop function when error term is smaller (almost converged).

The open-loop gain used in the second-order loop G(z) is given by one of the following equations:

If Error < Error Threshold,

$K = K_1$

If Error > Error Threshold,

$K = K_2$

The open-loop transfer function for the filter, including the gain parameter, is

$$G(z) = \frac{Kz^{-1}}{1-(1+P)z^{-1}+Pz^{-2}}$$

If the AGC is properly configured in terms of offset in request level, then there are no gains in the AGC loop except for the filter gain K. Under these circumstances, a closed-loop expression for the AGC loop is given by

$$G_{closed}(z) = \frac{G(z)}{1+G(z)} = \frac{Kz^{-1}}{1+(K-1-P)z^{-1}+Pz^{-2}}$$

The gain parameters K_1, K_2, and pole P are programmable through AGC loop gain 1, 2, and AGC pole location registers from 0 to 0.996 in steps of 0.0039 using 8-bit representation. For example, 1000 1001 represent (137/256 = 0.535156). The error threshold value is programmable between 0 dB and 96.3 dB in steps of 0.024 dB. This value is programmed in the 12-bit AGC error threshold register, using floating-point representation. It consists of four exponent bits and eight mantissa bits. Exponent bits are in steps of 6.02 dB and mantissa bits are in steps of 0.024 dB. For example, 0111'10001001 represents 7 × 6.02 + 137 × 0.024 = 45.428 dB.

The user defines the open-loop pole P and gain K, which also directly impact the placement of the closed-loop poles and filter characteristics. These closed-loop poles, P_1, P_2, are the roots of the denominator of the previous closed-loop transfer function and are given by

$$P_1, P_2 = \frac{(1+P-K) \pm \sqrt{(1+P-K)^2 - 4P}}{2}$$

Typically, the AGC loop performance is defined in terms of its time constant or settling time. In this case, the closed-loop poles should be set to meet the time constants required by the AGC loop.

The relationship between the time constant and the closed-loop poles that can be used for this purpose is

$$P_{1,2} = \exp\left[\frac{M_{CIC}}{Sample\ Rate \times \tau_{1,2}}\right]$$

where $\tau_{1,2}$ are the time constants corresponding to poles $P_{1,2}$.

The time constants can also be derived from settling times as given by

$$\tau = \frac{2\%\ settling\ time}{4} \text{ or } \frac{5\%\ settling\ time}{3}$$

M_{CIC} (CIC decimation is from 1 to 4,096) and either the settling time or time constant are chosen by the user. The sample rate is the sample rate of the stream coming into the AGC. If channels were interleaved in the output data router, then the combined sample rate into the AGC should be considered. This rate should be used in the calculation of poles in the previous equation, where the sample rate is mentioned.

The loop filter output corresponds to the signal gain that is updated by the AGC. Because all computation in the loop filter is done in logarithmic domain (to the Base 2) of the samples, the signal gain is generated using the exponent (power of 2) of the loop filter output.

The gain multiplier gives the product of the signal gain with both the I and Q data entering the AGC section. This signal gain is applied as a coarse 4-bit scaling and then as a fine scale 8-bit multiplier. Therefore, the applied signal gain is from 0 dB to 96.3 dB in steps of 0.024 dB. The initial signal gain is programmable using the AGC signal gain register. This register is again a 4 exponent + 8 mantissa bit floating-point representation similar to the error threshold. This is taken as the initial gain value before the AGC loop starts operating.

The products of the gain multiplier are the AGC scaled outputs with a 19-bit representation. These are in turn used as I and Q for calculating the power, and the AGC error and loop are filtered to produce the signal gain for the next set of samples. These AGC-scaled outputs can be programmed to have 4-, 5-, 6-, 7-, 8-, 10-, 12-, or 16-bit widths by using the AGC output word length word in the AGC control register. The AGC-scaled outputs are truncated to the required bit widths by using the clipping circuitry, as shown in Figure 39.

Average Samples Setting

Though it is complicated to express the exact effect of the number of averaging samples by using equations, intuitively it has a smoothing effect on the way the AGC loop addresses a sudden increase or a spike in the signal level. If averaging of four samples is used, the AGC addresses a sudden increase in signal level more slowly compared to no averaging. The same applies to the manner in which the AGC addresses a sudden decrease in the signal level.

Desired Clipping Level Mode

Each AGC can be configured so that the loop locks onto a desired clipping level or a desired signal level. Desired clipping level mode is selected by writing Logic 1 in the AGC clipping error mode bit in the AGC control register. For signals that tend to exceed the bounds of the peak-to-average ratio, the desired

clipping level option provides a way to prevent truncating those signals and still provide an AGC that attacks quickly and settles to the desired output level. The signal path for this mode of operation is shown with dotted lines in Figure 39; the operation is similar to the desired signal level mode.

First, the data from the gain multiplier is truncated to a lower resolution (4 bits, 5 bits, 6 bits, 7 bits, 8 bits, 10 bits, 12 bits, or 16 bits) as set by the AGC output word length word in the AGC control register. An error term (for both I and Q) is generated that is the difference between the signals before and after truncation. This term is passed to the complex squared magnitude block, for averaging and decimating the update samples and taking their square root to find rms samples as in desired signal level mode. In place of the request desired signal level, a desired clipping level is subtracted, leaving an error term to be processed by the second-order loop filter.

The rest of the loop operates the same way as the desired signal level mode. This way, the truncation error is calculated and the AGC loop operates to maintain a constant truncation error level. The only register setting that is different from the desired signal level mode settings is that the desired clipping level is stored in the AGC desired level registers instead of in the request signal level.

AGC Synchronization

When the AGC output is connected to a RAKE receiver, the RAKE receiver can synchronize the average and update section to update the average power for AGC error calculation and loop filtering. This external sync signal synchronizes the AGC changes to the RAKE receiver and makes sure that the AGC gain word does not change over a symbol period, which, therefore, provides a more accurate estimation. This synchronization can be accomplished by setting the appropriate bits of the AGC control register.

Sync Select Alternatives

The AGC can receive a sync as follows:

- Channel sync: The sync signal is used to synchronize the NCO of the channel under consideration.
- Pin sync: Select one of the four SYNC pins.
- Sync now bit: Through the AGC control register.

When the channel sync select bit of the AGC control register is Logic 1, the AGC receives the SYNC signal used by the NCO of the corresponding channel for the start. When this bit is Logic 0, the pin sync defined by the 2-bit SYNC pin select word in the AGC control register is used to provide the sync to the AGC. Apart from these two methods, the AGC control register also has a sync now bit that can be used to provide a sync to the AGC by writing to this register through the microport or serial port.

Sync Process

Regardless of how a sync signal is received, the syncing process is the same. When a sync is received, a start hold-off counter is loaded with the 16-bit value in the AGC hold-off register, which initiates the countdown. The countdown is based on the ADC input clock. When the count reaches 1, a sync is initiated. When a sync is initiated, the CIC decimation filter dumps the current value to the square root, error estimation, and loop filter blocks. After dumping the current value, it starts working toward the next update value. Additionally on a sync, AGC can be initialized if the initialize AGC on sync bit is set in the AGC control register. During initialization, the CIC accumulator is cleared and new values for CIC decimation, number of averaging samples, CIC scale, signal gain, open-loop gains K_1 and K_2, and pole parameter P are loaded from their respective registers. When the initialize on sync bit is cleared, these parameters are not loaded from the registers.

This sync process is also initiated when a channel comes out of sleep by using the start sync to the NCO. An additional feature is the first sync only bit in the AGC control register. When this bit is set, the first sync initiates the process only and the remaining sync signals are ignored. This is useful when syncing using a pin sync. A sync is required on the first pulse on this pin only. These additional features make AGC synchronization more flexible and applicable to varied circumstances.

PARALLEL PORT OUTPUT

The AD6636 incorporates three independent 16-bit parallel ports for output data transfer. The three parallel output ports share a common clock, PCLK. Each port consists of a 16-bit data bus, a REQuest signal, an ACKnowledge signal, three channel indicator pins, one I/Q indicator pin, one gain word indicator pin, and a common shared PCLK pin. The parallel ports can be configured to function in master or slave mode. By default, the parallel ports are in slave mode on power-up.

Each parallel port can output data from any or all of the AGCs, using the 1-bit enable bit for each AGC in the parallel port control register. Even when the AGC is not required for a certain channel, the AGC can be bypassed, but the data is still received from the bypassed AGC. The parallel port functionality is programmable through the two parallel port control registers.

Each parallel port can be programmed individually to operate in either interleaved I/Q mode or parallel I/Q mode. The mode is selected using a 1-bit data format bit in the parallel port control register. In both modes, the AGC gain word output can be enabled using a 1-bit append gain bit in the parallel port control register for individual output ports. There are six enable bits per output port, one for each AGC in the corresponding parallel port.

Interleaved I/Q Mode

Parallel port channel mode is selected by writing 0 to the data format bit for the parallel port in consideration. In this mode, I and Q words from the AGC are output on the same 16-bit data bus on a time-multiplexed basis. The 16-bit I word is output followed by the 16-bit Q word. The specific AGCs output by the port are selected by setting individual bits for each of the AGCs in the parallel port control register. Figure 40 shows the timing diagram for the interleaved I/Q mode.

When an output data sample is available for output from an AGC, the parallel port initiates the transfer by pulling the PxREQ signal high. In response, the processor receiving the data needs to pull the PxACK signal high, acknowledging that it is ready to receive the signal. In Figure 40, PxACK is already pulled high and, therefore, the 16-bit I data is output on the data bus on the next PCLK rising edge after PxREQ is driven logic high. The PxIQ signal also goes high to indicate that I data is available on the data bus. The next PCLK cycle brings the Q data onto the data bus. In this cycle, the PxIQ signal is driven low. When I data and Q data are output, the channel indicator pins PxCH[2:0] indicate the data source (AGC number).

Figure 40 is the timing diagram for interleaved I/Q mode with the AGC gain word disabled. Figure 41 is a similar timing diagram with the AGC gain word. I and Q data are as explained for Figure 40. In the PCLK cycle after the Q data, the AGC gain word is output on the data bus and the PxGAIN signal is pulled high to indicate that the gain word is available on the parallel port. Therefore, a minimum of three or four PCLK cycles are required to output one sample of output data on the parallel port without or with the AGC gain word, respectively.

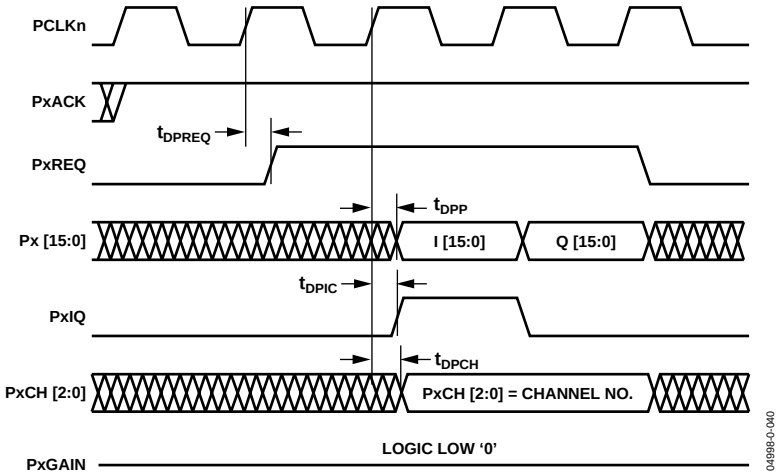


Figure 40. Interleaved I/Q Mode Without an AGC Gain Word

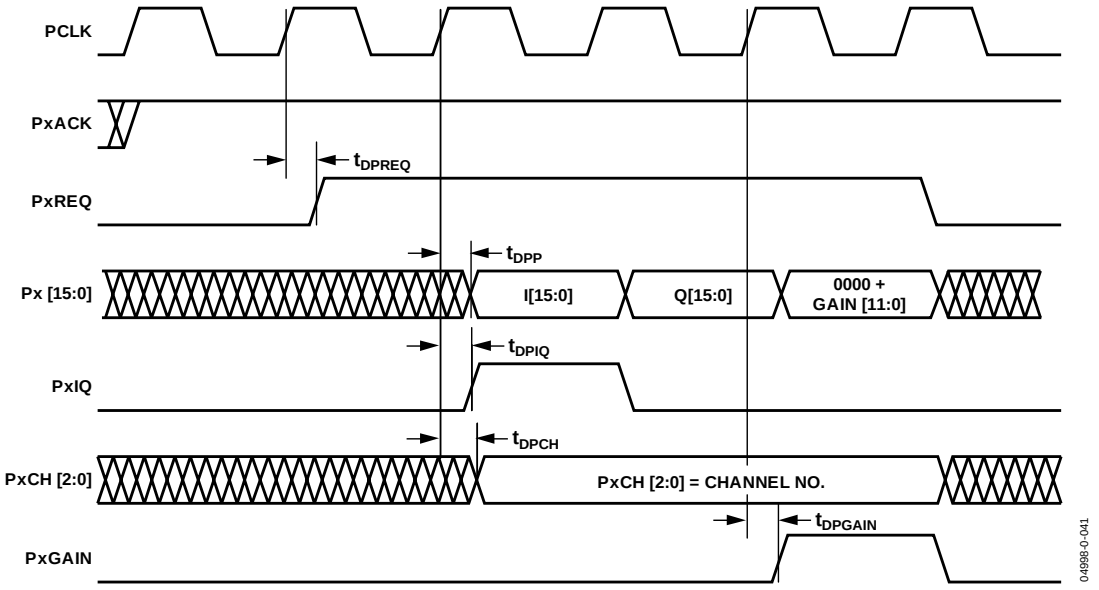


Figure 41. Interleaved I/Q Mode with an AGC Gain Word

Parallel IQ Mode

In this mode, eight bits of I data and eight bits of Q data are output on the data bus simultaneously during one PCLK cycle. The I byte is the most significant byte of the port, while the Q byte is the least significant byte. The PAIQ and PBIQ output indicator pins are set high during the PCLK cycle. Note that if data from multiple AGCs are output consecutively, the PAIQ and PBIQ output indicator pins remain high until data from all channels is output.

The PACH[2:0] and PBCH[2:0] pins provide a 3-bit binary value indicating the source (AGC number) of the data currently being output. Figure 42 is the timing diagram for parallel I/Q mode.

When an output data sample is available for output from an AGC, the parallel port initiates the transfer by pulling the PxREQ signal high. In response, the processor receiving the data needs to pull the PxACK signal high, acknowledging that it is ready to receive the signal.

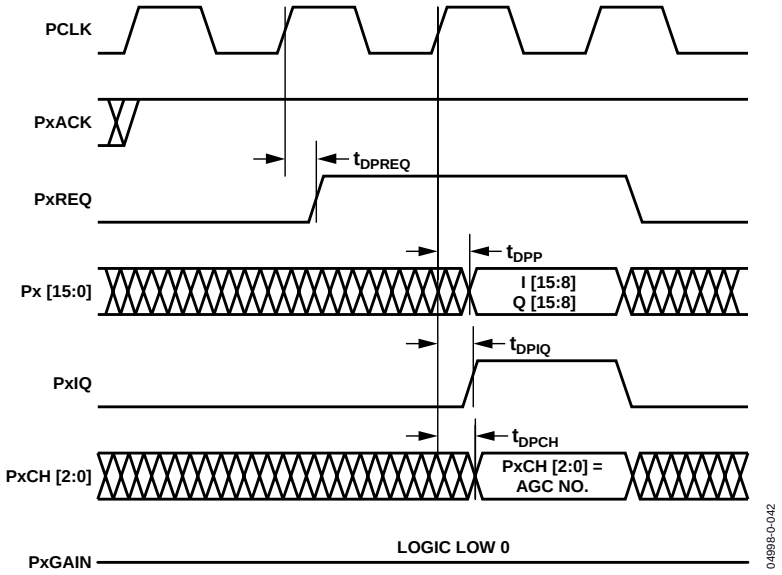


Figure 42. Parallel I/Q Mode Without an AGC Gain Word

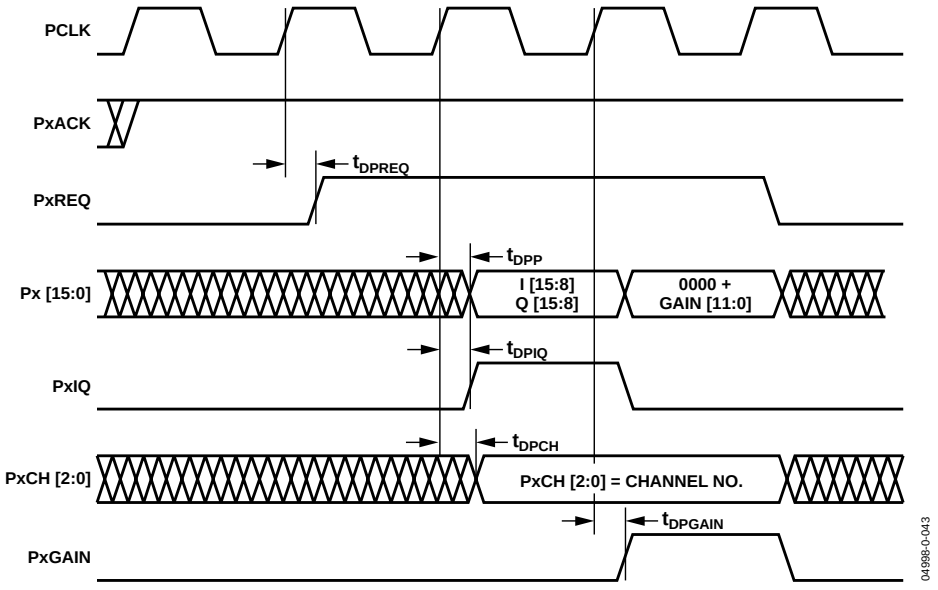


Figure 43. Parallel I/Q Mode with an AGC Gain Word

In Figure 42, the PxACK is already pulled high and, therefore, the 8-bit I data and 8-bit Q data are simultaneously output on the data bus on the next PCLK rising edge after PxREQ is driven logic high. The PxIQ signal also goes high to indicate that I/Q data is available on the data bus. When I/Q data is being output, the channel indicator pins PxCH[2:0] indicate the data source (AGC number).

Figure 42 is the timing diagram for interleaved I/Q mode with the AGC gain word disabled. Figure 43 is a similar timing diagram with the AGC gain word enabled. I and Q data are as shown in Figure 39. In the PCLK cycle after the I/Q data, the AGC gain word is output on the data bus, and the PxGAIN signal is pulled high to indicate that the gain word is available on the parallel port. During this PCLK cycle, the PxIQ signal is pulled low to indicate that I/Q data is not available on the data bus. Therefore, in parallel I/Q mode, a minimum of two PCLK cycles is required to output one sample of output data on the parallel port without and with the AGC gain word, respectively.

The order of data output is dependent on when data arrives at the port, which is a function of total decimation rate, DRCF/CRCF decimation phase, and start hold-off values. Priority order from highest to lowest is AGCs 0, 1, 2, 3, 4, and 5 for both parallel I/Q and interleaved modes of output.

Master/Slave PCLK Modes

The parallel ports can operate in either master or slave mode. The mode is set via the PCLK master mode bit in the Parallel Port Control 2 register. The parallel ports power up in slave mode to avoid possible contentions on the PCLK pin.

In master mode, PCLK is an output derived by dividing PLL_CLK down by the PCLK divisor. The PCLK divisor can have a value of 1, 2, 4, or 8, depending on the 2-bit PCLK divisor word setting in the Parallel Port Control 2 register. The highest PLCK rate in master mode is 200 MHz. Master mode is selected by setting the PCLK master mode bit in the Parallel Port Control 2 register.

$$PCLK\ Rate = \frac{PLL_CLK\ Rate}{PCLK\ Divisor}$$

In slave mode, external circuitry provides the PCLK signal. Slave mode PCLK signals can be either synchronous or asynchronous. The maximum slave mode PCLK frequency is also 200 MHz.

Parallel Port Pin Functions

Table 25 describes the functions of the pins used by the parallel ports.

Table 25. Parallel Port Pin Functions

Mnemonic	I/O	Function
PCLK	I/O	PCLK can operate as a master or as a slave. This setting is dependent on the 1-bit PCLK master mode bit in the Parallel Port Control 2 register. As an output (master mode), the maximum frequency is CLK/N, where CLK is AD6636 clock and N is an integer divisor of 1, 2, 4, or 8. As an input (slave mode), it can be asynchronous or synchronous relative to the AD6636 CLK. This pin powers up as an input to avoid possible contentions. Parallel port output pins change on the rising edge of PCLK.
PAREQ, PBREQ, PCREQ	O	Active high output. Synchronous to PCLK. A logic high on this pin indicates that data is available to be shifted out of the port. When an acknowledge signal is received, data starts shifting out and this pin remains high until all pending data has been shifted out.
PAACK, PBACK, PCACK	I	Active high asynchronous input. Applying a logic low on this pin inhibits parallel port data shifting. Applying a logic high to this pin when REQ is high causes the parallel port to shift out data according to the programmed data mode. ACK is sampled on the rising edge of PCLK. Assuming that REQ is asserted, the latency from the assertion of ACK to data appearing at the parallel port output is no more than 1.5 PCLK cycles. ACK can be held high continuously; in this case, when data becomes available, shifting begins 1 PCLK cycle after the assertion of REQ (see Figure 40, Figure 41, Figure 42, and Figure 43).
PAIQ, PBIQ, PCIQ		High whenever I data is present on the parallel port data bus; otherwise low. In parallel I/Q mode, both I data and Q data are available at the same time and, therefore, the PxIQ signal is pulled high.
PAGAIN, PBGAIN, PCGAIN		High whenever the AGC gain word is present on the parallel port data bus; otherwise low.
PACH[2:0], PBCH[2:0], PCCH[2:0]		These pins identify data in both of the parallel port modes. The 3-bit value identifies the source of the data (AGC number) on the parallel port when it is being shifted out.
PADATA[15:0], PBDATA[15:0], PCDATA[15:0]		Parallel output port data bus. Output format is twos complement. In parallel I/Q mode, 8-bit data is present; in interleaved I/Q mode, 16-bit data is available.

USER-CONFIGURABLE, BUILT-IN SELF-TEST (BIST)

Each channel of AD6636 includes a BIST block. The BIST, along with an internal test signal (pseudorandom test input signal), can be used to generate a signature. This signature can be compared with a known good device and an untested device to see if the untested device is functional.

BIST timer bits in the BIST control register can be programmed with a timer value that determines the number of clock cycles that the output of the channels (output of AGC) have accumulated. When the disable signature generation bit is written with Logic 0, the BIST timer is counted down and a signature register is written with the accumulated output of the AD6636 channel.

When the BIST timer expires, the signature register for I and Q paths can be read back to compare it with the signature register from a known good device.

CHIP SYNCHRONIZATION

The AD6636 offers two types of synchronization: start sync and hop sync. Start sync is used to bring individual channels out of sleep after programming. It can also be used while AD6636 is operational to resynchronize the internal clocks. Hop sync is used to change or update the NCO frequency tuning word and the NCO phase offset word.

Two methods can be used to initiate a start sync or hop sync:

- Soft sync is provided by the memory map registers and is applied to channels directly through the microport or serial port interface.
- Pin sync is provided using four hard-wired SYNC[3:0] pins. Each channel is programmed to listen to one of these SYNC pins and do a start sync or a hop sync when a signal is received on these pins.

The pin synchronization configuration register (Address 0x04) is used to make pin synchronization even more flexible. The part can be programmed to be edge-sensitive or level-sensitive for SYNC pins. In edge-sensitive mode, a rising edge on the SYNC pins is recognized as a synchronization event.

Start

Start refers to the startup of an individual channel or chip, or of multiple chips. If a channel is not used, it should be put into sleep mode to reduce power dissipation. Following a hard reset (low pulse on the $\overline{\text{RESET}}$ pin), all channels are placed into sleep mode. Alternatively, channels can be put to sleep manually by writing 0 to the sleep register.

Start with Soft Sync

The AD6636 can synchronize channels or chips under microprocessor control. The start hold-off counter, in conjunction with the soft start enable bit and the channel enable bits, enables this synchronization.

To synchronize the start of multiple channels via microprocessor control:

1. Write the channel enable register to enable one or more channels, if the channels are inactive.
2. Write the NCO start hold-off counter register(s) with the appropriate value (greater than 0 and less than 2^{16}).
3. Write 0x00 to the soft synchronization configuration register.
4. Write the soft sync channel enable bit(s) and soft start synchronization enable bit high in the soft synchronization configuration register. This starts the countdown by the start hold-off counter. When the count reaches 1, the channels are activated or resynchronized.

Note that when using SPI or SPORT for programming these registers, the last step in the above procedure needs to be repeated. Therefore, the soft synchronization configuration register is written twice.

Start with Pin Sync

Four sync pins (0, 1, 2, and 3) provide very accurate synchronization among channels. Each channel can be programmed to monitor any of the four sync pins.

To start the channels with a pin sync:

1. Write the channel register to enable one more channels, if the channels are inactive.
2. Write the NCO start hold-off counter register(s) with the appropriate value (greater than 0 and less than 2^{16}).
3. Program the channel NCO control registers to monitor the appropriate SYNC pins.
4. Write the start synchronization enable bit and SYNC pin enable bits high in the pin synchronization configuration register. This starts the countdown of the start hold-off counter. When the count reaches 1, the channels are activated or resynchronized.

Hop

Hop is a jump from one NCO frequency and/or phase offset to a new NCO frequency and/or phase offset. This change in frequency and/or phase offset can be synchronized via microprocessor control (soft sync) or via an external sync signal (pin sync).

Hop with Soft Sync

The AD6636 can synchronize a change in NCO frequency and/or phase offset of multiple channels or chips under microprocessor control. The NCO hop hold-off counter, in conjunction with the soft hop enable bit and the channel enable bits, enables this synchronization.

To synchronize the hop of multiple channels via microprocessor control:

1. Write the NCO frequency register(s) or phase offset register(s) to the new value.
2. Write the NCO frequency hold-off counter register(s) with the appropriate value (greater than 0 and less than 2^{16}).
3. Write 0x00 to the soft synchronization configuration register.
4. Write the soft hop synchronization enable bit and the corresponding soft sync channel enable bits high in the soft synchronization configuration register. This starts the countdown by the frequency hold-off counter. When the count reaches 1, the new frequency and/or phase offset is loaded into the NCO.

Note that when using SPI or SPORT for programming these registers, the last step in the above procedure needs to be repeated. Therefore, the soft synchronization configuration register is written twice.

Hop with Pin Sync

Four sync pins (0, 1, 2 and 3) provide very accurate synchronization among channels. Each channel can be programmed to look at any of the four sync pins.

To control the hop of channel NCO frequencies:

1. Write the NCO frequency register(s) or phase offset register(s) to the new value.
2. Write the NCO frequency hold-off counter(s) to the appropriate value (greater than 0 and less than 2^{16}).
3. Program the channel NCO control registers to monitor the appropriate SYNC pins.
4. Write the hop synchronization enable bit and SYNC pin enable bits high in the pin synchronization configuration register. This enables the countdown of the frequency hold-off counter. When the reaches 1, the new frequency and/or phase offset is loaded into the NCO.

SERIAL PORT CONTROL

The AD6636 serial port allows all memory to be accessed (programmed or readback) serially in one-byte words. Either serial port or microport can be used (but not both) at any given time. Serial port control is selected using the SMODE pin (0 = microport, 1 = serial port). Two serial port modes are available. An SPI-compatible port is provided as well as a SPORT. The choice of SPI or SPORT mode is selected using the MODE pin (0 = SPI, 1 = SPORT).

Each individual byte of serial data (address, instruction, and data) may be shifted in either MSB first or LSB first using the MSB_FIRST pin (1 = MSB first, 0 = LSB first). The serial chip select ($\overline{SCS}$) pin is brought low to access the device for serial control. When the $\overline{SCS}$ pin is held high, serial programming is inhibited.

Hardware Interface

The pins described in Table 26 comprise the physical interface between the user's programming device and the serial port of the AD6636. All serial pins are inputs except for SDO, which is an open-drain output and should be pulled high by an external pull-up resistor (suggested value 1 kΩ).

A complete read or write cycle requires a minimum of three bytes to transfer, consisting of address word, instruction word, and data-word(s). As many as 127 data-words can be transferred during a block transfer cycle. All address, instruction, and data-word(s) must be formatted LSB first or MSB first to match the state of the MSB_FIRST pin.

The first word for serial transfer is the internal register address. In LSB first mode, the address is the lower-most address for the block transfer (subsequent addresses are generated by internal increment). In MSB first, the address is highest address for the block transfer (subsequent addresses are generated by internal decrement).

The second word of serial transfer contains a one-bit read/write indicator (1 = read, 0 = write), and seven bits to define the number of data bytes to be transferred (N). For a single data byte transfer (N = 1); one byte is shifted into SDI for a write transfer, or shifted out of SDO for a read transfer, and the cycle is complete. For a block transfer, N write/read operations are performed, and the internal register address increments (MSB_FIRST = 0) or decrements (MSB_FIRST = 1) after each data byte is clocked into SDI for a write operation, or after each data byte is clocked out of SDO for a read operation.

Figure 44 to Figure 47 illustrate a three byte block transfer through the serial port. Read and write operations with MSB_FIRST high and low are shown. Note that the figures show the sequence for write/read transfer, and actual data should be shifted in or out based upon the status of the MSB_FIRST pin. The operation details are common to both SPI and SPORT modes, except for the use of framing signals and timing. Individual mode details follow. In single-byte transfer mode, the count in the second byte is reduced to 1, and the number of data bytes is reduced to 1.

Table 26. Serial Port Pins

Pin	Function
SCLK	Serial Clock in Both SPI and SPORT Modes. Should have a rise/fall time of 3 ns maximum.
MSB_FIRST	Indicates whether the first bit shifted in or out of the serial port is the MSB (1) or LSB (0) for both instruction and data-words. Also indicates if the first instruction word (address) is a block start or a block end for multiple byte transfers. This pin also controls the functionality when programming indirectly addressed registers.
STFS	Serial Transmit Frame Sync in SPORT Mode. STFS is not used in SPI mode.
SRFS	Serial Receive Frame Sync in SPORT Mode. SRFS is not used in SPI mode.
SDI	Serial Data Input in Both Modes. Serial data is clocked in on the rising edge of SCLK.
SDO	Serial Data Output in Both Modes. Serial data is clocked out on the rising edge of SCLK.
$\overline{SCS}$	Active-Low Serial Chip Select in Both Modes.
SMODE	Serial Mode. Part is programmed through the serial port when this pin is high.
MODE	Mode Pin. Selects between SPI (0) and SPORT (1) modes.

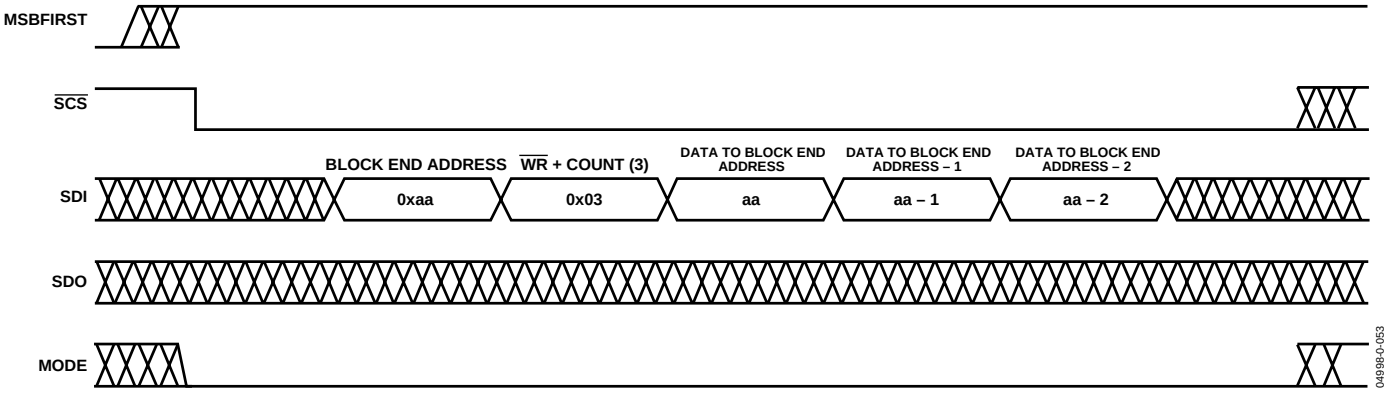


Figure 44. Serial Write of Three Bytes with MSB_FIRST = 1 (All Words are Written MSB First)

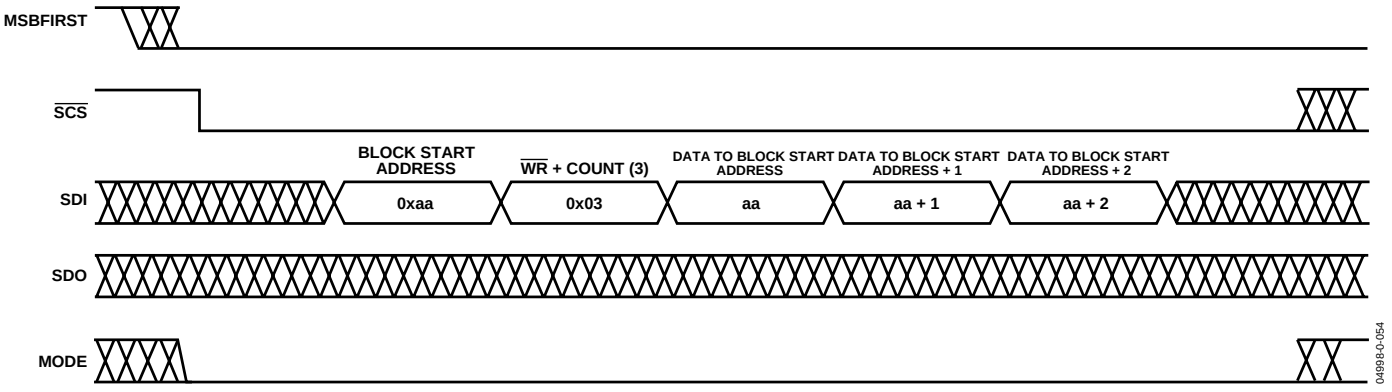


Figure 45. Serial Write of Three Bytes with MSB_FIRST = 0 (All Words are Written LSB First)

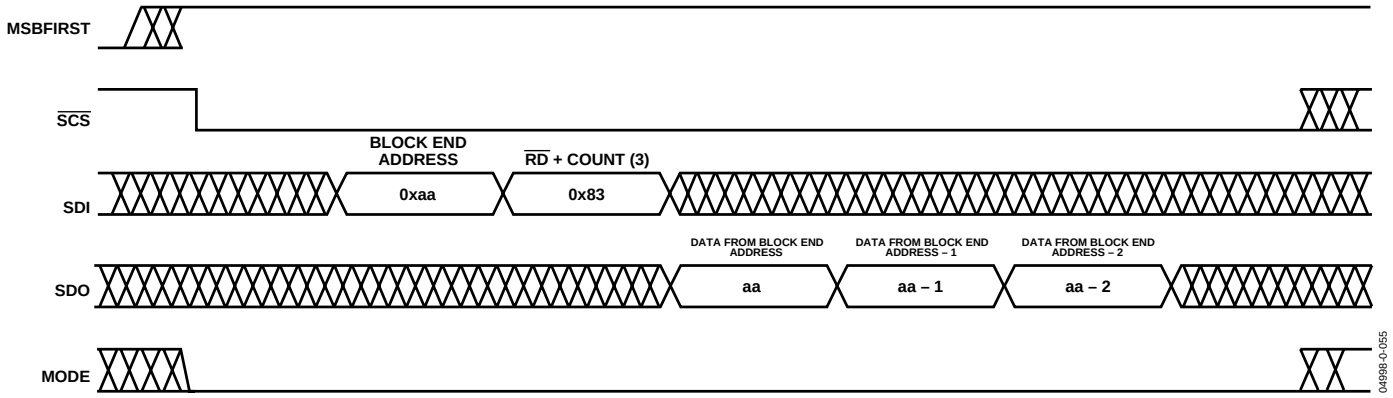


Figure 46. Serial Read of Three Bytes with MSB_FIRST = 1 (All Words are Written or Read MSB First)

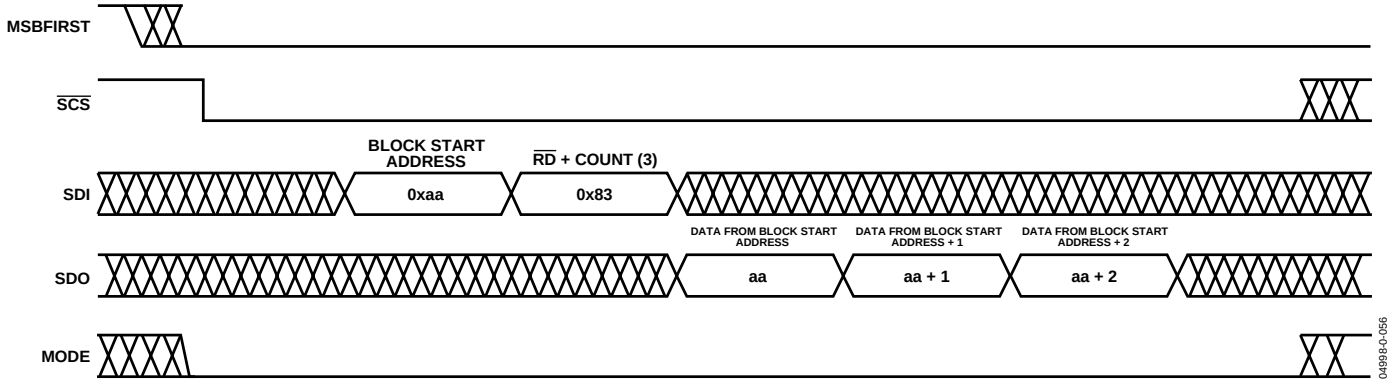


Figure 47. Serial Read of Three Bytes with MSB_FIRST = 0 (All Words are Written or Read LSB First)

SPI Mode Timing

In SPI mode, the SCLK should run only when data is being transferred and $\overline{SCS}$ is logic low. If SCLK runs when $\overline{SCS}$ is logic high, the internal shift register continues to run and instruction words or data are lost. No external framing is necessary. The $\overline{SCS}$ pin can be pulled low once for each byte of transfer, or kept low for the whole length of the transfer.

SPI Write

Data on the SDI pin is registered on the rising edge of SCLK. During a write, the serial port accumulates eight input bits of data before transferring one byte to the internal registers. Figure 48 and Figure 49 show one byte block transfer for writing in MSB_FIRST and LSB_FIRST modes.

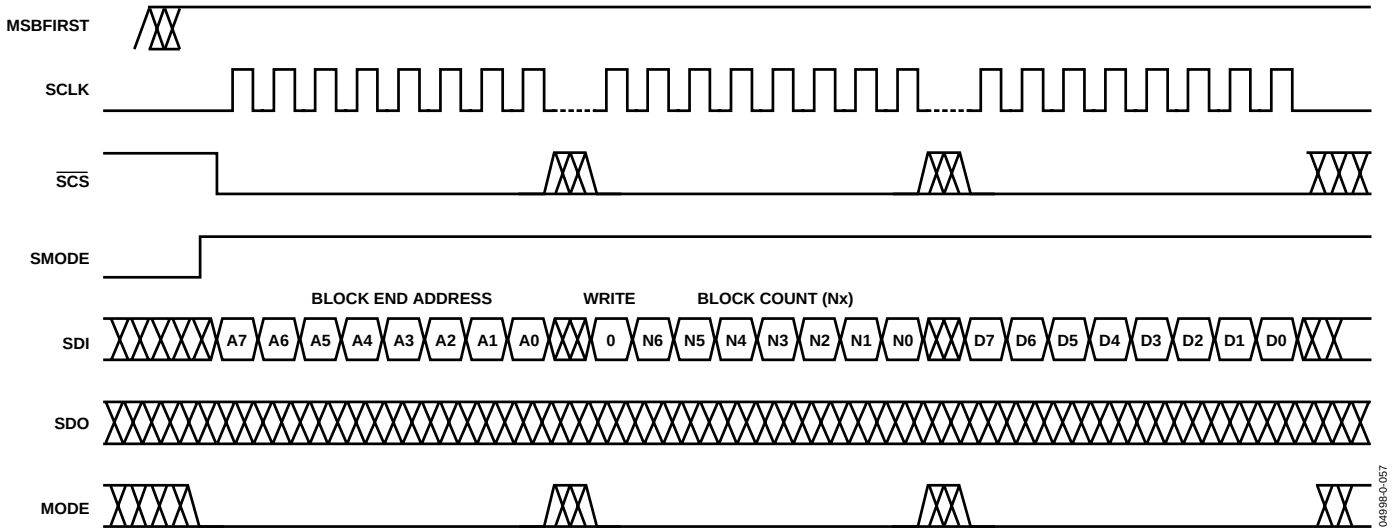


Figure 48. SPI Write MSB_FIRST = 1

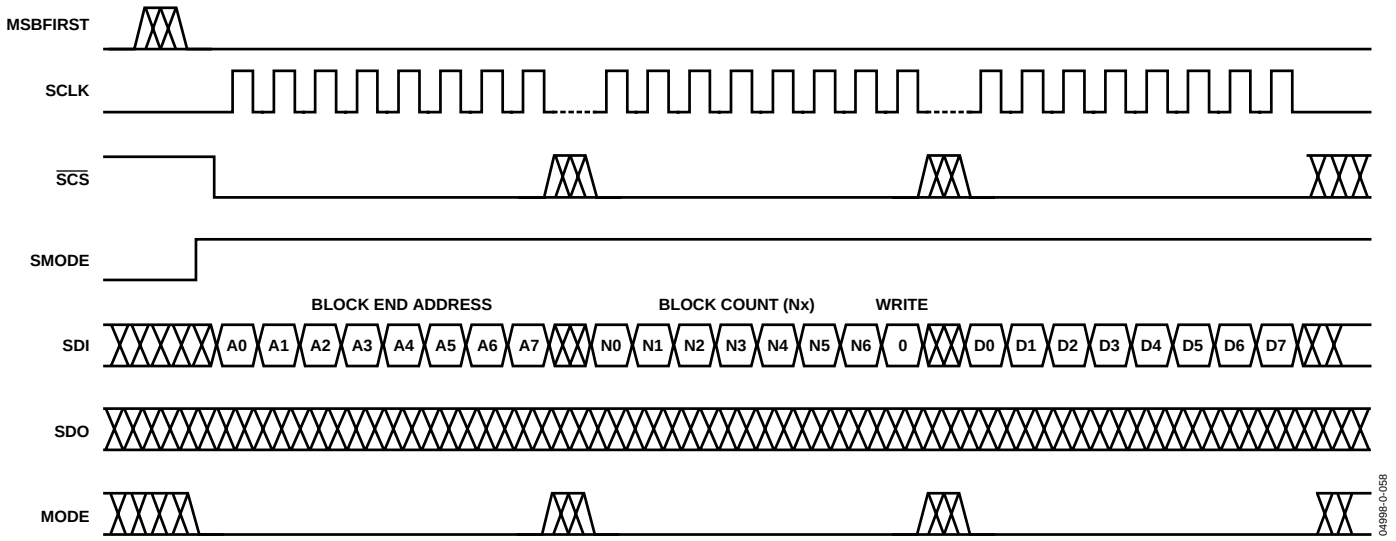


Figure 49. SPI Write MSB_FIRST = 0

SPI Read

During a typical read operation, a one-byte address and one-byte instruction are written to the serial port to instruct the internal control logic as to which registers are to be accessed. Register readback data shifts out on the rising edge of SCLK. The SDO pin is in a high impedance state at all times except during a read cycle.

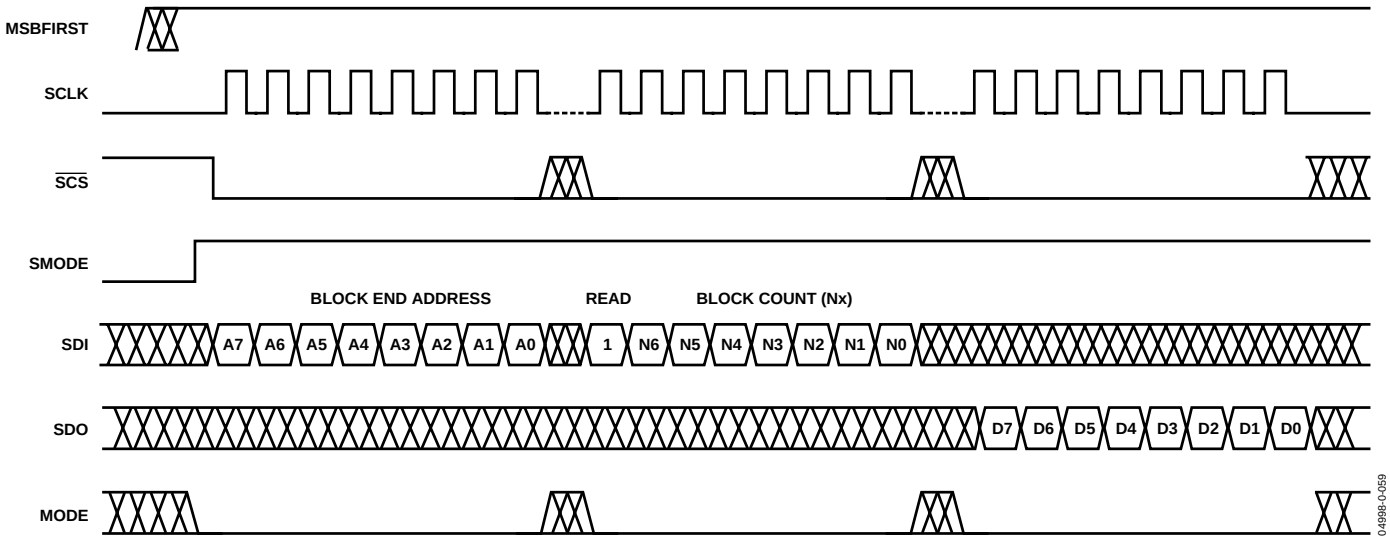


Figure 50. SPI Read MSB_FIRST = 1

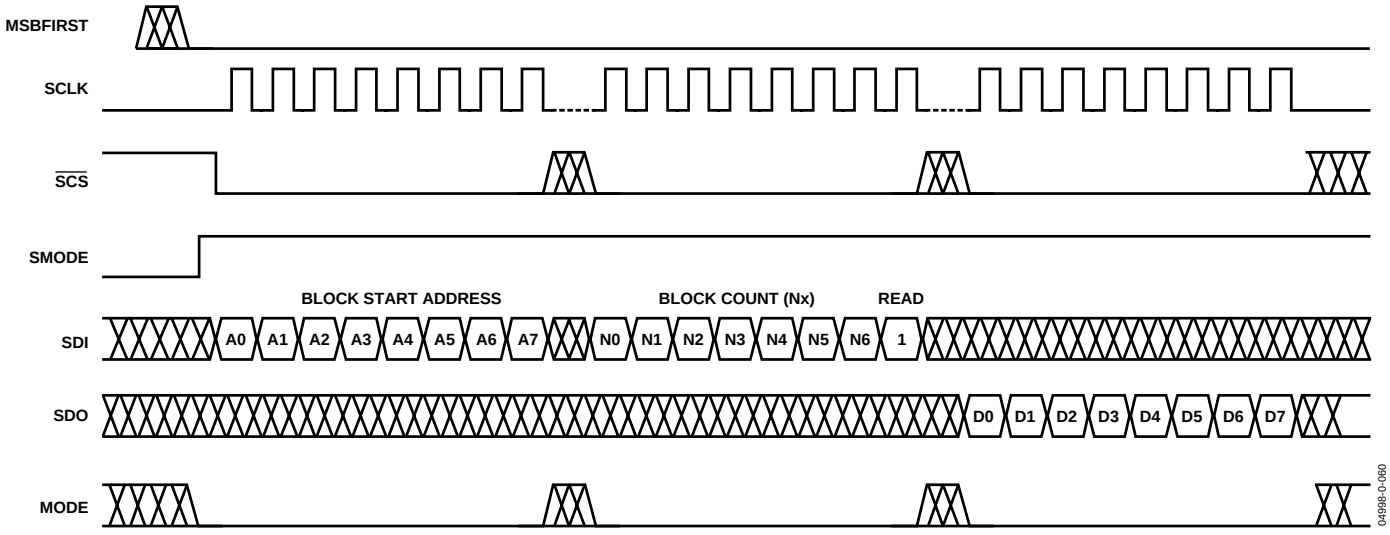


Figure 51. SPI Read MSB_FIRST = 0

SPORT Mode Timing

In SPORT mode, the SCLK continuously runs, and the external SRFS and STFS signals are used to frame the data. Incoming framing signals SRFS (receive) and STFS (transmit) are sampled on the falling edges of SCLK. All input and output data must be transmitted or received in 8-bit segments starting with the rising edge after SRFS or STFS is sampled.

SPORT Write

Serial data is sampled on the rising edge of SCLK. The data should be MSB or LSB first, depending on the polarity of the MSB_FIRST pin. The serial port begins to sample data on the rising edge of SCLK after SRFS is detected on the falling edge of SCLK. Once all 8 bits of one byte are shifted in, the data is transferred to the internal bus.

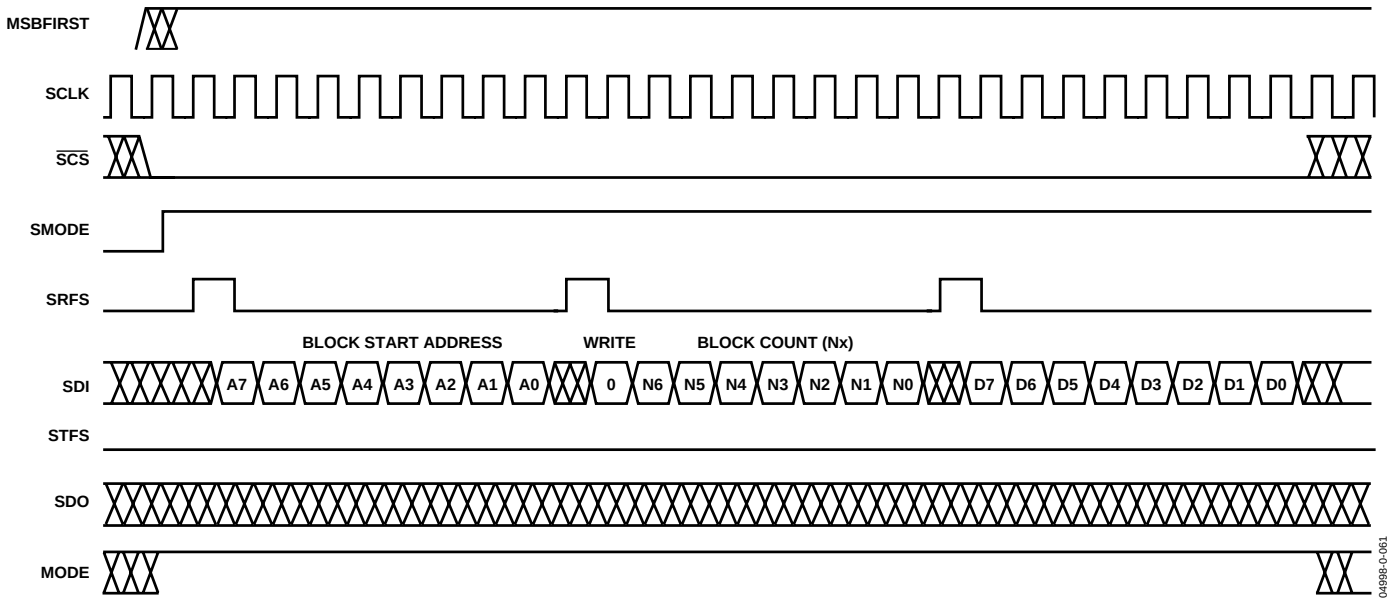


Figure 52. SPORT Write MSB_FIRST = 1

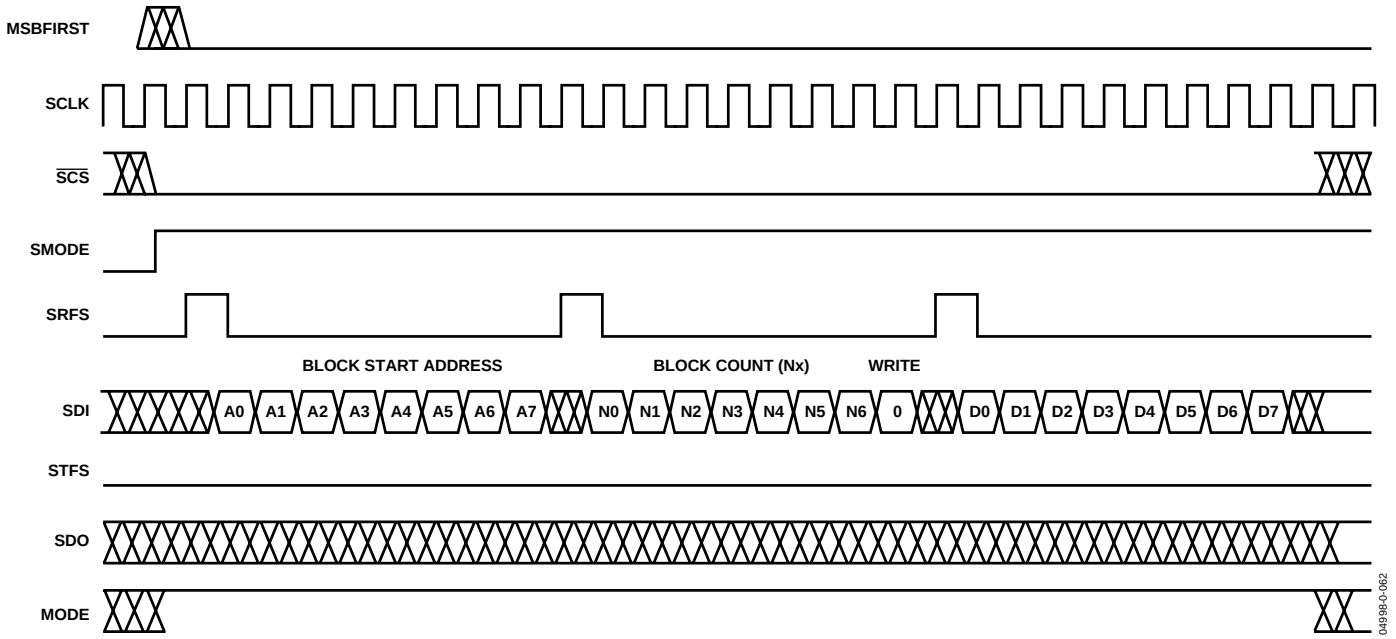


Figure 53. SPORT Write MSB_FIRST = 0

SPORT Read

For a typical SPORT read operation, the user must write an address byte and instruction byte to the serial port to instruct the internal control logic as to which registers are to be readback. STFS must be asserted for every 8-bit readback and is sampled on the falling edge of SCLK. Data is shifted out on the rising edge of SCLK. The SDO pin is in a high impedance state at all times except during a read operation.

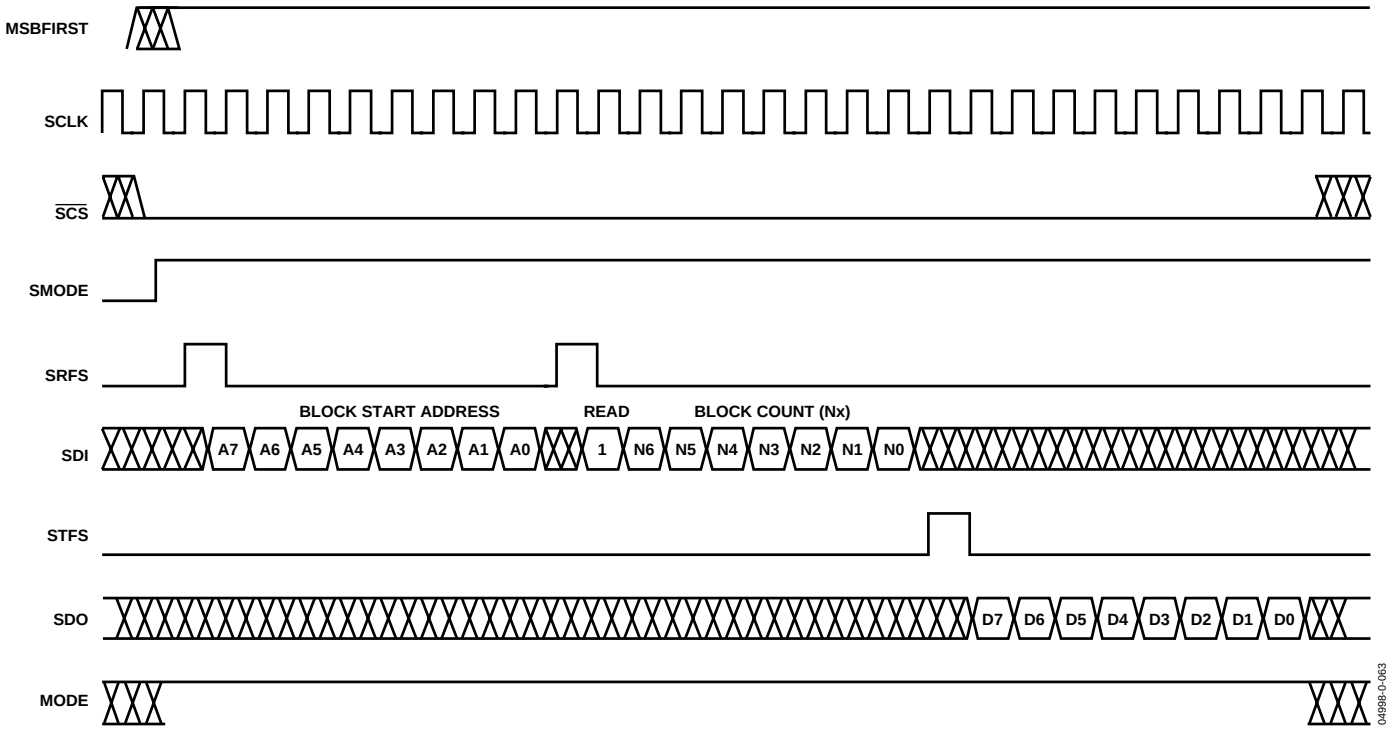


Figure 54. SPORT Read MSB_FIRST = 1

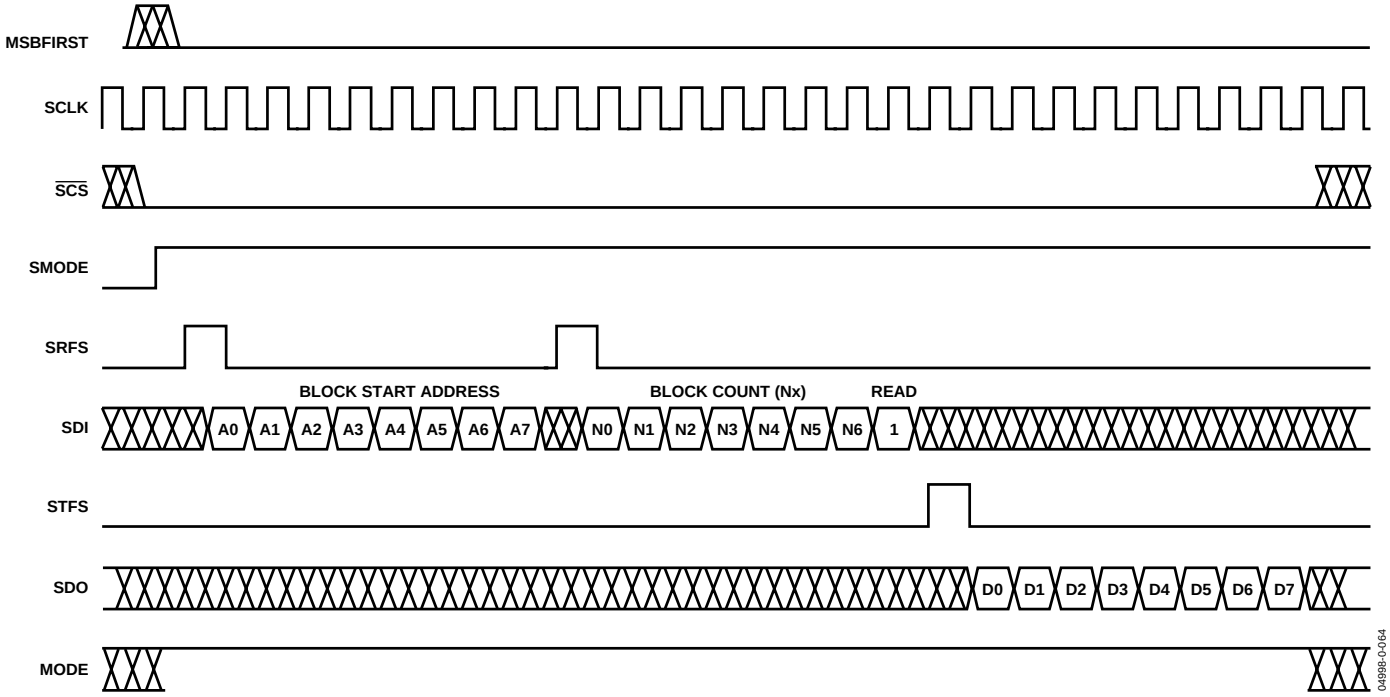


Figure 55. SPORT Read MSB_FIRST = 0

Programming Indirect Addressed Registers Using Serial Port

This section gives examples for programming CRCF coefficient RAM (with an indirect addressing scheme) using the serial port (either SPI or SPORT modes). Though the following specific examples are for CRCF coefficient RAM programming, they can be extended to other indirect addressed registers such as DRCF coefficient RAM. There are four possible programming scenarios, and examples are given for all scenarios using two commands: SerialWrite (data) and SerialRead. These commands signify an 8-bit write to, or an 8-bit read from, the serial port (SPI or SPORT).

SerialWrite (8-bit number): This is an 8-bit write to SPI or SPORT. In SPI mode, the SCLK is toggled eight times while $\overline{SCS}$ is pulled low. In SPORT mode, $\overline{SCS}$ is pulled low, SRFS is held high for one SCLK cycle, and eight bits of data are shifted into the SDI pin following the SRFS pulse. Though the 8-bit number argument shown in the following code is always shown MSB_FIRST, it is written with MSB shifting into the device first in MSB_FIRST mode, and it is written with LSB shifting into the device first in LSB_FIRST mode.

SerialRead(): This is an 8-bit read from the SDO pin in SPI or SPORT modes. In SPI mode, the SCLK toggles eight times while $\overline{SCS}$ is low. In SPORT mode, $\overline{SCS}$ is pulled low, STFS is held high for one SCLK cycle, and then the eight bits of data that shifted out on SDO following the STFS pulse are read. The data shifted out should be interpreted based on the polarity of the MSB_FIRST pin.

MSB_FIRST Mode Using Single-Byte Block Transfers

```
SerialWrite(0x98); //CRCF Start Address
SerialWrite(0x01);
SerialWrite(0x00);

SerialWrite(0x99); //CRCF Final Address
SerialWrite(0x01);
SerialWrite(N-1);  //N is the number of coefficients

for (i=0 ; i < N; i++) {      //writing registers

    SerialWrite(0x9E);        //MSB written first
    SerialWrite(0x01);
    //data bits[23:16]
    SerialWrite(coeff[i] >> 16 & 0xFF);

    SerialWrite(0x9D);
    SerialWrite(0x01);
    //data bits[15:8]
    SerialWrite(coeff[i] >> 8 & 0xFF);

    SerialWrite(0x9C);        //LSB written last
    SerialWrite(0x01);
    //data bits[7:0]
    SerialWrite(coeff[i] & 0xFF);

}

SerialWrite(0x98); //CRCF Start Address
SerialWrite(0x01);
SerialWrite(0x00);

SerialWrite(0x99); //CRCF Final Address
SerialWrite(0x01);
SerialWrite(N-1);  //N is the number of coefficients

for (i=0 ; i < N; i++) {      //reading registers

    SerialWrite(0x9E);        //MSB readback first
    SerialWrite(0x81);
    //data bits[23:16]
    Coeff[i] = SerialRead() << 16;

    SerialWrite(0x9D);
    SerialWrite(0x81);
    //data bits[15:8]
    Coeff[i] |= SerialRead() << 8;

    SerialWrite(0x9C);        //LSB readback last
    SerialWrite(0x81);
    //data bits[7:0]
    Coeff[i] |= SerialRead();

}
```

LSB_FIRST Mode Using Single-Byte Block Transfers

```
SerialWrite(0x98); //CRCF Start Address
SerialWrite(0x01);
SerialWrite(0x00);

SerialWrite(0x99); //CRCF Final Address
SerialWrite(0x01);
SerialWrite(N-1);  //N is the number of coefficients

for (i=0 ; i < N; i++) {      // writing registers

    SerialWrite(0x9C);        //LSB written first
    SerialWrite(0x01);
    //data bits[7:0]
    SerialWrite(coeff[i] & 0xFF);

    SerialWrite(0x9D);
    SerialWrite(0x01);
    //data bits[15:8]
    SerialWrite(coeff[i] >> 8 & 0xFF);

    SerialWrite(0x9E);        //MSB written last
    SerialWrite(0x01);
    //data bits[23:16]
    SerialWrite(coeff[i] >> 16 & 0xFF);

}

SerialWrite(0x98); //CRCF Start Address
SerialWrite(0x01);
SerialWrite(0x00);

SerialWrite(0x99); //CRCF Final Address
SerialWrite(0x01);
SerialWrite(N-1);  //N is the number of coefficients

for (i=0 ; i < N; i++) {      //reading registers

    SerialWrite(0x9C);        //LSB readback first
    SerialWrite(0x81);
    //data bits[7:0]
    Coeff[i] = SerialRead();

    SerialWrite(0x9D);
    SerialWrite(0x81);
    //data bits[15:8]
    Coeff[i] |= SerialRead() << 8;

    SerialWrite(0x9E);        //MSB readback last
    SerialWrite(0x81);
    //data bits[23:16]
    Coeff[i] |= SerialRead() << 16;

}
```

MSB_FIRST Mode Using Multibyte Block Transfers

```
SerialWrite(0x99); //CRCF Final Address
SerialWrite(0x02);
SerialWrite(N-1);  //N is the number of coefficients
SerialWrite(0x00);

for (i=0 ; i < N; i++) {      //writing registers

    SerialWrite(0x9E);
    SerialWrite(0x03);
    //data bits[23:16]
    SerialWrite(coeff[i] >> 16 & 0xFF);
    //data bits[15:8]
    SerialWrite(coeff[i] >> 8 & 0xFF);
    //data bits[7:0]
    SerialWrite(coeff[i] & 0xFF);

}

SerialWrite(0x99); //CRCF Final Address
SerialWrite(0x02);
SerialWrite(N-1);  //N is the number of coefficients
SerialWrite(0x00);

for (i=0 ; i < N; i++) {      //reading registers

    SerialWrite(0x9E);
    SerialWrite(0x83);
    //data bits[23:16]
    Coeff[i] = SerialRead() << 16;
    //data bits[15:8]
    Coeff[i] |= SerialRead() << 8;
    //data bits[7:0]
    Coeff[i] |= SerialRead();

}
```

LSB_FIRST Mode Using Multibyte Block Transfers

```
SerialWrite(0x98); //CRCF Start Address
SerialWrite(0x02);
SerialWrite(0x00);
SerialWrite(N-1);  //N is the number of coefficients

for (i=0 ; i < N; i++) {      //writing registers

    SerialWrite(0x9C);
    SerialWrite(0x03);
    //data bits[7:0]
    SerialWrite(coeff[i] & 0xFF);
    //data bits[15:8]
    SerialWrite(coeff[i] >> 8 & 0xFF);
    //data bits[23:16]
    SerialWrite(coeff[i] >> 16 & 0xFF);

}

SerialWrite(0x98); //CRCF Start Address
SerialWrite(0x02);
SerialWrite(0x00);
SerialWrite(N-1);  //N is the number of coefficients

for (i=0 ; i < N; i++) {      //reading registers

    SerialWrite(0x9C);
    SerialWrite(0x83);
    //data bits[7:0]
    Coeff[i] = SerialRead();
    //data bits[15:8]
    Coeff[i] |= SerialRead() << 8;
    //data bits[23:16]
    Coeff[i] |= SerialRead() << 16;
}
```

Connecting the AD6654 Serial Port to a Blackfin DSP

In SPI mode, the Black*fin*® DSP must act as a master to the AD6636 by providing the SCLK. SDO is an open-drain output, so that multiple slave devices can be connected together. Figure 56 shows a typical connection.

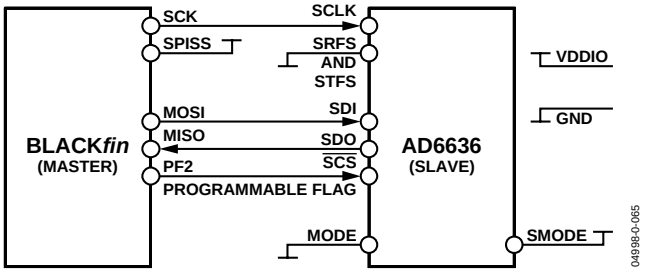


Figure 56. SPI Mode Serial Port Connection to Blackfin DSP

In SPORT mode, the Black*fin* provides the SCLK, SRFS, and STFS signals, as shown in Figure 57.

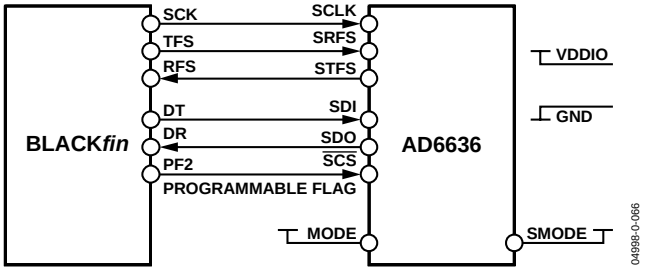


Figure 57. SPORT Mode Serial Port Connections to Blackfin

MICROPORT

The microport on the AD6636 can be used for programming the part, reading register values, and reading output data (I, Q, and RSSI words).

Note that, at any given point in time, either the microport or the serial port can be active, but not both. Some of the balls on the package are shared between the microport and the serial port and have dual functionality based on the SMODE pin. The microport is selected by pulling the SMODE pin low (ground).

Both read and write operations can be performed using the microport. The direct addressing scheme is used and any internal register can be accessed using an 8-bit address. The data bus can be either 8-bit or 16-bit as set by the chip I/O access control register. Microport operation is synchronous to CPUCLK, which must be supplied external to the AD6636 part. CPUCLK should be less than CLKA and 100 MHz.

The microport can operate in Intel® mode (separate read and write strobes) or in Motorola mode (single read/write strobe). The MODE pin is used to select between Intel (INM, MODE = 0) and Motorola (MNM, MODE = 1) modes. Some AD6636 pins have dual functionality based on the MODE pin. Table 27 lists the pin functions for both modes.

Table 27. Microport Programming Pins

Mnemonic	Intel Mode	Motorola Mode
$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
SMODE	Logic 0	Logic 0
MODE	Logic 0	Logic 1
A[7:0]	A[7:0]	A[7:0]
D[15:0]	D[15:0]	D[15:0]
R/$\overline{\text{W}}$ ($\overline{\text{WR}}$)	$\overline{\text{WR}}$	R/$\overline{\text{W}}$
$\overline{\text{DS}}$ ($\overline{\text{RD}}$)	$\overline{\text{RD}}$	$\overline{\text{DS}}$
$\overline{\text{DTACK}}$ (RDY)	RDY	$\overline{\text{DTACK}}$
$\overline{\text{CS}}$	$\overline{\text{CS}}$	$\overline{\text{CS}}$

Intel (INM) Mode

The programming port performs synchronous Intel-style reads and writes on the positive edge of the CPUCLK input when $\overline{\text{RESET}}$ is inactive (active low signal). The CPUCLK pin is driven by the programming device (CPUCLK of DSP or FPGA). During a write access, the A[7:0] address bus provides the address for access, and the D[15:0] bus (D[7:0] if the 8-bit data bus is used) is driven by the programming device. The data bus is driven by the AD6636 during a read operation. Intel mode uses separate read ($\overline{\text{RD}}$) and write ($\overline{\text{WR}}$) active-low data strobes to indicate both the type of access and the valid data for that access.

The chip select ($\overline{\text{CS}}$) is an active-low input that signals when an access is active on its programming port pins. During an access, the AD6636 drives RDY low to indicate that it is performing the access. When the internal read or write access is complete, the RDY pin is pulled high. Because the RDY pin is an open-drain output with a weak internal pull-up resistor (70 kΩ), an external pull-up resistor is recommended (see Figure 58). Figure 13 and Figure 14 are the timing diagrams for read and write cycles using the microport in INM mode. Do not read or write, to or from, addresses beyond those defined by the memory map (Address 0xE8 to Address 0xFF). Attempting to access these addresses causes the bus to hang because RDY does not go high to signal the end of the access.

For an asynchronous write operation in Intel (INM) mode, the CPUCLK should be running. Set up the data and address buses. Pull the $\overline{\text{WR}}$ signal low and then pull the $\overline{\text{CS}}$ signal low. The RDY goes low to indicate that the access is taking place internally. When RDY goes high, the write cycle is complete and $\overline{\text{CS}}$ can be pulled high to disable the microport.

For an asynchronous read operation on the Intel mode microport, set up the address bus and three-state the data bus. Pull the $\overline{\text{RD}}$ signal low and then pull the $\overline{\text{CS}}$ signal low. The RDY goes low to indicate an internal access. When RDY goes high, valid data is available on the data bus for read.

Motorola (MNM) Mode

The programming port performs synchronous Motorola-style reads and writes on the positive edge of CPUCLK when $\overline{\text{RESET}}$ is inactive (active low signal). The A[7:0] bus provides the address to access and the D[15:0] bus (D[7:0], if the 8-bit data bus is used) is externally driven with data during a write (driven by the AD6636 during a read). Motorola mode uses the R/$\overline{\text{W}}$ line to indicate the type of access (Logic 1 = read, Logic 0 = write), and the active-low data strobe ($\overline{\text{DS}}$) signal is used to indicate valid data.

The chip select ($\overline{\text{CS}}$) is an active-low input that signals when an access is active on its programming port pins. When the read/write cycle is complete, the AD6636 drives $\overline{\text{DTACK}}$ low. The $\overline{\text{DTACK}}$ signal goes high again after either the $\overline{\text{CS}}$ or $\overline{\text{DS}}$ signal is driven high. Because the $\overline{\text{DTACK}}$ pin is an open-drain output with a weak internal pull-up resistor (70 kΩ), an external pull-up resistor is recommended (see Figure 58). Figure 15 and Figure 16 are the timing diagrams for read and write cycles using the microport in MNM mode. Do not read or write, to or from, addresses beyond those defined by the memory map (Address 0xE8 to Address 0xFF). Attempting to access these addresses causes the bus to hang because $\overline{\text{DTACK}}$ does not go high to signal the end of the access.

For an asynchronous write operation on the Motorola mode microport, the CPUCLK should be running. Set up the data and address buses. Pull the R/$\overline{\text{W}}$ and $\overline{\text{DS}}$ signals low and then pull the $\overline{\text{CS}}$ signal low. The $\overline{\text{DTACK}}$ goes low after a few clock cycles to indicate that the write access is complete and that $\overline{\text{CS}}$ can be pulled high to disable the microport. For an asynchronous read operation on the Motorola mode microport, set up the address bus and three-state the data bus. Pull the $\overline{\text{RD}}$ signal low and then pull the $\overline{\text{CS}}$ signal low. The $\overline{\text{DTACK}}$ goes low after a few clock cycles to indicate that valid data is on the data bus.

Accessing Multiple AD6636 Devices

If multiple AD6636 devices are on a single board, the microport pins for these devices can be shared. In this configuration, a single programming device (DSP, FPGA, or microcontroller) can program all AD6636 devices connected to it.

Each AD6636 has four CHIPID pins that can be connected in 16 different ways. During a write/read access, the internal circuitry checks to see if the CHIPID bits in the chip I/O access control register (Address 0x02) are the same as the logic levels of the CHIPID pins (hardwired to the part). If the CHIPID bits and the CHIPID pins have the same value, then a write/read access is completed; otherwise, the access is ignored.

To program multiple devices using the same microport control and data buses, the devices should have separate CHIPID pin configurations. A write/read access can be made on the intended chip only; all other chips would ignore the access.

JTAG BOUNDARY SCAN

The AD6636 supports a subset of the IEEE Standard 1149.1 specification. For details of the standard, see the *IEEE Standard Test Access Port and Boundary-Scan Architecture*, an IEEE-1149 publication.

The AD6636 has five pins associated with the JTAG interface. These pins, listed in Table 28, are used to access the on-chip test access port. All input JTAG pins are pull-up except for TCLK, which is pull-down.

Table 28. Boundary Scan Test Pins

Mnemonic	Description
$\overline{\text{TRST}}$	Test Access Port Reset
TCLK	Test Clock
TMS	Test Access Port Mode Select
TDI	Test Data Input
TDO	Test Data Output

The AD6636 supports three op codes, listed in Table 29. These instructions set the mode of the JTAG interface.

Table 29. Boundary Scan Op Codes

Instruction	Op Code
BYPASS	11
SAMPLE/PRELOAD	01
EXTEST	00

A BSDL file for this device is available. Contact sales for more information.

EXTEST (2'b00)

Places the IC into an external boundary-test mode and selects the boundary-scan register to be connected between TDI and TDO. During this operation, the boundary-scan register is accessed to drive-test data off-chip via boundary outputs and receive test data off-chip from boundary inputs.

SAMPLE/PRELOAD (2'b01)

Allows the IC to remain in normal functional mode and selects the boundary-scan register to be connected between TDI and TDO. The boundary-scan register can be accessed by a scan operation to take a sample of the functional data entering and leaving the IC. Also, test data can be preloaded into the boundary scan register before an EXTEST instruction.

BYPASS (2'b11)

Allows the IC to remain in normal functional mode and selects a 1-bit bypass register between TDI and TDO. During this instruction, serial data is transferred from TDI to TDO without affecting operation of the IC.

MEMORY MAP

READING THE MEMORY MAP TABLE

Each row in the memory map table has four address locations. The memory map is roughly divided into four regions: global register map (Address 0x00 to Address 0x0B), input port register map (Address 0x0C to Address 0x67), channel register map (Address 0x68 to Address 0xBB), and output port register map (Address 0xBC to Address 0xE7). The channel register map is shared by all six channels and access to individual channels is given by the channel I/O access control register (Address 0x02).

In the memory map (see Table 30), the addresses are given in the right column. The column with the heading Byte 0 has the address given in the right column. The column Byte 1 has the address given by 1 more than the address listed in the right column (address offset of 1). Similarly, the address offset for the Byte 2 column is 2, and for the Byte 3 column is 3. For example, the second row lists 0x04 as the address in the right column. The pin synchronization configuration register has Address 0x04, the soft synchronization configuration register has Address 0x05, and the LVDS control register lists Address 0x07 and Address 0x06.

Bit Format

All registers are in little-endian format. For example, if a register takes 24 bits or three address locations, then the most significant byte is at the highest address location and the least significant byte is at a lowest address location. In all registers, the least significant bit is Bit 0 and the most significant bit is Bit 7. For example, the NCO frequency <31:0> register is 32 bits wide. Bit 0 (LSB) of this register is written at Bit 0 of Address 0x70 and Bit 32 (MSB) of this register is written at Bit 7 of Address 0x73.

When referring to a register that takes up multiple address locations, it is referred to by the address location of the most significant byte of the register. For example, the text reads "Port A dwell timer at Address 0x2A." Note that only the four most significant bits of this register are at this location, and this register also takes up Address 0x29 and Address 0x28.

Open Locations

All locations marked as open are currently not used. When required, these locations should be written with 0s. Writing to these locations is required only when part of an address location is open (for example, Address 0x78). If the whole address location is open (for example, Address 0x00), then this address location does not need to be written. If the open locations are readback using the microport or the serial port, the readback value is undefined (each bit can be independently 1 or 0), and these bits have no significance.

If an address location has more than one register or has one register with some open bits, then the order of these registers is as given in the table.

For example, Address 0x33: Open <7:5>, Port A Signal Monitor <4:0>. The open <7:5> is located at Bits <7:5>, and the Port A signal monitor <4:0> is located at Bits <4:0>.

Another example is Address 0x35: Open <15:10>, Port A Upper Threshold <9:0> Here, Bits <7:2> of Address 0x35 are open <15:10>. Bits <1:0> of Address 0x35 and Bits <7:0> of Address 0x34 make up the Port A upper threshold <9:0> register (Bit 1 of Address 0x35 is the MSB of the Port A upper threshold register).

Default Values

When coming out of reset, some of the address locations (but not all) are loaded with default values. When available, the default values for the registers are given in the table. If the default value is not listed, then these address locations are in an undefined state (Logic 0 or Logic 1) on $\overline{\text{RESET}}$.

Logic Levels

In the explanation of various registers, bit is set is synonymous with bit is set to Logic 1 or writing Logic 1 for the bit. Similarly clear a bit is synonymous with bit is set to Logic 0 or writing Logic 0 for the bit.

Table 30. Memory Map

8-Bit Hex Address	Byte 3	Byte 2	Byte 1	Byte 0	8-Bit Hex Address
0x03	Open <7:6>, Channel Enable <5:0>	Open<7:6>, Channel I/O Access Control<5:0>	Chip I/O Access Control <7:0> (Default 0x00)	Open<7:0>	0x00
0x07	Open <15:11>, LVDS Control<10:0> (Default 0x06FC)		Soft Synchronization Configuration<7:0>	Pin Synchronization Configuration<7:0>	0x04
0x0B	Interrupt Mask <15:0>		Interrupt Status <15:0> (Read-Only, Default 0x00)		0x08
ADC Input Port Register Map—Addresses 0x0C to 0x67					
0x0F	ADC Input Control <31:0>				0x0C
0x13	Open<15:0>		ADC CLK Control <15:0> (Default 0x0000)		0x10
0x17	Port AB, IQ Correction Control<15:0> (Default 0x0000)		Port CD, IQ Correction Control <15:0> (Default 0x0000)		0x14
0x1B	Port AB, DC Offset Correction I<15:0>		Port AB, DC Offset Correction Q<15:0>		0x18
0x1F	Port CD, DC Offset Correction I<15:0>		Port CD, DC Offset Correction Q<15:0>		0x1C
0x23	Port AB, Phase Offset Correction <15:0>		Port AB, Amplitude Offset Correction <15:0>		0x20
0x27	Port CD, Phase Offset Correction <15:0>		Port CD, Amplitude Offset Correction <15:0>		0x24
0x2B	Port A Gain Control <7:0>	Open<23:20>, Port A Dwell Timer <19:0>			0x28
0x2F	Open<7:0>	Port A Power Monitor Period <23:0>			0x2C
0x33	Open<7:5>, Port A Signal Monitor<4:0>	Port A Power Monitor Output <23:0>			0x30
0x37	Open<15:10>, Port A Lower Threshold <9:0>		Open <15:10>, Port A Upper Threshold <9:0>		0x34
0x3B	Port B Gain Control <7:0>	Open<23:20>, Port B Dwell Timer <19:0>			0x38
0x3F	Open<7:0>	Port B Power Monitor Period <23:0>			0x3C
0x43	Open<7:5>, Port B Signal Monitor<4:0>	Port B Power Monitor Output <23:0>			0x40
0x47	Open<15:10>, Port B Lower Threshold <9:0>		Open <15:10>, Port B Upper Threshold <9:0>		0x44
0x4B	Port C Gain Control <7:0>	Open<23:20>, Port C Dwell Timer <19:0>			0x48
0x4F	Open<7:0>	Port C Power Monitor Period <23:0>			0x4C
0x53	Open<7:5>, Port C Signal Monitor<4:0>	Port C Power Monitor Output <23:0>			0x50
0x57	Open<15:10>, Port C Lower Threshold <9:0>		Open <15:10>, Port C Upper Threshold <9:0>		0x54
0x5B	Port D Gain Control <7:0>	Open<23:20>, Port D Dwell Timer <19:0>			0x58
0x5F	Open<7:0>	Port D Power Monitor Period <23:0>			0x5C
0x63	Open<7:5>, Port D Signal Monitor<4:0>	Port D Power Monitor Output <23:0>			0x60
0x67	Open<15:10>, Port D Lower Threshold <9:0>		Open <15:10>, Port D Upper Threshold <9:0>		0x64
Channel Register Map—Addresses 0x68 to 0xBB					
0x6B	Open<15:0>		Open<15:9>, NCO Control<8:0>		0x68
0x6F	NCO Start Hold-Off Counter<15:0>		NCO Frequency Hold-Off Counter<15:0>		0x6C
0x73	NCO Frequency <31:0> (Default 0x0000 0000)				0x70
0x77	Open<15:0>		NCO Phase Offset<15:0> (Default 0x0000)		0x74
0x7B	Open<7:1>, CIC Bypass<0>	Open<7:5>, CIC Decimation<4:0>	Open<7:5>, CIC Scale Factor<4:0>	Open<7:4>, FIR-HB Control<3:0>	0x78
0x7F	Open<15:0>		Open<15:13>, MRCF Control<12:0>		0x7C
0x83	Open<7:6>, MRCF Coefficient 3 <5:0>	Open<7:6>, MRCF Coefficient 2 <5:0>	Open<7:6>, MRCF Coefficient 1 <5:0>	Open<7:6>, MRCF Coefficient 0 <5:0>	0x80
0x87	Open<7:6>, MRCF Coefficient 7 <5:0>	Open<7:6>, MRCF Coefficient 6 <5:0>	Open<7:6>, MRCF Coefficient 5 <5:0>	Open<7:6>, MRCF Coefficient 4 <5:0>	0x84
0x8B	Open<15:12>, DRCF Control Register<11:0>		Open <7:6>, DRCF Coefficient Offset<5:0>	Open<7>, DRCF Taps <6:0>	0x88
0x8F	Open<15:0>		Open <7:6>, DRCF Final Address<5:0>	Open <7:6>, DRCF Start Address<5:0>	0x8C
0x93	Open<15:0>		Open<15:14>, DRCF Coefficient Memory <13:0>		0x90

8-Bit Hex Address	Byte 3	Byte 2	Byte 1	Byte 0	8-Bit Hex Address
0x97	Open<15:12>, CRCF Control Register<11:0>		Open <7:6>, CRCF Coefficient Offset<5:0>	Open<7>, CRCF Taps <6:0>	0x94
0x9B	Open<15:0>		Open <7:6>, CRCF Final Address<5:0>	Open <7:6>, CRCF Start Address<5:0>	0x98
0x9F	Open<7:0>	Open<23:20>, CRCF Coefficient Memory <19:0>			0x9C
0xA3	Open<15:11>, AGC Control Register<10:0>		AGC Hold-Off Register<15:0>		0xA0
0xA7	Open<15:12>, AGC Update Decimation<11:0>		Open<15:12>, AGC Signal Gain <11:0>		0xA4
0xAB	Open<15:12>, AGC Error Threshold <11:0>		Open<7:6>, AGC Average Samples<5:0>	AGC Pole Location <7:0>	0xA8
0xAF	Open<7:0>	AGC Desired Level<7:0>	AGC Loop Gain2 <7:0>	AGC Loop Gain1 <7:0>	0xAC
0xB3	Open<7:0>	BIST I Path Signature Register<23:0> (Read-Only, Default 0xAD6636)			0xB0
0xB7	Open<7:0>	BIST Q Path Signature Register<23:0> (Read-Only, Default 0xAD6636)			0xB4
0xBB	Open <15:0>		BIST Control <15:0>		0xB8
Output Port Register Map—Addresses 0xBC to 0xE7					
0xBF	Open<7:0>	Parallel Port Output Control <23:0>			0xBC
0xC3	Open<15:0>		Open<15:10>, Output Port Control <9:0>		0xC0
0xC7	AGC0, I Output<15:0> (Read-Only)		AGC0, Q Output <15:0> (Read-Only)		0xC4
0xCB	AGC1, I Output <15:0> (Read-Only)		AGC1, Q Output <15:0> (Read-Only)		0xC8
0xCF	AGC2, I Output <15:0> (Read-Only)		AGC2, Q Output <15:0> (Read-Only)		0xCC
0xD3	AGC3, I Output <15:0> (Read-Only)		AGC3, Q Output <15:0> (Read-Only)		0xD0
0xD7	AGC4, I Output <15:0> (Read-Only)		AGC4, Q Output <15:0> (Read-Only)		0xD4
0xDB	AGC5, I Output <15:0> (Read-Only)		AGC5, Q Output <15:0> (Read-Only)		0xD8
0xDF	Open<15:12>, AGC0 RSSI Output<11:0> (Read-Only)		Open<15:12>, AGC1 RSSI Output<11:0> (Read-Only)		0xDC
0xE3	Open<15:12>, AGC2 RSSI Output<11:0> (Read-Only)		Open<15:12>, AGC3 RSSI Output<11:0> (Read-Only)		0xE0
0xE7	Open<15:12>, AGC4 RSSI Output<11:0> (Read-Only)		Open<15:12>, AGC5 RSSI Output<11:0> (Read-Only)		0xE4

GLOBAL REGISTER MAP

Chip I/O Access Control Register <7:0>

<7>: Synchronous Microport Bit. When this bit is set, the microport assumes that its controls signals (such as R/$\overline{W}$, $\overline{DS}$, and $\overline{CS}$) are synchronous to the CPUCLK. When cleared, asynchronous control signals are assumed and the microport control signals are resynchronized with CPUCLK inside the AD6636 part. Synchronous microport (when bit is set) has the advantage of requiring a fewer number of clock cycles for read/write access.

<6>: This bit is open.

<5:2>: Chip ID Bits. The chip ID bits are used to compare against the chip ID input pins, enabling or disabling I/O access for this specific chip. When more than one AD6636 part is sharing the microport, different CHIPID pins can be used to differentiate among the parts. A particular part gives I/O access only when the CHIPID pins have the same value as these chip ID bits.

<1>: This bit is open.

<0>: Byte Mode Bit. The byte mode bit selects the bit width for the microport operation. Table 31 shows details.

Table 31. Microport Data Bus Width Selection

Chip Access Control Register <0>	Microport Data Bus Bit Width
0 (default)	8-bit mode, using D<7:0>
1	16-bit mode, using D<15:0>

Channel I/O Access Control Register <5:0>

These bits enable/disable the channel I/O access capability.

<5>: Channel 5 Access Bit. When the Channel 5 access bit is set to Logic 1, any I/O write operation (from either the microport or the serial port) that addresses a register located within the channel register map updates the Channel 5 registers. Similarly, for a read operation, the contents of the desired address in the channel register map are output when this bit is set to Logic 1.

<4>: Channel 4 Access Bit. Similar to Bit <5> for Channel 4.

<3>: Channel 3 Access Bit. Similar to Bit <5> for Channel 3.

<2>: Channel 2 Access Bit. Similar to Bit <5> for Channel 2.

<1>: Channel 1 Access Bit. Similar to Bit <5> for Channel 1.

<0>: Channel 0 Access Bit. Similar to Bit <5> for Channel 0.

Note that if the access bits are set for more than one channel during write access, all channels with access are written with the same data. This is especially useful when more than one channel has similar configurations. During a read operation, if more than one channel has access, the read access is given to the channel with the lowest channel number. For example, if both Channel 4 and Channel 2 have access bits set, then read access is given to Channel 2.

Channel Enable Register <5:0>

<5>: Channel 5 Enable Bit. When this bit is set, Channel 5 logic is enabled. When this bit is cleared, Channel 5 is disabled and the channel's logic does not consume any power. On power-up, this bit comes up with Logic 0 and the channel is disabled. A start sync does not start Channel 5 unless this bit is set before issuing the start sync.

<4>: Channel 4 Enable Bit. Similar to Bit <5> for Channel 4.

<3>: Channel 3 Enable Bit. Similar to Bit <5> for Channel 3.

<2>: Channel 2 Enable Bit. Similar to Bit <5> for Channel 2.

<1>: Channel 1 Enable Bit. Similar to Bit <5> for Channel 1.

<0>: Channel 0 Enable Bit. Similar to Bit <5> for Channel 0.

Pin Synchronization Configuration <7:0>

<7>: Hop Synchronization Enable Bit. This bit is a global enable for any hop synchronization involving SYNC pins. When this bit is set, hop synchronization is enabled for all channels that are programmed for pin synchronization. When this bit is cleared, hop synchronization is not performed for any channel that is programmed for pin synchronization.

<6>: Start Synchronization Enable Bit. This bit is a global enable for any start synchronization involving SYNC pins. When this bit is set, start synchronization is enabled for all channels that are programmed for pin synchronization. When this bit is cleared, start synchronization is not performed for any channel that is programmed for pin synchronization.

<5>: First Sync Only Bit. When this bit is set, the NCO synchronization logic only recognizes the first synchronization event as valid. All other requests for synchronization events are ignored as long as this bit is set. When cleared, all synchronization events are acted upon.

<4>: Edge-Sensitivity Bit. When this bit is set, the rising edge on the SYNC pin(s) is detected as a synchronization event (edge-sensitive detection). When cleared, Logic 1 on the SYNC pin(s) is detected as a synchronization event (level-sensitive detection).

<3>: Enable Synchronization from SYNC3 Bit. When this bit is set, the SYNC3 pin can be used for synchronization. When this bit is cleared, the SYNC3 pin is ignored. This is a global enable for all SYNC pins, and each individual channel selects which pin it listens to.

<2>: Enable Synchronization from SYNC2 Bit. Similar to Bit <3> for the SYNC[2] pin.

<1>: Enable Synchronization from SYNC1 Bit. Similar to Bit <3> for the SYNC1 pin.

<0>: Enable Synchronization from SYNC0 bit. Similar to Bit <3> for the SYNC0 pin.

Soft Synchronization Configuration <7:0>

<7>: Soft Hop Synchronization Enable Bit. When this bit is set, hop synchronization is enabled for all channels selected using Bits 5:0. When this bit is cleared, hop synchronization is not performed for any channels selected using Bits 5:0.

<6>: Soft Start Synchronization Enable Bit. When this bit is set, start synchronization is enabled for all channels selected using Bits 5:0. When this bit is cleared, start synchronization is not performed for any channels selected using Bits 5:0.

Bits<5:0> form the SOFT_SYNC control bits. These bits can be written to by the controller to initiate the synchronization of a selected channel.

<5>: Soft Sync Channel 5 Enable Bit. When this bit is set, it enables Channel 5 to receive a hop sync or start sync, as defined by Bit 7 and Bit 6, respectively. When cleared, Channel 5 does not receive any soft sync.

<4>: Soft Sync Channel 4 Enable Bit. Similar to Bit <5> for Channel 4.

<3>: Soft Sync Channel 3 Enable Bit. Similar to Bit <5> for Channel 3.

<2>: Soft Sync Channel 2 Enable Bit. Similar to Bit <5> for Channel 2.

<1>: Soft Sync Channel 1 Enable Bit. Similar to Bit <5> for Channel 1.

<0>: Soft Sync Channel 0 Enable Bit. Similar to Bit <5> for Channel 0.

LVDS Control Register <10:0>

<10>: CMOS Mode Bit. When this bit is set, the ADC ports operate in CMOS mode. When this bit is cleared, the ADC ports operate in LVDS mode. The default is Logic 1 or CMOS mode. In LVDS mode, two CMOS ADC port pins are used to form one differential pair of LVDS ADC ports.

<9>: Reserved. This bit should always be written as Logic 1.

<8>: Autocalibrate Enable Bit. When this bit is set, the auto-calibration cycle is invoked for the LVDS pads. At the end of calibration, this calibration value is set for the LVDS pads.

When this bit is cleared, the output for the LVDS controller is taken from manual calibration value (Bits <7:0> of this register).

<7:4>: These bits are open.

<3:0>: Manual Calibration Value Bits. The value of these bits is used for manual LVDS calibration. When the autocalibrate bit is set, these bits are don't care.

Interrupt Status Register <15:0>

This register is read-only.

<15>: AGC5 RSSI Update Interrupt Bit. If the AGC5 update interrupt enable bit is set, this bit is set by the AD6636 whenever AGC5 updates a new RSSI word (the new word should be different from the previous word). If the AGC5 update interrupt enable bit is cleared, then this bit is not set (not updated). An interrupt is not generated in this case.

Note: For Bits <15:10>, no interrupt is generated, if the new RSSI word is the same as the previous RSSI word.

<14>: AGC4 RSSI Update Interrupt Bit. Similar to Bit <15> for the AGC4.

<13>: AGC3 RSSI Update Interrupt Bit. Similar to Bit <15> for the AGC3.

<12>: AGC2 RSSI Update Interrupt Bit. Similar to Bit <15> for the AGC2.

<11>: AGC1 RSSI Update Interrupt Bit. Similar to Bit <15> for the AGC1.

<10>: AGC0 RSSI Update Interrupt Bit. Similar to Bit <15> for the AGC0.

<9>: Channel 5 Data Ready Interrupt Bit. This bit is set to Logic 1 whenever the channel BIST signature registers are loaded with data. The conditions required for setting this bit are: the channel BIST signature registers is programmed for BIST signature generation and the Channel 5 data ready enable bit in the interrupt enable register is cleared. If the Channel 5 data ready enable bit in the interrupt enable register is set, the AD6636 does not set this bit on signature generation and an interrupt is not generated.

<8>: Channel 4 Data Ready Interrupt Bit. Similar to Bit <9> for Channel 4.

<7>: Channel 3 Data Ready Interrupt Bit. Similar to Bit <9> for Channel 3.

<6>: Channel 2 Data Ready Interrupt Bit. Similar to Bit <9> for Channel 2.

<5>: Channel 1 Data Ready Interrupt Bit. Similar to Bit <9> for Channel 1.

<4>: Channel 0 Data Ready Interrupt Bit. Similar to Bit <9> for Channel 0.

<3>: ADC Port D Power Monitoring Interrupt Bit. This bit is set by the AD6636 whenever the ADC Port D power monitor interrupt enable bit is set and the Port D power monitor timer runs out (end of the Port D power monitor period). If the ADC Port D power monitoring interrupt enable bit is cleared, the AD6636 does not set this bit and does not generate an interrupt.

Note: In real input CMOS mode, all four input ports exist. In complex input CMOS mode, only ADC Ports A and C function. In real input LVDS mode, only ADC Ports A and C function.

<2>: ADC Port C Power Monitoring Interrupt Bit. Similar to Bit <3> for ADC Port C.

<1>: ADC Port B Power Monitoring Interrupt Bit. Similar to Bit <3> for ADC Port B.

<0>: ADC Port A Power Monitoring Interrupt Bit. Similar to Bit <3> for ADC Port A.

Interrupt Enable Register <15:0>

<15>: AGC5 RSSI Update Enable Bit. When this bit is set, the AGC5 RSSI update interrupt is enabled, allowing an interrupt to be generated when the RSSI word is updated. When this bit is cleared, an interrupt cannot be generated for this event. Also, see the Interrupt Status Register <15:0> section.

<14>: AGC4 RSSI Update Enable Bit. Similar to Bit <15> for the AGC4.

<13>: AGC3 RSSI Update Enable Bit. Similar to Bit <15> for the AGC3.

<12>: AGC2 RSSI Update Enable Bit. Similar to Bit <15> for the AGC2.

<11>: AGC1 RSSI Update Enable Bit. Similar to Bit <15> for the AGC1.

<10>: AGC0 RSSI Update Enable Bit. Similar to Bit <15> for the AGC0.

<9>: Channel 5 Data Ready Enable Bit. When this bit is set, the Channel 5 data ready interrupt is enabled, allowing an interrupt to be generated when Channel 5 BIST signature registers are updated. When this bit is cleared, an interrupt cannot be generated for this event.

<8>: Channel 4 Data Ready Enable Bit. Similar to Bit <9> for Channel 4.

<7>: Channel 3 Data Ready Enable Bit. Similar to Bit <9> for Channel 3.

<6>: Channel 2 Data Ready Enable Bit. Similar to Bit <9> for Channel 2.

<5>: Channel 1 Data Ready Enable Bit. Similar to Bit <9> for Channel 1.

<4>: Channel 0 Data Ready Enable Bit. Similar to Bit <9> for Channel 0.

<3>: ADC Port D Power Monitoring Enable Bit. When this bit is set to Logic 1, the ADC Port D power monitoring interrupt is enabled allowing an interrupt to be generated when ADC Port D power monitoring registers are updated. When set to Logic 1, the ADC Port D power monitoring interrupt is disabled.

<2>: ADC Port C Power Monitoring Enable Bit. Similar to Bit <3> for ADC Port C.

<1>: ADC Port B Power Monitoring Enable Bit. Similar to Bit <3> for ADC Port B.

<0>: ADC Port A Power Monitoring Enable Bit. Similar to Bit <3> for ADC Port A.

INPUT PORT REGISTER MAP

ADC Input Control Register <27:0>

These bits are general control bits for the ADC input logic.

<27>: PN Active Bit. When this bit is set, the pseudorandom number generator is active. When this bit is cleared, the PN generator is disabled and the seed is set to its default value.

<26>: EXP Lock Bit. When this bit is set along with the PN active bit, then the EXP signal for pseudorandom input is locked to 000 (giving full-scale input). When this bit is cleared, EXP bits for pseudorandom input are randomly generated input data bits.

<25>: Port C Complex Data Active Bit. When this bit is set, the data inputs on Port C and Port D are interpreted as complex inputs (Port C for the in-phase signal and Port D for the quadrature phase signal). This complex input is passed on as the input from ADC Port C. When this bit is cleared, the data on ADC Port C and ADC Port D are interpreted as real and independent input.

Note that complex input mode is available only in CMOS input mode.

<24>: Port A Complex Data Active Bit. When this bit is set, the data input on Port A and Port B are interpreted as complex input (Port A for the in-phase signal and Port B for the quadrature phase signal). This complex input is passed on as input from ADC Port A. When this bit is cleared, the data on ADC Port A and ADC Port B are interpreted as real and independent input.

Note that complex input mode is available only in CMOS input mode.

<23>: Channel 5 Complex Data Input Bit. When this bit is set, Channel 5 gets complex input data from the source that is selected by the crossbar mux select bits. When this bit is cleared, Channel 5 receives real input data (see Table 32).

<22:20>: Channel 5 Crossbar Mux Select Bits. These bits select the source of input data for Channel 5 (see Table 32).

Table 32. Channel 5 Input Configuration

Complex Data Input Bit	Crossbar Mux Select Bits	Configuration
0	000	ADC Port A Drives Input (Real)
0	001	ADC Port B Drives Input (Real)
0	010	ADC Port C Drives Input (Real)
0	011	ADC Port D Drives Input (Real)
0	100	PN Sequence Drives Input (Real)
1	000	Ports A and B Drive Complex Input
1	001	Ports C and D Drive Complex Input
1	010	PN Sequence Drives Complex Input

<19>: Channel 4 Complex Data Input Bit. Similar to Bit <23> for Channel 4.

<18:16>: Channel 4 Crossbar Mux Select Bits. Similar to Bits <22:20> for Channel 4.

<15>: Channel 3 Complex Data Input Bit. Similar to Bit <23> for Channel 3.

<14:12>: Channel 3 Crossbar Mux Select Bits. Similar to Bits <22:20> for Channel 3.

<11>: Channel 2 Complex Data Input Bit. Similar to Bit <23> for Channel 2.

<10:8>: Channel 2 Crossbar Mux Select Bits. Similar to Bits <22:20> for Channel 2.

<7>: Channel 1 Complex Data Input Bit. Similar to Bit <23> for Channel 1.

<6:4>: Channel 1 Crossbar Mux Select Bits. Similar to Bits <22:20> for Channel 1.

<3>: Channel 0 Complex Data Input Bit. Similar to Bit <23> for Channel 0.

<2:0>: Channel 0 Crossbar Mux Select Bits. Similar to Bits <22:20> for Channel 0.

ADC CLK Control Register <11:0>

These bits control the ADC clocks and internal PLL clock.

<11>: ADC Port D CLK Invert Bit. When this bit is set, the inverted ADC Port D clock is used to register ADC Input Port D data into the part. When this bit is cleared, the clock is used as is, without any inversion or phase change.

<10>: ADC Port C CLK Invert Bit. Similar to Bit <11> for ADC Port C.

<9>: ADC Port B CLK Invert Bit. Similar to Bit <11> for ADC Port B.

<8>: ADC Port A CLK Invert Bit. Similar to Bit <11> for ADC Port A.

<7:6>: ADC Pre-PLL Clock Divider Bits. These bits control the PLL clock divider. The PLL clock is derived from the ADC Port A clock.

Table 33. PLL Clock Divider Select Bits

PLL Clock Divider Bits <12:11>	Divide-by Value
00	Divide-by-1, Bypass
01	Divide-by-2
10	Divide-by-4
11	Divide-by-8

<5:1>: PLL Clock Multiplier Bits. These bits control the PLL clock multiplier. The output of the PLL clock divider is multiplied with the binary value of these bits. The valid range for the multiplier is from 4 to 20. A value outside this range powers down the PLL, and the PLL clock is the same as the ADC Port A clock.

<0>: This bit is open (write Logic 0).

Port AB, I/Q Correction Control <15:0>

<15:12>: Amplitude Loop BW. These bits set the decimation value used in the integrator for the amplitude offset-estimation feedback loop. A value of 0 sets a decimation of 2^{12} and a value of 11 sets decimation of 2^{24}. Each increment of these bits increases the decimation value by a power of 2.

<11:8>: Phase Loop BW. These bits set the decimation value used in the integrator for the phase offset-estimation feedback loop. A value of 0 sets a decimation of 2^{12} and a value of 11 sets decimation of 2^{24}. Each increment of these bits increases the decimation value by a power of 2.

<7:4>: DC Loop BW. These bits set the decimation and interpolation value used in the low-pass filters for the dc offset estimation feedback loop. A value of 0 sets a decimation/interpolation of 2^{12} and a value of 11 sets decimation/interpolation of 2^{24}. Each increment of these bits increases the decimation/interpolation value by a power of 2.

<3>: Reserved.

<2>: Port AB Amplitude Correction Enable Bit. When the amplitude correction enable bit is set, the amplitude correction function of the I/Q correction logic for the AB port is enabled. When this bit cleared, the amplitude correction value is given by the value of the AB amplitude correction register. If the Port A complex data active bit of the ADC input control register is cleared (real input mode), this bit is a don't care.

<1>: Port AB Phase Correction Enable Bit. When this bit is set, the phase correction function of the I/Q correction logic for the AB port is enabled. When this bit is cleared, the phase correction value is given by the value of the AB phase correction register. If the Port A complex data active bit of the ADC input control register is cleared (real input mode), this bit is a don't care.

<0>: Port AB DC Correction Enable Bit. When this bit is set, the dc offset correction function of the I/Q correction block for the AB port is enabled. When this bit is cleared, the dc offset correction value is given by the value of the AB offset correction registers. If the Port A complex data active bit of the ADC input control register is cleared (real input mode), this bit is a don't care.

Port CD, I/Q Correction Control <15:0>

<15:12>: Amplitude Loop BW. These bits set the decimation value used in the integrator for the amplitude offset estimation feedback loop. A value of 0 sets a decimation of 2^{12} and a value of 11 sets decimation of 2^{24}. Each increment of these bits increases the decimation value by a power of 2.

<11:8>: Phase Loop BW. These bits set the decimation value used in the integrator for the phase offset estimation feedback loop. A value of 0 sets a decimation of 2^{12} and a value of 11 sets decimation of 2^{24}. Each increment of these bits increases the decimation value by a power of 2.

<7:4>: DC Loop BW. These bits set the decimation and interpolation value used in the low-pass filters for the dc offset estimation feedback loop. A value of 0 sets a decimation/interpolation of 2^{12} and a value of 11 sets decimation/interpolation of 2^{24}. Each increment of these bits increases the decimation/interpolation value by a power of 2.

<3>: Reserved.

<2>: Port CD Amplitude Correction Enable Bit. When this bit is set, the amplitude correction function of the I/Q correction logic for the AB port is enabled. When this bit is cleared, the amplitude correction value is given by the value of the AB amplitude correction register. If the Port A complex data active bit of the ADC input control register is cleared (real input mode), this bit is a don't care.

<1>: Port CD Phase Correction Enable Bit. When this bit is set, the phase correction function of the I/Q correction logic for the AB port is enabled. When this bit is cleared, the phase correction value is given by the value of the AB phase correction register. If the Port A complex data active bit of the ADC input control register is cleared (real input mode), this bit is a don't care.

<0>: Port CD DC Correction Enable Bit. When the dc correction enable bit is set, the dc offset correction function of the I/Q correction block for the AB port is enabled. When cleared, the dc offset correction value is given by the value of the AB offset correction registers. If the Port A complex data active bit of the ADC input control register is cleared (real input mode), this bit is a don't care.

Port AB, DC Offset Correction I <15:0>

This register holds the in-phase signal dc offset correction value for complex data stream when dc correction is enabled. This value should be set manually when automatic correction is disabled. This 16-bit value is subtracted from the 16-bit ADC Port A data (in-phase signal). This data is a don't care in real input mode.

Port AB, DC Offset Correction Q <15:0>

This register holds the quadrature phase signal dc offset correction value for complex data stream when dc correction is enabled. This value should be set manually when automatic correction is disabled. This 16-bit value is subtracted from the 16-bit ADC Port B data (quadrature phase signal). This data is a don't care in real input mode.

Port CD, DC Offset Correction I <15:0>

This register holds the in-phase signal dc offset correction value for complex data stream when dc correction is enabled. This value should be set manually when automatic correction is disabled. This 16-bit value is subtracted from the 16-bit ADC Port C data (in-phase signal). This data is a don't care in real input mode.

Port CD, DC Offset Correction Q <15:0>

This register holds the quadrature phase signal dc offset correction value for complex data stream when dc correction is enabled. This value should be set manually when automatic correction is disabled. This 16-bit value is subtracted from the 16-bit ADC Port D data (quadrature phase signal). This data is a don't care in real input mode.

Port AB, Phase Offset Correction <15:0>

This register holds the phase offset correction value for complex data stream when the AB port phase correction is enabled. This value is set manually when automatic correction is disabled. This value is calculated as tan(phase_mismatch), where phase_mismatch is the mismatch in phase between I (in-phase signal) and Q (quadrature phase signal). This 14-bit value is multiplied with 16-bit Q (quadrature phase signal, Input Port B) and added to 16-bit I (in-phase signal, Input Port A). This data is a don't care in real input mode.

Port AB, Amplitude Offset Correction <15:0>

This register holds the amplitude offset correction value for complex data stream when the AB port amplitude correction is enabled. This value is set manually when automatic correction is disabled. This value is calculated as (Mag(Q) − Mag(I)), where I is the in-phase signal and Q is the quadrature phase signal. This 14-bit value is multiplied with 16-bit Q (quadrature phase signal, Input Port B) and added to 16-bit Q (quadrature phase signal, Input Port B). This data is a don't care in real input mode.

Port CD, Phase Offset Correction <15:0>

This register holds the phase offset correction value for the complex data stream when CD port phase correction is enabled. This value should be set manually when automatic correction is disabled. This value should be calculated as tangent (phase_mismatch), where phase_mismatch is the mismatch in phase between I (in-phase signal) and Q (quadrature phase signal). This 14-bit value is multiplied with 16-bit Q (quadrature phase signal, Input Port D) and added to 16-bit I (in-phase signal, Input Port C). This data is a don't care in real input mode.

Port CD, Amplitude Offset Correction <15:0>

This register holds the amplitude offset correction value for complex data stream when CD port amplitude correction is enabled. This value is set manually when automatic correction is disabled. This value is calculated as (Mag(Q) − Mag(I)), where I is the in-phase signal and Q is the quadrature phase signal. This 14-bit value is multiplied with 16-bit Q (quadrature phase signal, Input Port D) and added to 16-bit Q (quadrature phase signal, Input Port D). This data is a don't care in real input mode.

Port A Gain Control <7:0>

<7>: This bit is open.

<6:1>: This 6-bit word specifies the relinearization pipe delay to be used in the ADC input gain control block. The decimal representation of these bits is the number of input clock cycle pipeline delays between the external EXP data output and the internal application of relinearization based on EXP.

<0>: Gain Control Enable Bit. This bit controls the configuration of the EXP<2:0> bits for Channel A. When the gain control enable bit is Logic 1, the EXP<2:0> bits are configured as outputs. When this bit is cleared, the EXP<2:0> bits are inputs.

Port A Dwell Timer <19:0>

This register is used to set the dwell time for the gain control block. When gain control block is active and detects a decrease in the signal level below the lower threshold value (programmable), a dwell time counter is initiated to provide temporal hysteresis. Doing so prevents the gain from being switched continuously. Note that the dwell timer is turned on after a drop below the lower threshold is detected in the signal level only.

Port A Power Monitor Period <23:0>

This register is used in the power monitoring logic to set the period of time for which ADC input data is monitored. This value represents the monitor period in number of ADC port clock cycles.

Port A Power Monitor Output <23:0>

This register is read-only and contains the current status of the power monitoring logic output. The output is dependent on the power monitoring mode selected. When the power monitor block is enabled, this register is updated at the end of each power monitor period. This register is updated even if an interrupt signal is not generated.

Port A Upper Threshold <9:0>

This register serves the dual purpose of specifying the upper threshold value in the gain control block and in the power monitoring block, depending on which block is active. Any ADC port input data having a magnitude greater than this value triggers a gain change in the gain control block. Any ADC port input data having a magnitude greater than this value is monitored in the power monitoring block (in peak detect or threshold crossing mode). The value of the register is compared with the absolute magnitude of the input port data. For real input, the absolute magnitude is the same as the input data; for positive and negative data, the absolute magnitude is the value of the data after removing the negative sign.

Port A Lower Threshold <9:0>

This register is used in the gain control block and represents the magnitude of the lower threshold for ADC port input data. Any ADC input data having a magnitude below the lower threshold initiates the dwell time counter. The value of the register is compared with the absolute magnitude of the input port data.

For real input, the absolute magnitude is the same as the input data; for positive and negative data, the absolute magnitude is the value of the data after removing the negative sign.

Port A Signal Monitor <4:0>

This register controls the functions of the power monitoring block.

<4>: Disable Power Monitor Period Timer Bit. When this bit is set, the power monitor period timer no longer controls the update of the power monitor holding register. A user read to the power monitor holding register updates this register. When this bit is cleared, the power monitor period register controls the timer and, therefore, controls the update rate of the power monitor holding register.

<3>: Clear-on-Read Bit. When this bit is set, the power monitor holding register is cleared every time this register is read. This bit controls whether the power monitoring function is cleared after a read of the power monitor period register. If this bit is set, the monitoring function is cleared after the read. If this bit is Logic 0, the monitoring function is not cleared. This bit is a don't care if the disable integration counter bit is clear.

<2:1>: Monitor Function Select Bits. Table 34 lists the functions of these bits.

Table 34. Monitor Function Select Bits

Monitor Function Select	Function Enabled
00	Peak Detect Mode
01	Mean Power Monitor Mode
10	Threshold Crossing Mode
11	Invalid Selection

<0>: Monitor Enable Bit. When this bit is set, the power monitoring function is enabled and operates as selected by Bits <2:1> of the signal monitor register. When this bit is cleared, the power monitoring function is disabled and the signal monitor register <2:1> bits are don't care. This bit defaults to 0 on power-up.

Note: Gain control, dwell timer, power monitor period, signal monitor, power monitoring output, and lower threshold and upper threshold registers for Port B, Port C, and Port D work similarly to the corresponding registers definitions for Port A.

CHANNEL REGISTER MAP

Channel control registers are common to all six channels and access to specific channels is determined by the channel I/O access register (Address 0x02).

NCO Control <8:0>

These bits control the NCO operation.

<8:7>: NCO Sync Start Select Bits. These bits determine which SYNC input pin is used by this channel for a start synchronization operation. Table 35 describes the selection.

Table 35. Sync Start Select Bits

NCO Control <8:7>	SYNC Pin Used for Start Synchronization
00	SYNC0
01	SYNC1
10	SYNC2
11	SYNC3

<6:5>: NCO Sync Hop Select Bits. These bits determine which SYNC input pin is used by this channel for a hop synchronization operation. Table 36 describes the selection.

Table 36. Sync Hop Select Bits

NCO Control <6:5>	SYNC Pin Used for Hop Synchronization
00	SYNC0
01	SYNC1
10	SYNC2
11	SYNC3

<4>: This bit is open.

<3>: NCO Bypass Bit. When this bit is set, the NCO is bypassed and shuts down for power savings. When a NCO frequency of dc or 0 Hz is required, this bit can be used for power savings. When this bit is cleared, the NCO operates as programmed.

<2>: Clear NCO Accumulator Bit. When this bit is set, the clear NCO accumulator bit synchronously clears the phase accumulator on all frequency hops in this channel. When this bit is cleared, the accumulator is not cleared and phase continuous hops are implemented.

<1>: Phase Dither Enable Bit. When this bit is set, phase dithering in the NCO is enabled. When this bit is cleared, phase dithering is disabled.

<0>: Amplitude Dither Enable Bit. When this bit is set, amplitude dithering in the NCO is enabled. When this bit is cleared, amplitude dithering is disabled.

Channel Start Hold-Off Counter <15:0>

When a start synchronization (software or hardware) occurs on the channel, the value in this register is loaded into a down-counter. When the counter has finished counting down to 0, the channel operation is started.

NCO Frequency Hop Hold-Off Counter <15:0>

When a hop sync occurs, a counter is loaded with the NCO frequency hold-off register value. The 16-bit counter starts counting down. When it reaches 0, the new frequency value in the shadow register is written to the NCO frequency register (see the Numerically Controlled Oscillator (NCO) section).

NCO Frequency <31:0>

The value in this register is used to program the NCO tuning frequency. The value to be programmed is given by

$$NCO\ Frequency\ Register = \frac{NCO_FREQUENCY}{CLK} \times 2^{32}$$

where:

NCO_FREQUENCY is the desired NCO tuning frequency.

CLK is the ADC clock rate.

The value given by the equation should be loaded into the register in binary format.

NCO Phase Offset <15:0>

The value in the register is loaded into the phase accumulator of the NCO block every time a start sync or hop sync is received by the channel. This allows individual channels to be started with a known nonzero phase. If Bit <2> of the NCO control register (clear phase accumulator on hop) is cleared, the NCO phase offset is not loaded on a hop sync,. This NCO offset register value is interpreted as a 16-bit unsigned integer. A 0x0000 in this register corresponds to a 0 radian offset, and a 0xFFFF corresponds to an offset of $2\pi\ (1 - 1/(2^{16}))$ radians.

CIC Bypass <0>

When this bit is set, the entire CIC filter is bypassed. The output of CIC filter is driven straight from the input without any change. When this bit is cleared, the CIC filter operates in normal mode as programmed. Writing Logic 1 to this bit disables both the CIC decimation operation and the CIC scaling operation.

CIC Decimation <4:0>

This 5-bit word specifies the CIC filter decimation value minus 1. A value of 0x00 is a decimation of 1 (bypass), and 0x1F is a decimation of 32. Writing a value of 0 in this register bypasses CIC filtering but does not bypass the CIC scaling operation.

CIC Scale Factor <4:0>

This 5-bit word specifies the CIC filter scale factor used to compensate for the gain provided by the CIC filter. The recommended value is given by

$$CIC\ Scale\ Register = ceil(5 \times \log_2 (M_{CIC})) - 5$$

where:

M_{CIC} is the decimation rate of the CIC (one more than the value in the CIC decimation register).

The *ceil* operation gives the closest integer greater than or equal to the argument.

The valid range for this register is decimal 0 to 20.

FIR-HB Control <3:0>

<3>: FIR1 Enable Bit. When this bit is set, the FIR1 fixed-coefficient filter is enabled. When cleared, FIR1 is bypassed.

<2>: HB1 Enable Bit. When this bit is set, the HB1 half-band filter is enabled. When cleared, HB1 is bypassed.

<1>: FIR2 Enable Bit. When this bit is set, the FIR2 fixed-coefficient filter is enabled. When cleared, FIR2 is bypassed.

<0>: HB2 Enable bit. When this bit is set, the HB2 half-band filter is enabled. When cleared, HB2 is bypassed.

MRCF Control Register <12:0>

<12:10>: MRCF Data Select Bits. These bits are used to select the input source for the MRCF filter. Each MRCF filter can be driven by output from the HB2 filter of any channel independently. Table 37 shows the available selections.

Table 37. MRCF Data Select Bits

MRCF Data Select<2:0>	MRCF Input Source
000	MRCF input taken from Channel 0
001	MRCF input taken from Channel 1
010	MRCF input taken from Channel 2
011	MRCF input taken from Channel 3
1x0	MRCF input taken from Channel 4
1x1	MRCF input taken from Channel 5

<9>: Interpolating Half-Band Enable Bit. When this bit is set, the interpolating half-band filter, driven by the output of the CRCF block, is enabled. When cleared, the interpolating half-band filter is bypassed and its output is the same as its input. The interpolating half-band filter doubles the data rate.

<8>: This bit is open.

<7>: Half-Rate Bit. When this bit is set, the MRCF filter operates using half the PLL clock rate. This is used for power savings when there is sufficient time to complete MRCF filtering using only half the PLL clock rate. When this bit is cleared, the MRCF filter operates at the full PLL clock rate. (See the MonoRate RAM Coefficient Filter section.)

<6:4>: MRCF Number of Taps Bits. This 3-bit word should be written with one less than the number of taps that are calculated by the MRCF filter. The filter length is given by the decimal value of this register plus 1. A value of 0 represents a 1-tap filter and maximum value of 7 represents an 8-tap filter.

<3:2>: MRCF Scale Factor Bits. The output of the MRCF filter is scaled according to the value of these bits. Table 38 describes the attenuation corresponding to each setting.

Table 38. MRCF Scale Factor

MRCF Scale<1:0>	Scale Factor
00	18.06 dB attenuation (left-shift 3 bits)
01	12.04 dB attenuation (left-shift 2 bits)
10	6.02 dB attenuation (left-shift 1 bit)
11	No scaling (0 dB)

<1>: This bit is open.

<0>: MRCF Bypass Bit. When this bit is set, the MRCF filter is bypassed and, therefore, the output of the MRCF is the same as its input. When this bit is cleared, the MRCF has normal operation as programmed by its control register.

MRCF Coefficient Memory

The MRCF coefficient memory consists of eight coefficients, each six bits wide. The memory extends from Address 0x80 to Address 0x87. The coefficients should be written in twos complement format.

DRCF Control Register <11:0>

<11>: DRCF Bypass Bit. When this bit is set, the DRCF filter is bypassed and, therefore, its output is the same as its input. When this bit is cleared, the DRCF has normal operation as programmed by the rest of this control register.

<10>: Symmetry Bit. When this bit is set, it indicates that the DRCF is implementing a symmetrical filter and only half the impulse response needs to be written into the DRCF coefficient RAM. When this bit is cleared, the filter is asymmetrical and complete impulse response of the filter should be written to the coefficient RAM. When this filter is symmetrical, it can implement up to 128 filter taps.

<9:8>: DRCF Multiply Accumulate Scale Bits. The output of the DRCF filter is scaled according to the value of these bits. Table 39 lists the attenuation corresponding to each setting.

Table 39. DRCF Multiply Accumulate Scale Bits

DRCF Scale<1:0>	Scale Factor
00	18.06 dB attenuation (left-shift 3 bits)
01	12.04 dB attenuation (left-shift 2 bits)
10	6.02 dB attenuation (left-shift 1 bit)
11	No scaling (0 dB)

<7:4>: DRCF Decimation Rate. This 4-bit word should be written with one less than the decimation rate of the DRCF filter. A value of 0 represents a decimation rate of 1 (no rate change), and the maximum value of 15 represents a decimation of 16. Filtering can be implemented irrespective of the decimation rate.

<3:0>: DRCF Decimation Phase Bits. This 4-bit word represents the decimation phase used by the DRCF filter. The valid range is 0 up to $M_{DRCF} - 1$, where M_{DRCF} is the decimation rate of the DRCF filter. This word is primarily used for synchronization of multiple channels of the AD6636, when more than one channel is used for filtering one signal (one carrier).

DRCF Coefficient Offset <7:0>

This register is used to specify which section of the 64-word coefficient memory is used for a filter. It can be used to select between multiple filters that are loaded into memory and referenced by this pointer. This register is shadowed, and the filter pointer is updated every time a new filter is started. This allows the coefficient offset to be written even while a filter is being computed without disturbing operation. The next sample comes out of the DRCF with the new filter.

DRCF Taps <6:0>

This register is written with one less than the number of taps that are calculated by the DRCF filter. The filter length is given by the decimal value of this register plus 1. A value of 0 represents a 1-tap filter, and a value of 0x28 (40 decimal) represents a 41-tap filter.

DRCF Start Address <5:0>

This register is written with the starting address of the DRCF coefficient memory to be updated.

DRCF Final Address <5:0>

This register is written with the ending address of the DRCF coefficient memory to be updated.

DRCF Coefficient Memory <13:0>

This memory consists of 64 words, and each word is 14 bits wide. The data written to this memory space is expected to be 14-bit, twos complement format. See the Decimating RAM Coefficient Filter (DRCF) section for the method to program the coefficients into the coefficient memory.

CRCF Control Register <11:0>

<11>: CRCF Bypass Bit. When this bit is set, the DRCF filter is bypassed and, therefore, its output is the same as its input. When this bit is cleared, the CRCF has normal operation as programmed by its control register.

<10>: Symmetry Bit. When this bit is set, it indicates that the CRCF is implementing a symmetrical filter and only half the impulse response needs to be written into the CRCF coefficient RAM. When this bit is cleared, the filter is asymmetrical and the complete impulse response of the filter should be written into the coefficient RAM. When this filter is symmetrical, it can implement up to 128 filter taps.

<9:8>: CRCF Multiply Accumulate Scale Bits. The output of the CRCF filter is scaled according to the value of these bits. Table 40 lists the attenuation corresponding to each setting.

Table 40. CRCF Multiply Accumulate Scale Bits

CRCF Scale<1:0>	Scale Factor
00	18.06 dB attenuation (left-shift 3 bits)
01	12.04 dB attenuation (left-shift 2 bits)
10	6.02 dB attenuation (left-shift 1 bit)
11	No scaling (0 dB)

<7:4>: CRCF Decimation Rate. This 4-bit word should be written with one less than the decimation rate of the CRCF filter. A value of 0 represents a decimation rate of 1 (no rate change) and the maximum value of 15 represents a decimation of 16. Filtering operation is done irrespective of the decimation rate.

<3:0>: CRCF Decimation Phase. This 4-bit word represents the decimation phase used by the CRCF filter. The valid range is 0 to $M_{CRCF} - 1$, where M_{CRCF} is the decimation rate of the CRCF filter. This word is primarily used for synchronization of multiple channels of the AD6636, when more than one channel is used for filtering one signal (one carrier).

CRCF Coefficient Offset <5:0>

This register is used to specify which section of the 64-word coefficient memory is used for a filter. It can be used to select between multiple filters that are loaded into memory and referenced by this pointer. This register is shadowed, and the filter pointer is updated every time a new filter is started. This allows the coefficient offset to be written even while a filter is being computed without disturbing operation. The next sample comes out of the CRCF with the new filter.

CRCF Taps <6:0>

This register is written with one less than the number of taps that are calculated by the CRCF filter. The filter length is given by the decimal value of this register plus 1. A value of 0 represents a 1-tap filter, and a value of 0x28 (40 decimal) represents a 41-tap filter.

CRCF Coefficient Memory

This memory has 64 words that have 20 bits each. The memory contains the CRCF filter coefficients. The data written to this memory space is 20-bit in twos complement format. See the Channel RAM Coefficient Filter section for the method to program the coefficients into the coefficient memory.

AGC Control Register <10:0>

<10>: Channel Sync Select Bit. When this bit is set, the AGC uses the sync signal from the channel for its synchronization. When this bit is cleared, the SYNC pin used for synchronization is defined by Bits <9:8> of this register.

<9:8>: SYNC Pin Select Bits. When Bit <10> of this register is cleared, these bits specify the SYNC pin used by AGC for synchronization. These bits are don't care when Bit <10> of the AGC control register is set to Logic 1.

Table 41. SYNC Pin Select Bits

AGC Control Bits <9:8>	SYNC Pin Used by AGC
00	SYNC0
01	SYNC1
10	SYNC2
11	SYNC3

<7:5>: AGC Word Length Control Bits. These bits define the word length of the AGC output. The output word can be 4 bits to 8 bits, 10 bits, 12 bits, or 16 bits wide. Table 42 shows the possible selections.

Table 42. AGC Word Length Control Bits

AGC Control Bits <7:5>	Output Word Length (Bits)
000	16
001	12
010	10
011	8
100	7
101	6
110	5
111	4

<4>: AGC Mode Bit. When this bit is cleared, the AGC operates to maintain a desired signal level. When this bit is set, it operates to maintain a constant clipping level. See the Automatic Gain Control section for details about these modes.

<3>: AGC Sync Now Bit. This bit is used to synchronize a particular AGC irrespective of the channel through the programming ports (microport or serial port). When this bit is set, the AGC block updates a new output sample (RSSI sample) and starts working toward a new update sample.

<2>: Initialize on Sync Bit. This bit is used to determine whether or not the AGC should initialize on a sync. When this bit is set during a synchronization, the CIC filter is cleared and new values for CIC decimation, number of averaging samples, CIC scale, signal gain K, and pole parameter P are loaded. When Bit <2> = 0, the above-mentioned parameters are not updated, and the CIC filter is not cleared. In both cases, an AGC update sample is output from the CIC filter, and the decimator starts operating toward the next output sample whenever a sync occurs.

<1>: First Sync Only. This bit is used to ignore repetitive synchronization signals. In some applications, the synchronization signal occurs periodically. If this bit is cleared, each synchronization request resynchronizes the AGC. If this bit is set, only the first occurrence causes the AGC to synchronize and updates the AGC gain values periodically, depending on the decimation factor of the AGC CIC filter.

<0>: AGC Bypass Bit. When this bit is set, the AGC section is bypassed. The N-bit representation from the interpolating half-band filters is still reduced to a lower bit width representation as set by Bits <7:5> of the AGC control register. A truncation at the output of the AGC accomplishes this task.

AGC Hold-Off Register <15:0>

The AGC hold-off counter is loaded with the value written to this address when either a soft sync or pin sync comes into the channel. The counter begins counting down. When it reaches 1, a sync is sent to the AGC. This sync may or may not initialize the AGC, as defined by the control word. The AGC loop is updated with a new sample from the CIC filter whenever a sync occurs. If this register is Logic 1, the AGC is updated immediately when the sync occurs. If this register is Logic 0, the AGC cannot be synchronized.

AGC Update Decimation <11:0>

This 12-bit register sets the AGC decimation ratio from 1 to 4096. An appropriate scaling factor should be set to avoid loss of bits. The decimation ratio is given by the decimal value of the AGC update decimation<11:0> register contents plus 1, that is, 12'0x000 describes a decimation ratio of 1, and 12'0xFFF describes a decimation ratio of 4096.

AGC Signal Gain <11:0>

This register is used to set the initial value for a signal gain used in the gain multiplier. This 12-bit value sets the initial signal gain in the range of 0 dB and 96.296 dB in steps of 0.024 dB. Initial signal gain (SG) in dB should be converted to a register setting by

$$Register\ Value = round\left[\frac{SG}{20\log_{10}(2)}\times 256\right]$$

AGC Error Threshold <11:0>

This 12-bit register is the comparison value used to determine which loop gain value (K_1 or K_2) to use for optimum operation. When the magnitude-of-error signal is less than the AGC error threshold value, then K_1 is used; otherwise, K_2 is used. The word format of the AGC error threshold register is four bits to the left of the binary point and eight bits to the right. See the Automatic Gain Control section for details.

$$Register\ Value = round\left[\frac{Error\ Threshold}{20\log_{10}(2)}\times 256\right]$$

AGC Average Samples <5:0>

This 6-bit register contains the scale used for the CIC filter and the number of power samples to be averaged before being sent to the CIC filter.

<5:2>: CIC Scale. This 4-bit word defines the scale used for the CIC filter. Each increment of this word increases the CIC scale by 6.02 dB.

<1:0>: Number of AGC Average Samples. This defines the number of samples to be averaged before they are sent to the CIC decimating filter (see Table 43).

Table 43. Number of AGC Average Samples

AGC Average Samples <1:0>	Number of Samples Taken
00	1
01	2
10	3
11	4

AGC Pole Location <7:0>

This 8-bit register is used to define the open-loop filter pole location P. Its value can be set from 0 to 0.996 in steps of 0.0039. This value of P is updated in the AGC loop each time the AGC is initialized. This open-loop pole location directly impacts the closed-loop pole locations, see the Automatic Gain Control section.

AGC Desired Level <7:0>

This register contains the desired signal level or desired clipping level, depending on operational mode. This desired request level (R) can be set in dB from 0 to 23.99 in steps of 0.094 dB. The request level (R) in dB should be converted to a register setting by

$$Register\ Value = round\left[\frac{R}{20\log_{10}(2)}\times 64\right]$$

AGC Loop Gain2 <7:0>

This 8-bit register is used to define the second possible open-loop gain, K_2. Its value can be set from 0 to 0.996 in steps of 0.0039. This value of K_2 is updated each time the AGC is initialized. When the magnitude-of-error signal in the loop is greater than the AGC error threshold, then K_2 is used by the loop. K_2 is updated only when the AGC is initialized.

AGC Loop Gain1 <7:0>

This 8-bit register is used to define the open-loop gain K_1. Its value can be set from 0 to 0.996 in steps of 0.0039. This value of K is updated in the AGC loop each time the AGC is initialized. When the magnitude-of-error signal in the loop is less than the AGC error threshold, then K_1 is used by the loop. K_1 is updated only when the AGC is initialized.

I Path Signature Register <15:0>

This 16-bit signature register is for the I path of the channel logic. The signature register records data on the networks that leave the channel logic just before entering the second data router.

Q Path Signature Register <15:0>

This 16-bit signature register is for the Q path of the channel logic. The signature register records data on the networks that leave the channel logic just before entering the second data router.

BIST Control <15:0>

<15>: Disable Signature Generation Bit. When this bit is active high, the signature registers do not produce a pseudorandom output value, but instead directly load the 24-bit input data. When this bit is cleared, the signature register produces a pseudorandom output for every clock cycle that it is active. See the User-Configurable, Built-In Self-Test (BIST) section for details.

<14:0>: BIST Timer Bits. The <14:0> bits of this register form a 15-bit word that is loaded into the BIST timer. After loading the BIST timer, the signature register is enabled for operation while the timer is actively counting down. (See the User-Configurable, Built-In Self-Test (BIST) section.)

OUTPUT PORT REGISTER MAP

This part of the memory map deals with the output data and controls for parallel output ports.

Parallel Port Output Control <23:0>

<23>: Port C Append RSSI Bit. When this bit is set, an RSSI word is appended to every I/Q output sample, irrespective of whether the RSSI word is updated in the AGC. When this bit is cleared, an RSSI word is appended to an I/Q output sample only when the RSSI word is updated. The RSSI word is not output for subsequent I/Q samples until the next time the RSSI is updated in the AGC.

<22>: Port C, Data Format Bit. When this bit is set, the port is configured for 8-bit parallel I/Q mode. When cleared, the port is configured for 16-bit interleaved I/Q mode. See the Parallel Port Output section for details.

<21>: Port C, AGC5 Enable Bit. When this bit is set, AGC5 data (I/Q data) is output on parallel Output Port C (data bus). When this bit is cleared, AGC5 data does not appear on Output Port C.

<20>: Port C, AGC4 Enable Bit. Similar to Bit <21> for AGC4.

<19>: Port C, AGC3 Enable Bit. Similar to Bit <21> for AGC3.

<18>: Port C, AGC2 Enable Bit. Similar to Bit <21> for AGC2.

<17>: Port C, AGC1 Enable Bit. Similar to Bit <21> for AGC1.

<16>: Port C, AGC0 Enable Bit. Similar to Bit <21> for AGC0.

<15>: Port B Append RSSI Bit. When this bit is set, an RSSI word is appended to every I/Q output sample, irrespective of whether or not the RSSI word is updated in the AGC. When this bit is cleared, an RSSI word is appended to an I/Q output sample when the RSSI word is updated only. The RSSI word is not output for subsequent I/Q samples until the next time the RSSI is updated in the AGC.

<14>: Port B, Data Format Bit. When this bit is set, the port is configured for 8-bit parallel I/Q mode. When this bit is cleared, the port is configured for 16-bit interleaved I/Q mode. See the Parallel Port Output section.

<13>: Port B, AGC5 Enable Bit. When this bit is set, AGC5 data (I/Q data) is output on parallel output Port A (data bus). When this bit is cleared, AGC5 data does not appear on output Port C.

<12>: Port B, AGC4 Enable Bit. Similar to Bit <13> for AGC4.

<11>: Port B, AGC3 Enable Bit. Similar to Bit <13> for AGC3.

<10>: Port B, AGC2 Enable Bit. Similar to Bit <13> for AGC2.

<9>: Port B, AGC1 Enable Bit. Similar to Bit <13> for AGC1.

<8>: Port B, AGC0 Enable Bit. Similar to Bit <13> for AGC0.

<7>: Port A Append RSSI Bit. When this bit is set, an RSSI word is appended to every I/Q output sample, irrespective of whether or not the RSSI word is updated in the AGC. When this bit is cleared, an RSSI word is appended to an I/Q output sample only when the RSSI word is updated. The RSSI word is not output for subsequent I/Q samples until the next time RSSI is updated again in the AGC.

<6>: Port A, Data Format Bit. When this bit is set, the port is configured for 8-bit parallel I/Q mode. When this bit is cleared, the port is configured for 16-bit interleaved I/Q mode. See the Parallel Port Output section.

<5>: Port A, AGC5 Enable Bit. When this bit is set, AGC5 data (I/Q data) is output on parallel output Port A (data bus). When this bit is cleared, AGC5 data does not appear on output Port C.

<4>: Port A, AGC4 Enable Bit. Similar to Bit <5> for AGC4.

<3>: Port A, AGC3 Enable Bit. Similar to Bit <5> for AGC3.

<2>: Port A, AGC2 Enable Bit. Similar to Bit <5> for AGC2.

<1>: Port A, AGC1 Enable Bit. Similar to Bit <5> for AGC1.

<0>: Port A, AGC0 Enable Bit. Similar to Bit <5> for AGC0.

Output Port Control <9:0>

<9:8>: PCLK Divisor Bits. When a parallel port is in master mode, the PCLK is derived from the PLL_CLK. These bits define the value of the divisor used to divide the PLL_CLK to obtain the PCLK. These bits are don't care in slave mode.

Table 44. PCLK Divisor Bits

PCLK Divisor <9:8>	Divisor Value
00	1
01	2
10	4
11	8

<7>: PCLK Master Mode Bit. When the PCLK master mode bit is set, the PCLK pin is configured as an output and the PCLK is driven by the PLL_CLK. Data is transferred out of the AD6636 synchronous to this output clock. When this bit is cleared, the PCLK pin is configured as an input. The user is required to provide a PCLK, and data is transferred out of the AD6636 synchronous to this input clock. On power-up, this bit is cleared to avoid contention on the PCLK pin.

<6:4>: Complex Control Bits. These bits are described in Table 45.

Table 45. Complex Control Bits

Complex Control <6:4>		Comment
000	No complex filters	Stream control register controls AGC usage.
001	Str0/Str1 combined	Ch 0 and Ch 1 form a complex filter.
010	Str0/Str1 combined, Str2/Str3 combined	Ch 0 and Ch 1 form a complex filter; Ch 2 and Ch 3 form a complex filter.
011	Str0/Str1 combined, Str2/Str3 combined, Str4/Str5 combined	Ch 0 and Ch 1 form a complex filter; Ch 2 and Ch 3 form a complex filter; Ch 4 and Ch 5 form a complex filter.
101	Str0/Str1 combined	Ch 0 and Ch 1 form a biphase filter.
110	Str0/Str1 combined, Str2/Str3 combined	Ch 0 and Ch 1 form a biphase filter; Ch 2 and Ch 3 to form a biphase filter.
111	Str0/Str1 combined, Str2/Str3 combined, Str4/Str5 combined	Ch 0 and Ch 1 to form a biphase filter; Ch 2 and Ch 3 to form a biphase filter; Ch 4 and Ch 5 to form a biphase filter.

<3:0>: Stream Control Bits. These bits are described in Table 46.

Table 46. Stream Control Bits

Stream Control Bits	Output Streams (str0, str1, str2, str3, str4, str5)	No. of Streams
0000	Ch 0/Ch 1 combined; Ch 2, Ch 3, Ch 4, Ch 5 independent	5
0001	Ch 0/Ch 1/Ch 2 combined; Ch 3, Ch 4, Ch 5 independent	4
0010	Ch 0/Ch 1/Ch 2/Ch 3 combined; Ch 4, Ch 5 independent	3
0011	Ch 0/Ch 1/Ch 2/Ch 3/Ch 4 combined; Ch 5 independent	2
0100	Ch 0/Ch 1/Ch 2/Ch 3/Ch 4/Ch 5 combined	1
0101	Ch 0/Ch 1/Ch 2 combined, Ch 3/Ch 4/Ch 5 combined	2
0110	Ch 0/Ch 1 combined, Ch 2/Ch 3 combined, Ch 4/Ch 5 combined	3
0111	Ch 0/Ch 1 combined, Ch 2/Ch 3 combined, Ch 4, Ch 5 independent	4
1000	Ch 0/Ch 1/Ch 2 combined, Ch 3/Ch 4 combined, Ch 5 independent	3
1001	Ch 0/Ch 1/Ch 2/Ch 3 combined, Ch 4/Ch 5 combined.	2
Default	Independent channels	6

AGC0, I Output <15:0>

This read-only register provides the latest in-phase output sample from AGC0. Note that AGC0 may be bypassed, and that AGC0 here is representative of the datapath only.

AGC0, Q Output <15:0>

This read-only register provides the latest quadrature-phase output sample from AGC0. Note that AGC0 may be bypassed, and that AGC0 here is representative of the datapath only.

AGC1, I Output <15:0>

This read-only register provides the latest in-phase output sample from AGC1. Note that AGC1 may be bypassed and that AGC1 here is representative of the datapath only.

AGC1, Q Output <15:0>

This read-only register provides the latest quadrature-phase output sample from AGC1. Note that AGC1 may be bypassed and that AGC1 here is representative of the datapath only.

AGC2, I Output <15:0:

This read-only register provides the latest in-phase output sample from AGC2. Note that AGC2 may be bypassed and that AGC2 here is representative of the datapath only.

AGC2, Q Output <15:0>

This read-only register provides the latest quadrature-phase output sample from AGC2. Note that AGC2 may be bypassed and that AGC2 here is representative of the datapath only.

AGC3, I Output <15:0>

This read-only register provides the latest in-phase output sample from AGC3. Note that AGC3 may be bypassed and that AGC3 here is representative of the datapath only.

AGC3, Q output <15:0>

This read-only register provides the latest quadrature-phase output sample from AGC3. Note that AGC3 may be bypassed and that AGC3 here is representative of the datapath only.

AGC4, I Output <15:0>

This read-only register provides the latest in-phase output sample from AGC4. Note that AGC4 may be bypassed and that AGC4 here is representative of the datapath only.

AGC4, Q Output <15:0>

This read-only register provides the latest quadrature-phase output sample from AGC4. Note that AGC4 may be bypassed and that AGC4 here is representative of the datapath only.

AGC5, I Output <15:0>

This read-only register provides the latest in-phase output sample from AGC5. Note that AGC5 may be bypassed and that AGC5 here is representative of the datapath only.

AGC5, Q Output <15:0>

This read-only register provides the latest quadrature-phase output sample from AGC5. Note that AGC5 may be bypassed and that AGC5 here is representative of the datapath only.

AGC0, RSSI Output <11:0>

This read-only register provides the latest RSSI output sample from AGC0. This register is updated only when AGC0 is enabled and operating.

AGC1, RSSI Output <11:0>

This read-only register provides the latest RSSI output sample from AGC1. This register is updated only when AGC1 is enabled and operating.

AGC2, RSSI Output <11:0>

This read-only register provides the latest RSSI output sample from AGC2. This register is updated only when AGC2 is enabled and operating.

AGC3, RSSI Output <11:0>

This read-only register provides the latest RSSI output sample from AGC3. This register is updated only when AGC3 is enabled and operating.

AGC4, RSSI Output <11:0>

This read-only register provides the latest RSSI output sample from AGC4. This register is updated only when AGC4 is enabled and operating.

AGC5, RSSI Output <11:0>

This read-only register provides the latest RSSI output sample from AGC5. This register is updated only when AGC5 is enabled and operating.

DESIGN NOTES

The following guidelines describe circuit connections, layout requirements, and programming procedures for the AD6636. The designer should review these guidelines before starting the system design and layout.

- The AD6636 requires the following power-up sequence. The VDDCORE (1.8 V) must settle into nominal voltage levels before the VDDIO attains the minimum. This ensures that, on power-up, the JTAG does not take control of the I/O pins.
- Input clocks (CLKA, CLKB, CLKC, CLKD) and input port pins (INA[15:0] to IND[15:0], EXPA[2:0] to EXPD[2:0]) are not 5 V tolerant. Care should be taken to drive these pins within the limits of VDDIO (3.0 V to 3.6 V).
- When the ADC output has less than 16 bits of resolution, it should be connected to the MSBs of the input port (MSB-justified). The remaining LSBs should be connected to ground.
- The number format used in this part is twos complement. All input ports and output ports use twos complement data format. The formats for individual internal registers are given in the memory map description of these registers.
- In both microport and serial port operation, the $\overline{\text{DTACK}}$ (RDY, SDO) pin is an open-drain output and therefore should be pulled high externally using a pull-up resister. The recommended value for the pull-up resistor is from 1 kΩ to 5 kΩ.

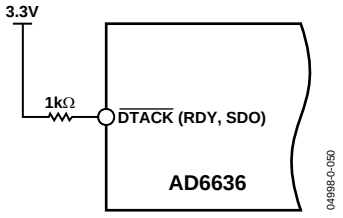


Figure 58. $\overline{\text{DTACK}}$, SDO Pull-Up Resistor Circuit

- A simple RC circuit is used on the EXT_FILTER pin to balance the internal RC circuit on this pin and maintain a good PLL clock lock. The recommended circuit is shown in Figure 59, with the RC circuit connected to VDDCORE. This RC circuit should be placed as close as possible to the AD6636 part. This layout ensures that the PLL clock is void of noise and spurs and the PLL lock is maintained closely.

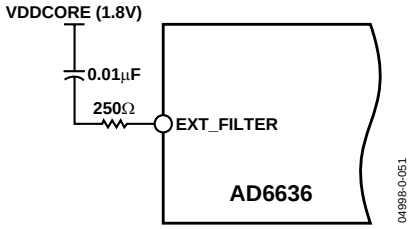


Figure 59. EXT_FILTER Circuit for PLL Clock

- By default, the PLL CLK is disabled. It can be enabled by programming the PLL multiplier and divider bits in the ADC CLK control register. When the PLL CLK is enabled by programming this register, it takes between 50 µs and 200 µs to settle. While the PLL loop settles, the voltage at the EXT_FILTER pin increases from 0 V to VDDCORE (1.8 V) and settles there. Channel registers and output port registers (Address 0x68 to Address 0xE7) should not be programmed before the PLL loop settles.
- The LVDS_RSET pin is used to calibrate the current in the LVDS pads. The recommended circuit for this pin is shown in Figure 60. This resistor should be placed as close as possible to the AD6636 part. If CMOS mode input is used, this resistor is not required.

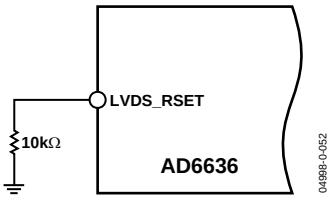


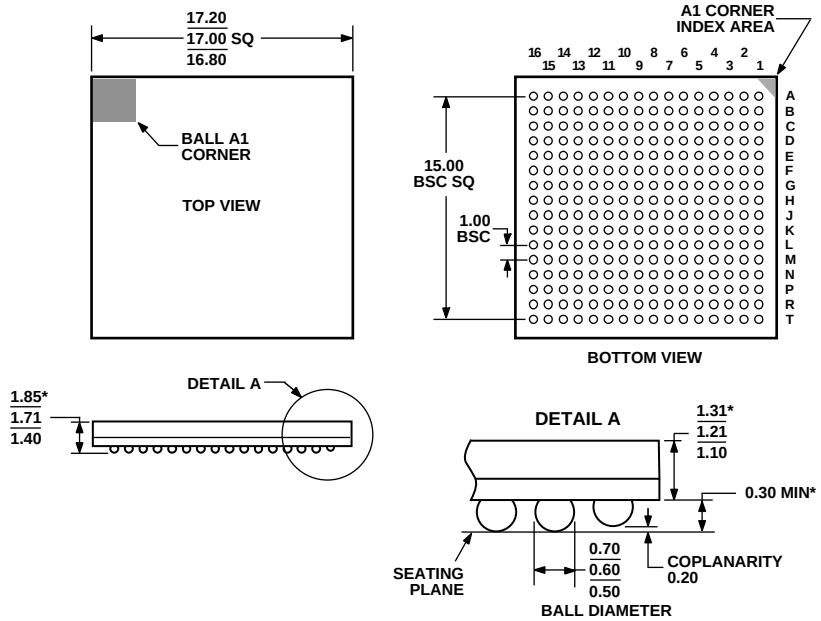
Figure 60. LVDS_RSET Circuit for LVDS Calibration

- To reset the AD6636 part, the user needs to provide a minimum pulse of 30 ns to the $\overline{\text{RESET}}$ pin. The $\overline{\text{RESET}}$ pin should be connected to GND (or pulled low) during power-up of the part. The $\overline{\text{RESET}}$ pin can be pulled high after the power supplies have settled to nominal values (1.8 V and 3.3 V). At this point, a pulse (pull low and high again) should be provided to give a $\overline{\text{RESET}}$ to the part.
- Most AD6636 pins are driven by both JTAG circuitry and normal function circuitry specific to each pin. $\overline{\text{TRST}}$ is the reset pin for JTAG. When $\overline{\text{TRST}}$ is pulled low, JTAG is in reset and all pins function in normal mode (driven by the functional circuit). If JTAG is not used in the design, the $\overline{\text{TRST}}$ pin should be pulled low at all times.

If JTAG is used, the designer should ensure that the $\overline{\text{TRST}}$ pin is pulled low during power-up. After the power supplies have settled to nominal values (1.8 V and 3.3 V), the $\overline{\text{TRST}}$ pin can be pulled high for JTAG control. When JTAG control is no longer required, the $\overline{\text{TRST}}$ pin should ideally be pulled low again.

- The CPUCLK (SCLK) is the clock used for programming via the microport (serial port). This clock needs to be provided by the designer to the part (slave clock). The designer should ensure that this clock's frequency is less than or equal to the frequency of the CLKA signal. Additionally, the frequency of the CPUCLK (SCLK) should always be less than 100 MHz.
- CLKA, CLKB, CLKC, and CLKD are used as individual clocks to input data into Input Port A, Input Port B, Input Port C, and Input Port D, respectively. These clocks must have the same frequency and should be generated ideally from the same clock source. Note that CLKA is used to drive the internal circuitry and the PLL clock multiplier. Therefore, even if Input Port A is not used, CLKA should be driven by the input clock.
- The microport data bus is 16 bits wide. Both 8-bit and 16-bit modes are available using this part. If 8-bit mode is used, the MSB of the data bus (D[15:8]) can be left floating or connected to GND.
- The output parallel port has a one clock cycle overhead. If two channels (with the same data rates) are output on one output port in 16-bit interleaved I/Q mode along with an AGC word, this requires three clock cycles for one sample from each channel (one clock each for I data, Q data, and gain data). Therefore, the total number of clock cycles required to output the data is 3 clocks/channel × 2 channels + 1 (overhead) = 7 clock cycles.
- The number of clock cycles required for each channel can be 3 (interleaved I + Q + gain word), 2 (parallel I /Q + gain), 2 (interleaved I + Q), or 1 (interleaved I/Q). Designers should make sure that sufficient time is allowed to output these channels on one output port. Also note that the I, Q, and gain for a particular channel all come out on a single output port and cannot be divided among output ports.
- When CRCF and DRCF filters are disabled, the coefficient memory cannot be read back because the clock to the coefficient RAM is also off.
- In the Intel mode microport, the beginning of a read and write access is indicated by the RDY pin going low. The access is complete only when the RDY pin goes high. In the Motorola mode microport, the completion of a read and write access is indicated by the $\overline{\text{DTACK}}$ going low. In both modes, $\overline{\text{CS}}$, $\overline{\text{RD}}$ ($\overline{\text{DS}}$), and $\overline{\text{WR}}$ (R/$\overline{\text{W}}$) should be active until access is complete; otherwise, an incomplete access results.
- In both Intel and Motorola microport modes care should be taken not to read or write, to or from, addresses beyond those defined by the memory map (Address 0xE8 to Address 0xFF). Attempting to access these addresses causes the bus to hang because $\overline{\text{DTACK}}$ (RDY) does not go high to signal the end of the access.
- In both Intel and Motorola modes, if $\overline{\text{CS}}$ is held low even after microport read or write access is complete, the microport initiates a second access. This is a problem while writing or reading from coefficient RAM, where each access writes to or reads from a different RAM address. This can be corrected by writing to one coefficient RAM address at a time, that is, the coefficient start and stop address registers have the same value.
- In SPI mode programming, the $\overline{\text{SCS}}$ pin needs to go high (inactive) after writing or reading each byte (eight clock cycles on the SCLK pin).

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-192-AAF-1
EXCEPT FOR DIMENSIONS INDICATED BY A "*" SYMBOL.

Figure 61. 256-Lead Chip Scale Ball Grid Array [CSP_BGA]
(BC-256-2)
Dimensions shown in millimeters

ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
AD6636BBCZ[1]	−40°C to +85°C	6-Channel Part, 256-Lead CSP_BGA	BC-256-2
AD6636CBCZ[1]	−40°C to +85°C	4-Channel Part, 256-Lead CSP_BGA	BC-256-2
AD6636BC/PCB		Evaluation Board with AD6636 (6-Channel Part) and Software	

[1] Z = Pb-free part.

NOTES

© 2005 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
D04998–0–6/05(A)

ANALOG DEVICES

www.analog.com