CMX983
Analogue Front End (AFE) for Digital Radio

D/983/8 June 2018

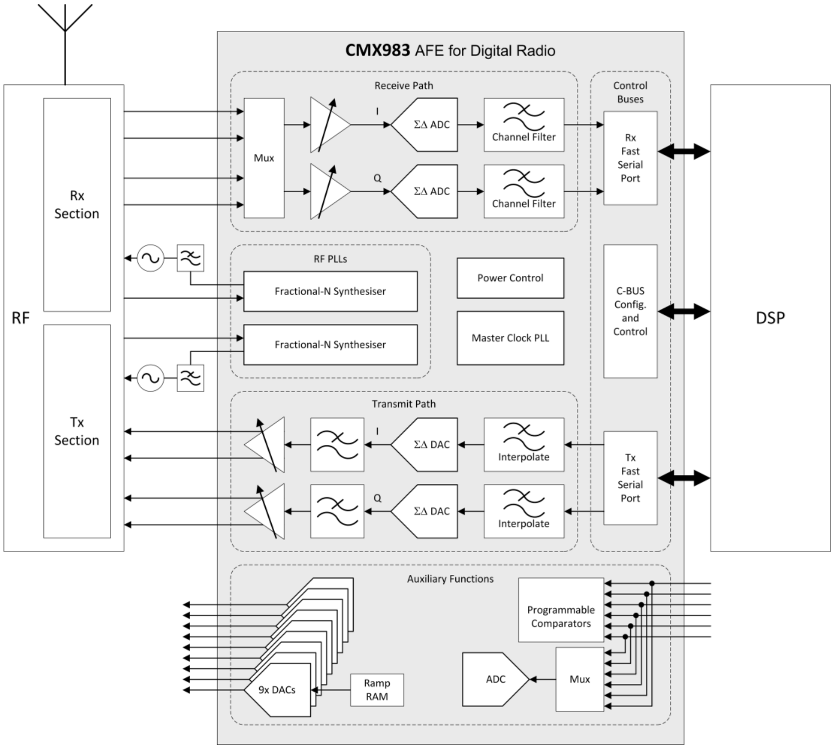
Provisional Information

Features

- **Rx channel**
  - Two 16-bit Σ-Δ ADCs
  - Programmable Channel Filter
- **Tx Channel**
  - Two 14-bit Σ-Δ DACs
  - Programmable Channel Filter
- **RF Support**
  - Two 2.1GHz Fractional-N Synthesisers
- **Auxiliary Functions**
  - 10-bit ADC supporting 10 inputs
  - Five Analogue Comparators
  - 10-bit DACs driving 9 outputs
- **DSP Interface**
  - C-BUS Control and Configuration Port
  - Fast Serial Interface for Rx/Tx Data
- **Narrow bandwidth and wide bandwidth operating modes**
- **Direct connection to**
  - CMX998 Cartesian Loop Transmitter
  - CMX994 Direct Conversion Receiver
- **Low power operation**
  - 3.3V and 1.8V supplies
- **Small 64-pin VQFN Package**

Applications

- Software Defined Radio (SDR)
- Satellite Communication
- Wireless Data Terminals
- Digital PMR/LMR Radio
  - TETRA, DMR, PDT, APCO P25, etc.



1 Brief Description

The CMX983 is an Analogue Front End (AFE) IC that bridges the gap between a digital radio's RF section and the DSP. Specifically designed to meet the needs of a Software Defined Radio (SDR), the CMX983 performs critical DSP-intensive functions, provides dual channel analogue to digital and digital to analogue conversion, includes two RF fractional-N synthesisers, and embeds a host of auxiliary ADCs and DACs for use within the radio system.

The CMX983 meets the low operating power requirements of SDR terminals and is powered from separate 3.3V and 1.8V power supplies. A facility is provided to allow the synthesiser charge pumps to be operated at up to 5V, providing low noise operation. The device is available in a small 64-lead VQFN package.

CONTENTS

Section Page

Table Page

Figure Page

It is recommended that you check for the latest product datasheet version from the Datasheets page of the CML website: [www.cmlmicro.com].

History

Version	Changes	Date
8	- Section 15.3 (Operating Characteristics), Auxiliary DAC maximum offset voltage increased from 10mV to 12mV	20th June 2018
7	- Wide bandwidth (50kHz) mode added.	4th May 2016
6	- Section 15.3 (Operating Characteristics), Synthesiser 1 and 2, RF Input Sensitivity: Note 12 added, providing additional clarification and qualification to the stated figures.	30th June 2015
5	- Section 15.5 – Typical Performance Characteristics added	28th Nov 2014
4	- Section 12 – Diagram modified, advice regarding connection to single ended VCO added - Section 12 – PLL1_CON and PLL2_CON bits 13-11 description expanded - Section 12 – PLL1_FLCK and PLL2_FLCK bits 1-0 description expanded - Section 12 – PLL1_BLEED and PLL2_BLEED guidelines added - Section 12.2 – References to modulator type corrected - Section 12.3, 12.3.1, 12.3.2 – lock detector configuration guidelines added - Section 15 – Performance Specification: corrections made	22nd Sept 2014
3	- New title for device and datasheet - Front page – features, applications and system diagram replaced - Brief Description and General Description rewritten - Section 12 – Frac-N Synthesisers: expanded to include mode select options - Section 12 – PLL1_BLEED - $51: 8-bit Write and PLL2_BLEED - $5A: 8-bit Write registers added - Section 12 – PLL_CFG - $CE: 16-bit Write register added - Section 13, Figure 22 edited to show alternate routing for AuxADC Channels 6 and 7 - Section 15 – Performance Specification: entire section rewritten to include data from Evaluation testing	6th May 2014
2	Front page applications list added and brief description updated, addition of 2 extra AuxADC inputs, PLL Lock Detect output, clarification of PLL status and the application of RESETN, standardisation of power supply nomenclature.	2nd Aug 2013
1	First release, Advance Information	20th May 2013

2 Block Diagram

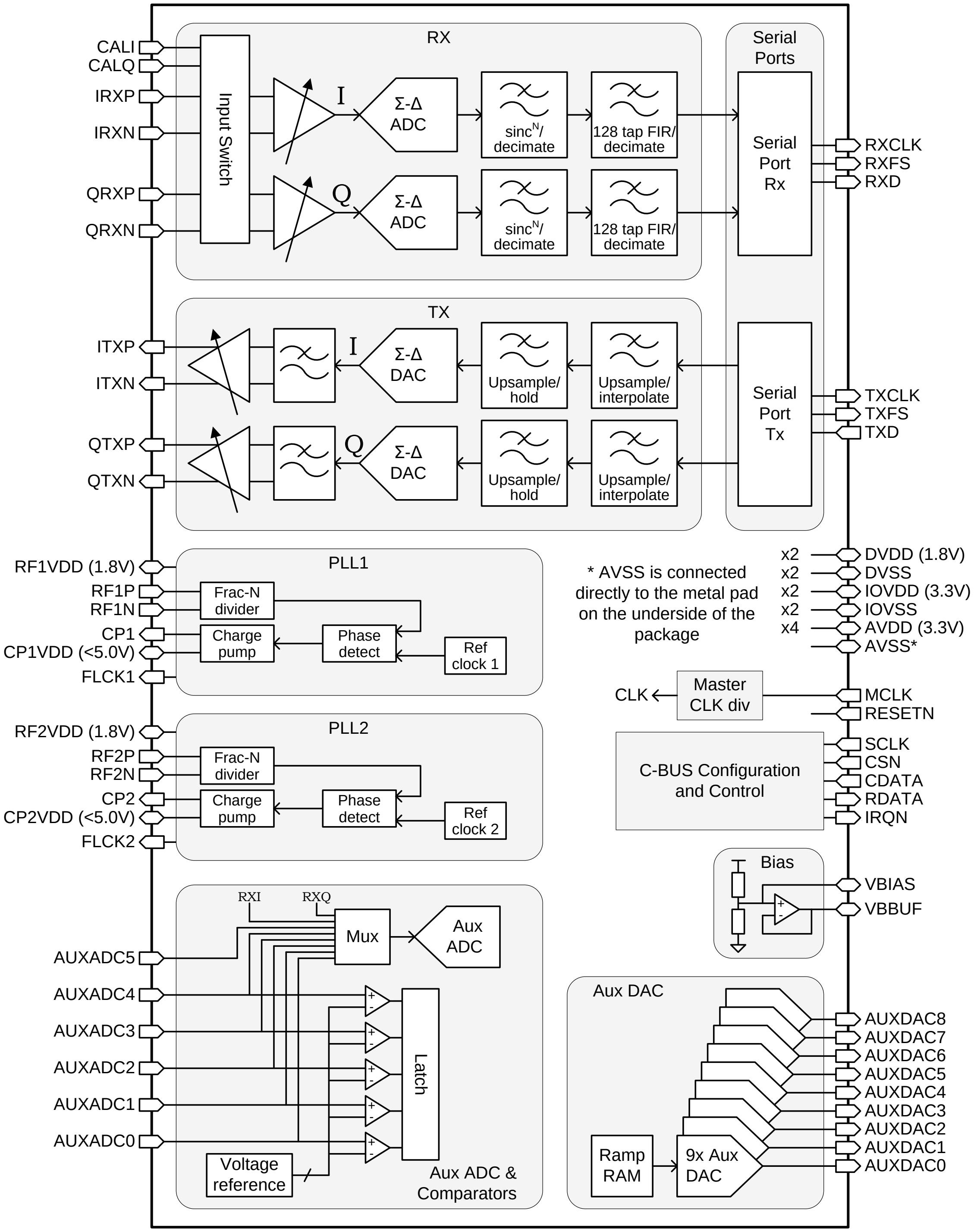


Figure 1 Block Diagram

3 General Description

The CMX983 is an Analogue Front End (AFE) for a DSP used in Software Defined Radio systems and acts as a bridge between the analogue and digital sections of advanced digital radio systems. The device also performs critical DSP-intensive functions with low operating power thereby reducing the overall system power consumption.

The receive path accepts differential analogue baseband I/Q signals. These are converted to digital, decimated and passed through programmable FIR channel filters to simplify host DSP processing and data extraction. The transmit path accepts digital I/Q data streams. These are up-sampled, interpolated and converted to analogue format, then driven off-chip for external up-conversion and transmission.

Two fractional-N RF synthesisers are provided which are capable of operating with external VCOs of up to 2.1GHz. These have the advantage, over integer-N synthesisers, of allowing a higher PLL reference frequency, which gives improved noise performance and agility. The fractional modulator can be configured for either 16-bit or 24-bit operation, enabling very fine frequency resolution. The synthesisers also include a fast-lock feature that helps minimise lock time when switching channels and provides an indication of when lock is achieved.

An auxiliary ADC (with multiplexed inputs) and a number of auxiliary DACs are included for control and measurement functions such as AFC, AGC and RSSI. The ADC has a digital threshold compare function, and one of the DACs has a programmable auto-ramping feature that is especially useful for controlling the ramp-up and ramp-down profile of the transmitter power amplifier.

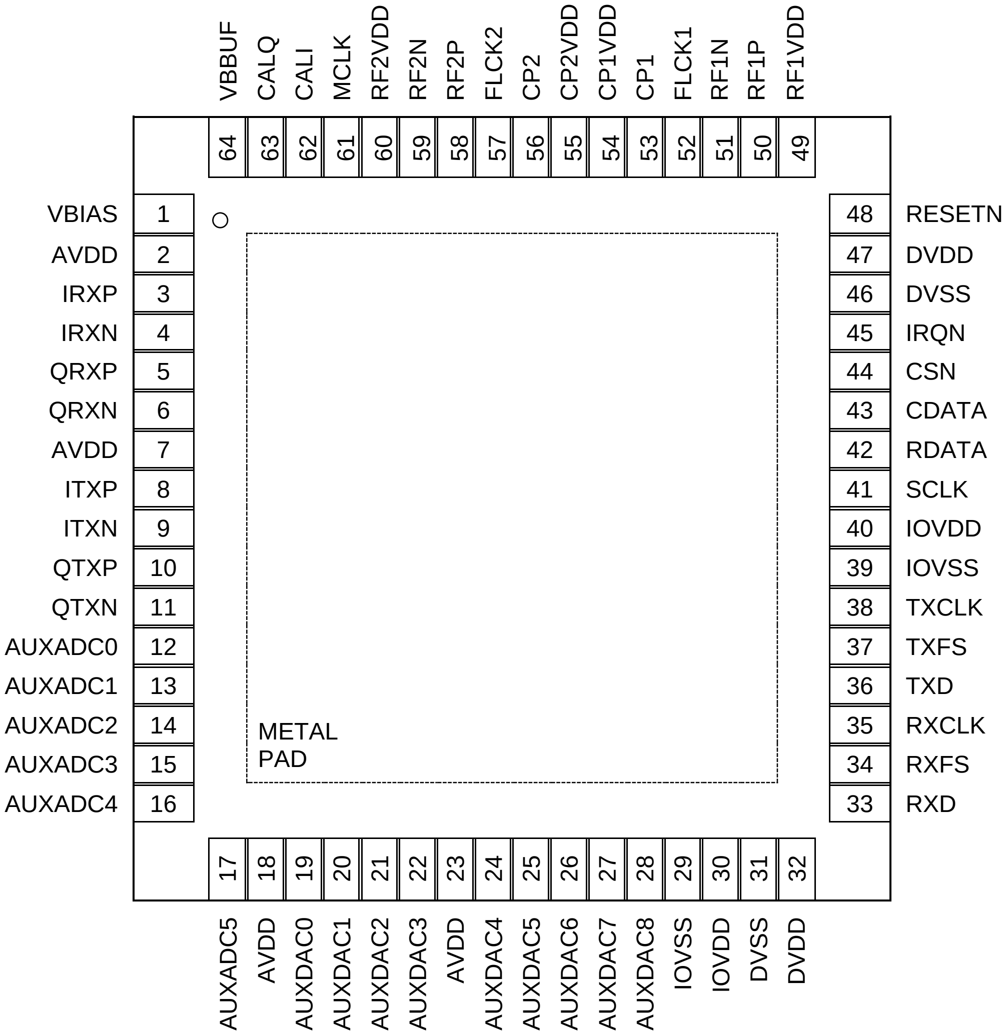
Five analogue comparators with programmable thresholds are provided; these share the auxiliary ADC input pins and give the user the option of sensing monitoring signals with very low power consumption.
The CMX983 is clocked from a full-swing logic level, or by low-amplitude sine wave or clipped sine wave. This may be used directly as a reference source or fed to an on-chip PLL capable of generating a wide range of internal clock frequencies.

The CMX983 is suitable for radio systems employing channel bandwidths up to 25kHz in normal bandwidth and 50kHz in wide bandwidth modes to support satellite communication, high performance wireless data and professional two-way radio systems. The CMX983 is highly configurable and supports numerous sample rates and filtering characteristics, which enables a high level of functionality, integration and connectivity with RF building block ICs. The CMX983 connects seamlessly with CML’s CMX994, CMX994A and CMX994E Direct Conversion Receivers and the CMX998 Cartesian Feedback Loop Transmitter to provide a complete, small form factor, RF-to-digital baseband solution.

The main ADCs and DACs can be operated with wider bandwidth signals by doubling their sigma-delta clock rates. To optimise the overall performance of the ADCs at these higher clock rates, RX_BIAS0 and RX_BIAS1 register settings should be modified to increase ADC bias current as described in section 9.1.

The Rx and Tx channels interface to the external DSP through dedicated fast serial ports. A separate C-BUS port is provided for general control and configuration of the CMX983 and a configurable interrupt generator may be used to minimise the load that servicing the device places on the host processor.

4 Pin and Signal List



The exposed metal pad at the underside
of the package must connect to AVSS

Figure 2 CMX983Q1 Pin Arrangement (top view)

Table 1 Pin and Signal List

Package Q1 Pin No.	Pin Name	Type	Signal Description
1	VBIAS	O/P	Internally generated bias voltage of VDDA/2
2	AVDD	PWR	Analogue power (3.3V)
3	IRXP	I/P	I channel positive input
4	IRXN	I/P	I channel negative input
5	QRXP	I/P	Q channel positive input
6	QRXN	I/P	Q channel negative input
7	AVDD	PWR	Analogue power (3.3V)
8	ITXP	O/P	Positive output for I channel
9	ITXN	O/P	Negative output for I channel
10	QTXP	O/P	Positive output for Q channel
11	QTXN	O/P	Negative output for Q channel
12	AUXADC0	I/P	Auxiliary ADC 0 input
13	AUXADC1	I/P	Auxiliary ADC 1 input
14	AUXADC2	I/P	Auxiliary ADC 2 input
15	AUXADC3	I/P	Auxiliary ADC 3 input
16	AUXADC4	I/P	Auxiliary ADC 4 input
17	AUXADC5	I/P	Auxiliary ADC 5 input
18	AVDD	PWR	Analogue power (3.3V)
19	AUXDAC0	O/P	Auxiliary DAC 0 output
20	AUXDAC1	O/P	Auxiliary DAC 1 output
21	AUXDAC2	O/P	Auxiliary DAC 2 output
22	AUXDAC3	O/P	Auxiliary DAC 3 output
23	AVDD	PWR	Analogue power (3.3V)
24	AUXDAC4	O/P	Auxiliary DAC 4 output
25	AUXDAC5	O/P	Auxiliary DAC 5 output
26	AUXDAC6	O/P	Auxiliary DAC 6 output
27	AUXDAC7	O/P	Auxiliary DAC 7 output (Auxiliary ADC 6 input can be selected)
28	AUXDAC8	O/P	Auxiliary DAC 8 output (Auxiliary ADC 7 input can be selected)
29	IOVSS	PWR	IO driver ground (0V)
30	IOVDD	PWR	IO driver power (3.3V)
31	DVSS	PWR	Digital ground (0V)
32	DVDD	PWR	Core power (1.8V)
33	RXD	O/P	Serial port receive data
34	RXFS	O/P	Serial port receive frame sync

Package Q1 Pin No.	Pin Name	Type	Signal Description
35	RXCLK	O/P	Serial port receive clock
36	TXD	I/P	Serial port transmit data
37	TXFS	O/P	Serial port transmit frame sync
38	TXCLK	O/P	Serial port transmit clock
39	IOVSS	PWR	IO driver ground (0V)
40	IOVDD	PWR	IO driver power (3.3V)
41	SCLK	I/P	C-BUS serial clock input from the µC
42	RDATA	T/S	C-BUS serial data output (3-state) to the µC
43	CDATA	I/P	C-BUS serial data input from the µC
44	CSN	I/P	C-BUS chip select input (active low) from the µC
45	IRQN	O/P	C-BUS interrupt request (open drain, active low) to the µC
46	DVSS	PWR	Digital ground (0V)
47	DVDD	PWR	Core power (1.8V)
48	RESETN	I/P	Device reset pin (active low)
49	RF1VDD	PWR	RF power (1.8V)
50	RF1P	I/P	PLL1 VCO positive input
51	RF1N	I/P	PLL1 VCO negative input
52	FLCK1	O/P	PLL1 fast-lock output
53	CP1	O/P	PLL1 charge pump output
54	CP1VDD	PWR	PLL1 charge pump input supply
55	CP2VDD	PWR	PLL2 charge pump input supply
56	CP2	O/P	PLL2 charge pump output
57	FLCK2	O/P	PLL2 fast-lock output
58	RF2P	I/P	PLL2 VCO positive input
59	RF2N	I/P	PLL2 VCO negative input
60	RF2VDD	PWR	RF power (1.8V)
61	MCLK	I/P	Master clock input
62	CALI	I/P	I channel test calibration input
63	CALQ	I/P	Q channel test calibration input
64	VBBUF	O/P	Buffered mid-rail reference voltage
PAD	AVSS	PWR	Analogue ground (0V)

Signal Definitions

Notes:

I/P	=	Input
O/P	=	Output
BI	=	Bidirectional
T/S	=	3-state Output
PWR	=	Power Connection
NC	=	No Connection

Table 2 Definition of Power Supply and Reference Voltages

Signal Name	Pins	Usage
V_{DD} Analogue, AV_{DD}	AVDD	3.3V positive supply rail for the analogue circuits
V_{BIAS}	VBIAS	Internal analogue reference level, derived from AV_{DD}
V_{BBUF}	VBBUF	Buffered mid-rail reference voltage (=AV_{DD} /2)
V_{DD} RF, $RF1V_{DD}$, $RF2V_{DD}$	RF1VDD, RF2VDD	1.8V positive supply rail for RF power
V_{DD} Charge Pump, $CP1V_{DD}$, $CP2V_{DD}$	CP1VDD, CP2VDD	<5.0V positive supply rail for the Charge Pumps
V_{SS} Analogue, AV_{SS}	AVSS	Ground for all analogue circuits (central metal pad)
IOV_{DD}	IOVDD	3.3V positive supply rail for the I/O pads
DV_{DD}	DVDD	1.8V positive supply rail for the digital core circuits
V_{SS} Digital, DV_{SS}	DVSS, IOVSS	Ground for all digital circuits

5 External Components

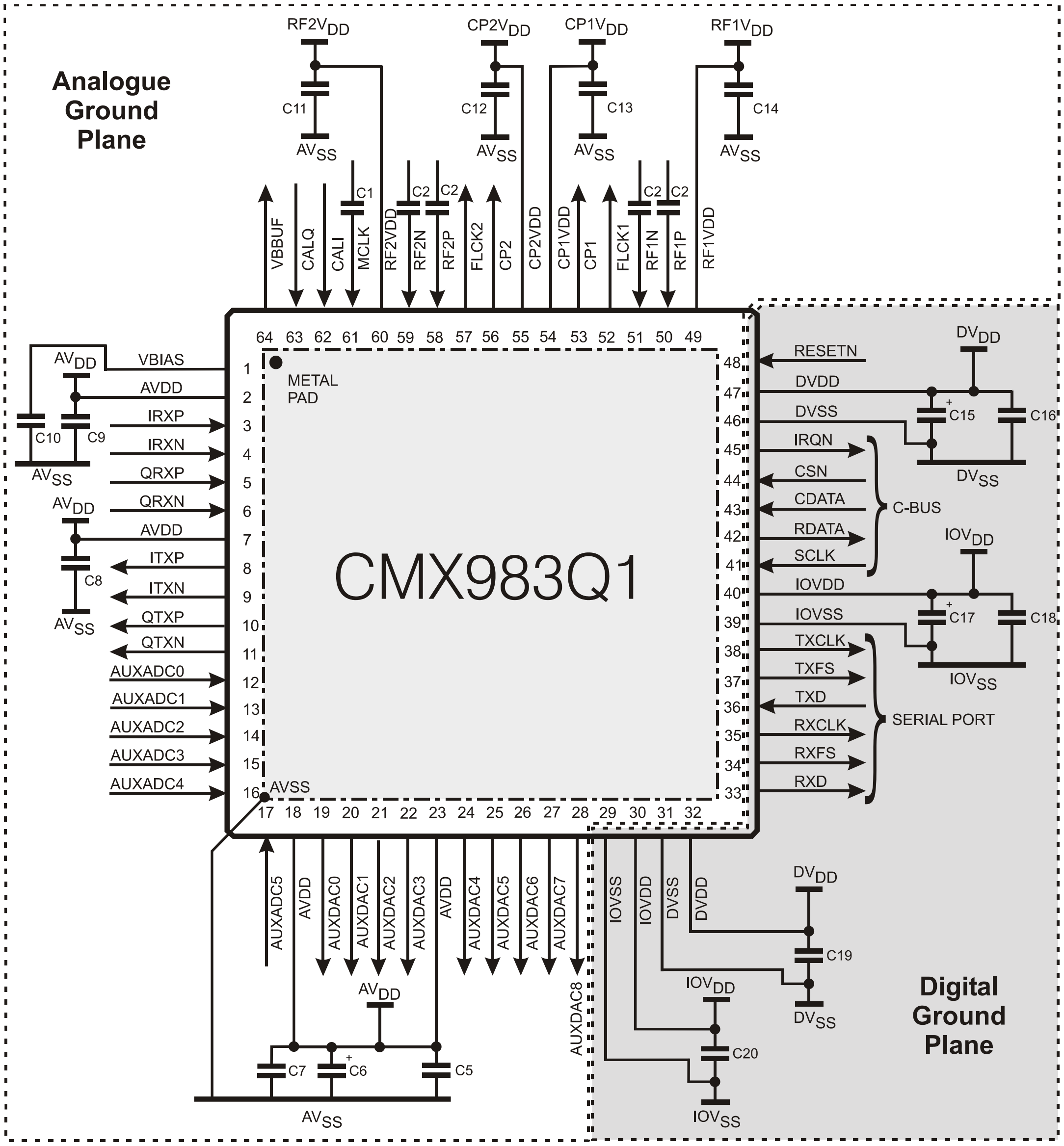


Figure 3 Recommended External Components - General

C1 =	10nF	(Qty = 4) C3 =	1nF	(Qty = 4) R3 =	6.2kΩ
(Qty = 4) C2 =	1nF	(Qty = 4) C4 =	3.9nF	(Qty = 4) R4 =	1.2kΩ
C5, C7, C8, C9 =	10nF	C10 =	100nF	C11, C12, C13, C14 =	10nF
C16, C19 =	10nF	C18, C20 =	10nF	C6, C15, C17 =	10µF

Resistors ±5%, capacitors and inductors ±20%, in Figure 3 to Figure 6 inclusive, unless otherwise stated.

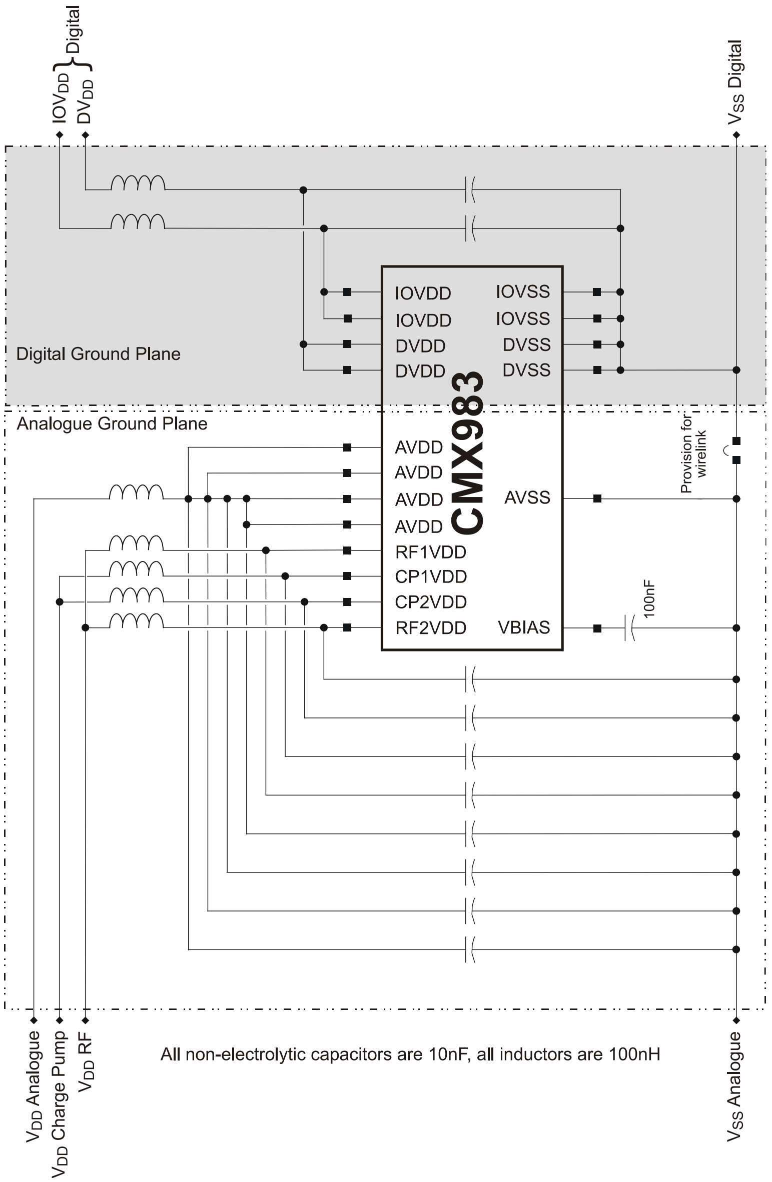


Figure 4 Power Supply Decoupling

To achieve good noise performance, V_{DD} and V_{BIAS} decoupling and protection of the receive path from extraneous in-band signals are very important. It is recommended that the printed circuit board be laid out with ground planes in the CMX983 area to provide a low impedance connection between the VSS pins and the V_{DD} and V_{BIAS} decoupling capacitors. 100nH inductors or 10Ω resistors, in combination with 10nF capacitors, should be used to decouple the power supplies, as shown in Figure 4.

Differential I/Q Inputs and Outputs:

Rx Inputs

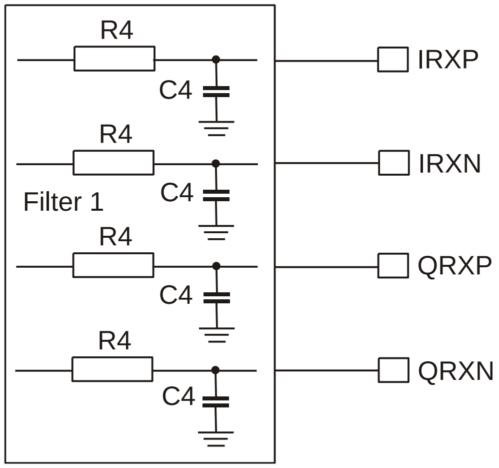


Figure 5 Recommended External Components – Rx Inputs

Example values: R4 = 1.2kΩ, C4 = 3.9nF (R4 x C4 time constant gives -3dB at 34kHz ±10%)

The anti-alias filter stage formed by R4 and C4 should have a low enough cutoff frequency to give adequate attenuation of unwanted signals near the main ADC clock frequency f_{CR1} but also have a high enough cutoff frequency to minimise the group delay variation within the passband in order to prevent intersymbol interference (ISI). If these conflicting requirements cannot be achieved simultaneously the filter should be designed to give adequate attenuation of the unwanted signals near f_{CR1} and compensation for the group delay variation should be applied in the Rx channel FIR filter. Alternatively, a higher order anti-alias filter with a flat group delay response could be used to replace components R4 and C4. In either case, positioning the anti-alias filter components close to the chip inputs may help reduce noise pick-up.

Tx Outputs

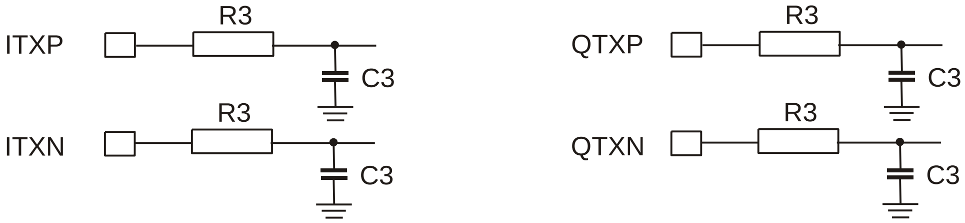


Figure 6 Recommended External Components – Tx Outputs

Example values: R3 = 22kΩ, C3 = 68pF (R3 x C3 time constant should give -3dB at 106kHz ±10%)

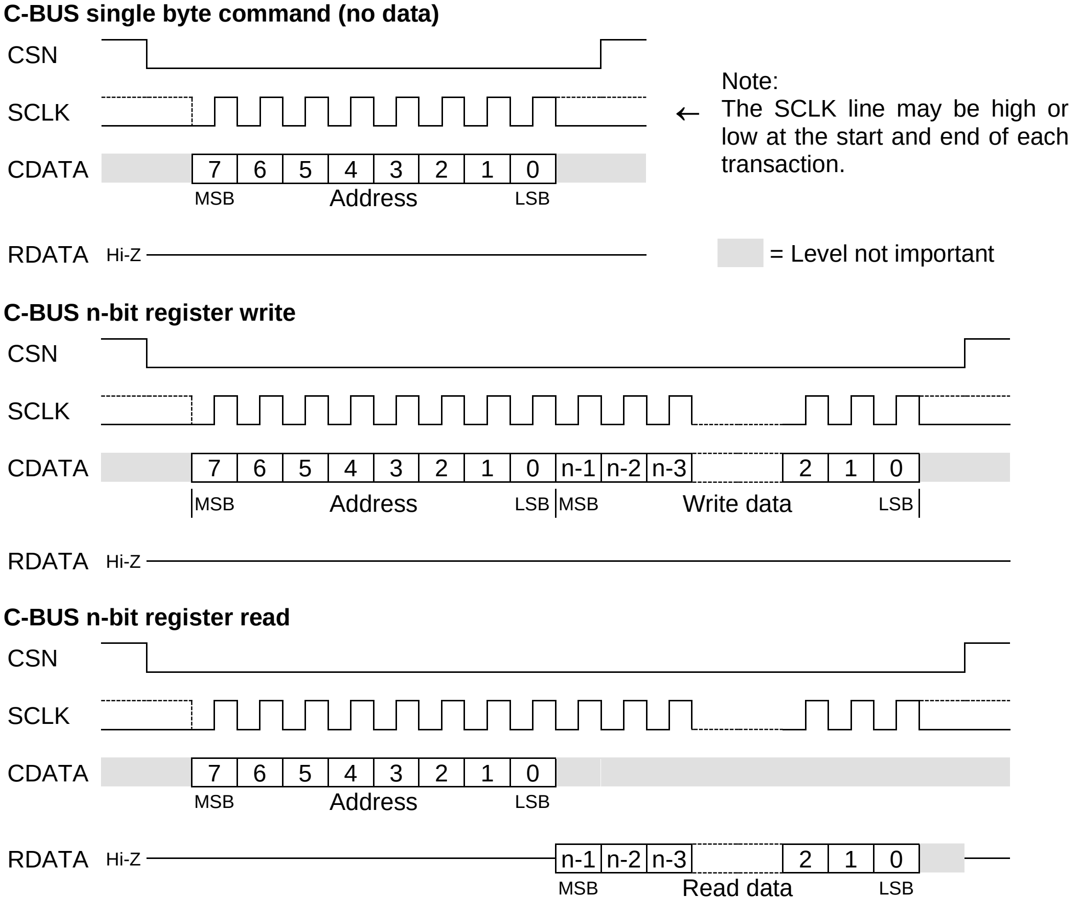
For each transmit channel the RC stage formed by R3 and C3 combines with the internal 2nd-order continuous time filter to create a 3rd-order Bessel filter with a -3dB cutoff frequency of approximately 80kHz.

6 C-BUS Interface

6.1 C-BUS Operation

This block provides for the transfer of data and control or status information between the CMX983 and the host processor over the C-BUS serial bus. Each transaction consists of a single register address byte sent from the host which may be followed by a data word sent from the host to be written into one of the C-BUS's write-only registers, or a data word read out from one of the C-BUS's read-only registers; all C-BUS data words are a multiple of 8 bits wide, the width depending on the source or destination register. Note that certain C-BUS transactions require only an address byte to be sent from the host, no data transfer being required. The operation of the C-BUS is illustrated in Figure 7.

Data sent from the host on the CDATA (command data) line is clocked into the CMX983 on the rising edge of the SCLK input. Data sent from the CMX983 to the host on the RDATA (reply data) line is valid when SCLK is high. The CSN line must be held low during a data transfer and kept high between transfers. The C-BUS interface is compatible with most common µC serial interfaces and may also be easily implemented with general-purpose µC I/O pins controlled by a simple software routine.



C-BUS single byte command (no data)

Note: The SCLK line may be high or low at the start and end of each transaction.

= Level not important

C-BUS n-bit register write

C-BUS n-bit register read

Figure 7 Basic C-BUS Transactions

To maximise data bandwidth across the C-BUS interface, some read and write registers are capable of data-streaming operation. This allows a single address byte to be followed by the transfer of multiple read or write data words, all within the same C-BUS transaction. This can significantly increase the transfer rate of large data blocks, as shown in Figure 8.

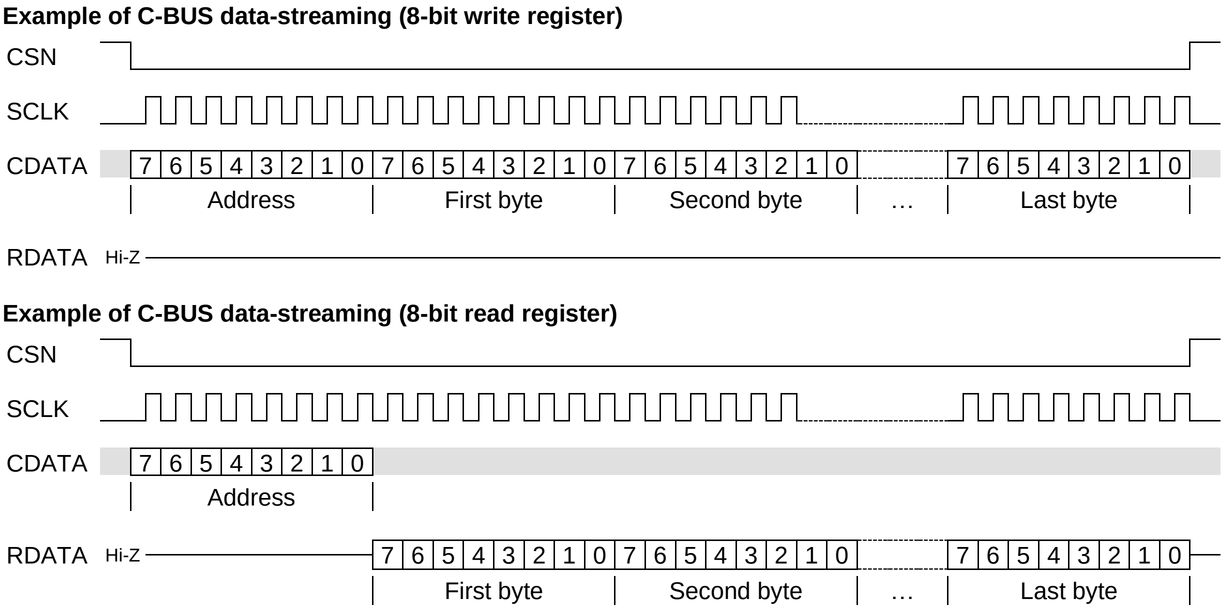


Figure 8 C-BUS Data-streaming Operation

6.2 C-BUS Register Details

A summary of the C-BUS addresses and registers is shown below. After power-up, the CMX983 must be reset using the RESETN pin before the C-BUS can be used. Then, before the internal system clock (CLK) is running, C-BUS accesses are limited to the GENRESET command, the STATUS register, and the clock control registers CLK_CON and CLKPLL_CON0/1. After CLK is running, indicated by STATUS register bit 7 going high, then the rest of the C-BUS registers can be accessed.

C-BUS address	C-BUS type	No. of data bits	Register name ►	C-BUS address	C-BUS type	No. of data bits	Register name ►
$01	Cmd	-	**GENRESET**	$5C	Wr	16	**PLL2_IDIV**
$08	Rd	8	**STATUS**	$5D	Wr	16	**PLL2_FDIV0**
$09	Wr	8	**INT_ENAB**	$5E	Wr	8	**PLL2_FDIV1**
$10	Wr	8	**VBIAS_CON**	$5F	Rd	8	**PLL2_STATUS**
$11	Wr	16	**CLK_CON**	$60	Cmd	-	**AUXADC_START**
$12	Wr	16	**CLKPLL_CON0**	$61	Cmd	-	**AUXADC_ABORT**
$13	Wr	16	**CLKPLL_CON1**	$62	Wr	16	**AUXADC_CLK**
$14	Cmd	-	**CLK_OFF**	$63	Wr	16	**AUXADC_PWRUP**
$15	Cmd	-	**CLK_ON**	$64	Wr	16	**AUXADC_CON**
$1D	Wr	16	**RX_INPUT**	$65	Wr	16	**AUXADC_THR0**
$1E	Wr	16	**RX_OVF**	$66	Wr	16	**AUXADC_THR1**
$1F	Wr	8	**RX_CON0**	$67	Wr	16	**AUXADC_THR2**
$20	Wr	16	**RX_CON1**	$68	Wr	16	**AUXADC_THR3**
$21	Wr	16	**RX_CON2**	$69	Wr	16	**AUXADC_THR4**
$25	Wr	8	**RX_CON3**	$6A	Wr	16	**AUXADC_THR5**
$26	Wr	16	**RX_VERNIER**	$6B	Wr	16	**AUXADC_THR6**
$27	Wr	16	**RX_BITSEL1**	$6C	Wr	16	**AUXADC_THR7**
$28	Wr	16	**RX_BITSEL2**	$6D	Rd	8	**AUXADC_STAT**
$29	Wr	16 $^{(DS)}$	**RX_COEFF0**	$6E	Rd	16	**AUXADC_DATA0**
$2A	Wr	16 $^{(DS)}$	**RX_COEFF1**	$6F	Rd	16	**AUXADC_DATA1**
$2B	Wr	16 $^{(DS)}$	**RX_COEFF2**	$70	Rd	16	**AUXADC_DATA2**
$2C	Wr	16 $^{(DS)}$	**RX_COEFF3**	$71	Rd	16	**AUXADC_DATA3**
$2D	Wr	8	**RX_ADDR**	$72	Rd	16	**AUXADC_DATA4**
$2E	Rd	16	**RX_STATUS**	$73	Rd	16	**AUXADC_DATA5**
$2F	Wr	16	**RX_ST_ENAB**	$74	Rd	16	**AUXADC_DATA6**
$30	Wr	8	**TX_CON0**	$75	Rd	16	**AUXADC_DATA7**
$31	Wr	16	**TX_CON1**	$76	Wr	8	**AUXCMP_CON0**
$34	Wr	16	**TX_CON2**	$77	Wr	8	**AUXCMP_CON1**
$35	Wr	16	**TX_GAIN**	$78	Wr	8	**AUXCMP_CON2**
$36	Wr	16 $^{(DS)}$	**TX_COEFF0**	$79	Wr	8	**AUXCMP_CON3**
$37	Wr	16 $^{(DS)}$	**TX_COEFF1**	$7A	Wr	8	**AUXCMP_CON4**
$38	Wr	8	**TX_ADDR**	$7B	Rd	8	**AUXCMP_STAT**
$39	Rd	8	**TX_STATUS**	$7C	Wr	8	**AUXCMP_ST_EN**

C-BUS address	C-BUS type	No. of data bits	Register name ►	C-BUS address	C-BUS type	No. of data bits	Register name ►
$3A	Wr	8	**TX_ST_ENAB**	$82	Wr	16	**AUXDAC_CLK**
$40	Wr	8	**RXPORT_CON0**	$83	Wr	16 (DS)	**AUXDAC_RAMD**
$41	Wr	8	**RXPORT_CON1**	$84	Wr	8	**AUXDAC_RAMA**
$48	Wr	8	**TXPORT_CON0**	$85	Cmd	-	**AUXDAC_UP**
$49	Wr	8	**TXPORT_CON1**	$86	Cmd	-	**AUXDAC_DOWN**
$4E	Wr	16	**PLL1_CON**	$87	Cmd	-	**AUXDAC_CYCLE**
$4F	Wr	16	**PLL1_LOCKDET**	$88	Cmd	-	**AUXDAC_RST**
$50	Wr	16	**PLL1_FLCK**	$89	Wr	16	**AUXDAC_DATA0**
$51	W	8	**PLL1_BLEED**	$8A	Wr	16	**AUXDAC_DATA1**
$52	Wr	8	**PLL1_RDIV**	$8B	Wr	16	**AUXDAC_DATA2**
$53	Wr	16	**PLL1_IDIV**	$8C	Wr	16	**AUXDAC_DATA3**
$54	Wr	16	**PLL1_FDIV0**	$8D	Wr	16	**AUXDAC_DATA4**
$55	Wr	8	**PLL1_FDIV1**	$8E	Wr	16	**AUXDAC_DATA5**
$56	Rd	8	**PLL1_STATUS**	$8F	Wr	16	**AUXDAC_DATA6**
$57	Wr	16	**PLL2_CON**	$90	Wr	16	**AUXDAC_DATA7**
$58	Wr	16	**PLL2_LOCKDET**	$91	Wr	16	**AUXDAC_DATA8**
$59	Wr	16	**PLL2_FLCK**	$CB	Wr	16	**RX_BIAS0**
$5A	W	8	**PLL2_BLEED**	$CC	Wr	16	**RX_BIAS1**
$5B	Wr	8	**PLL2_RDIV**	$CE	Wr	16	**PLL_CFG**

(DS) – These registers are capable of data-streaming transactions.
Note: C-BUS locations not defined in the above table are reserved and must not be written to.

Table 3 C-BUS Register Map

6.3 C-BUS General Reset

The CMX983 can be reset using the C-BUS GENRESET command – this has the same result as resetting the device using the RESETN pin. After reset is applied, the CMX983 will stay in a quiescent state until the system clock generator is programmed (see section 8).

GENRESET - $01

C-BUS command, no data required

When the GENRESET command is sent from the host processor, the internal device reset is applied on the eighth rising edge of the SCLK pin and is released on the subsequent rising edge of CSN.

6.4 C-BUS Status and Interrupt

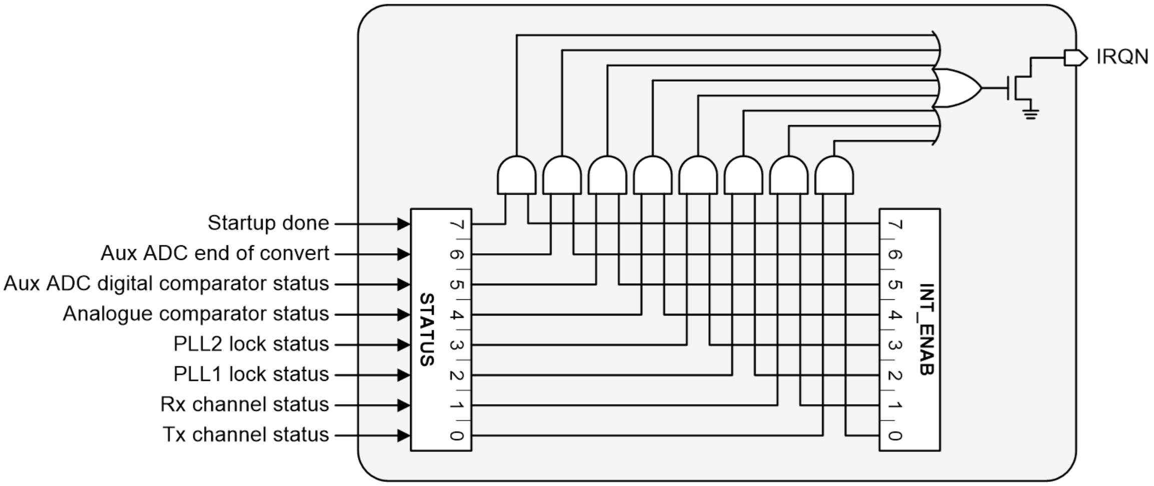


Figure 9 C-BUS Status and Interrupt

The C-BUS STATUS register, shown in Figure 9, is a read-only register that contains the status of various circuits within the CMX983. The STATUS register can be polled by the host processor, or it can be interrupt driven: the interrupt pin IRQN will be asserted when any bit of the STATUS register is set to 1 and the associated bit in the INT_ENAB register is also set to 1. Enabling an interrupt by setting an INT_ENAB bit (0→1) after the corresponding STATUS register bit has already been set to 1 will also cause the IRQN output to be asserted. The IRQN pin is an active low open-drain output.

STATUS - $08: 8-bit Read
Reset value = $00

Bit:

7	6	5	4	3	2	1	0
Start-up done	Aux ADC end of conv	Aux ADC digital comp status	Ana-logue comp status	PLL2 lock status	PLL1 lock status	Rx chan status	Tx chan status

STATUS b7: Startup done
This status bit gets set to 1, if enabled, when the startup timer in the system clock generator has reached its endcount value. This indicates that it is safe to access the other C-BUS registers within the CMX983. This bit gets automatically cleared to 0 when it is read.

STATUS b6: Aux ADC end of convert
This bit gets set to 1 when an Aux ADC convert sequence completes, and automatically gets cleared to 0 when it is read.

STATUS b5: Aux ADC digital comparator status
This bit gets set to 1 if any of the bits in the AUXADC_STAT register are set to 1, indicating that one or more ADC conversion results were within the programmed threshold range. To clear this status bit, the AUXADC_STAT register must be read.

STATUS b4: Analogue comparator status

This bit gets set to 1 if any of the bits in the AUXCMP_STAT register are set to 1 (indicating that an analogue comparator input has crossed its threshold) as long as the associated bit in the AUXCMP_ST_EN register is also set to 1. This status bit can be cleared either by reading the AUXCMP_STAT register or by clearing the associated enable bit(s) in the AUXCMP_ST_EN register.

STATUS b3: PLL2 lock status
This bit indicates the PLL2 lock status. It can only be cleared by clearing PLL2_STATUS bits 1-0 (see section 12 for details).

STATUS b2: PLL1 lock status
This bit indicates the PLL1 lock status It can only be cleared by clearing PLL1_STATUS bits 1-0 (see section 12 for details).

STATUS b1: Rx channel status
This bit gets set to 1 if any of the bits in the RX_STATUS register are set to 1 and the associated bit in the RX_ST_EN register is also set to 1. For details about clearing this bit, see section 0.

STATUS b0: Tx channel status
This bit gets set to 1 if any of the bits in the TX_STATUS register are set to 1 and the associated bit in the TX_ST_EN register is also set to 1. For details about clearing this bit, see section 10.

INT_ENAB - $09: 8-bit Write
Reset value = $80

Bit:	7	6	5	4	3	2	1	0
	Interrupt enable							

INT_ENAB b7-0: Interrupt enable
Setting any of these bits to 1 enables the corresponding bit in the STATUS register to generate an interrupt. This will cause the active-low open-drain IRQN pin to pull down. After a reset, bit 7 of this register will be high which allows a “startup done” interrupt to be generated.

7 Bias Generator

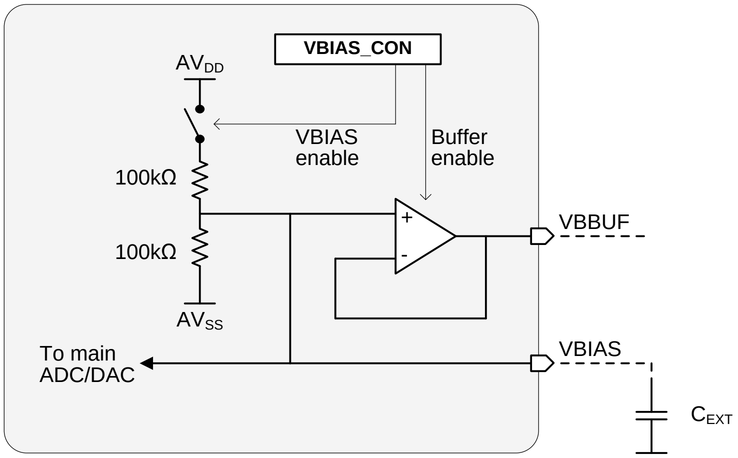


Figure 10 Bias Voltage Generator

The bias generator provides a mid-rail reference voltage (AV$_{DD}$/2) that is used by the main ADCs and DACs. An external decoupling capacitor is required on the VBIAS pin; no other connections should be made to this pin. A buffered version of the bias voltage, available on the VBBUF pin, can be used to drive external circuitry.

VBIAS_CON - $10: 8-bit Write
Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	VBIAS enab	Buffer enab

VBIAS_CON b7-2: Reserved, set to 0

VBIAS_CON b1: VBIAS enable
Set to 1 to enable the mid-rail VBIAS generator – this must be done before using the main ADC or DAC channels. Set to 0 to disable and powersave the VBIAS generator. Note that the V$_{BIAS}$ voltage takes some time to settle, determined by the effective 50kΩ source impedance and the value of the external capacitor C$_{EXT}$.

VBIAS_CON b0: Buffer enable
Set to 1 to enable the VBIAS buffer amplifier. Set to 0 to disable and powersave the VBIAS buffer. When disabled, the VBBUF pin will go high impedance.

8 System Clock Generator

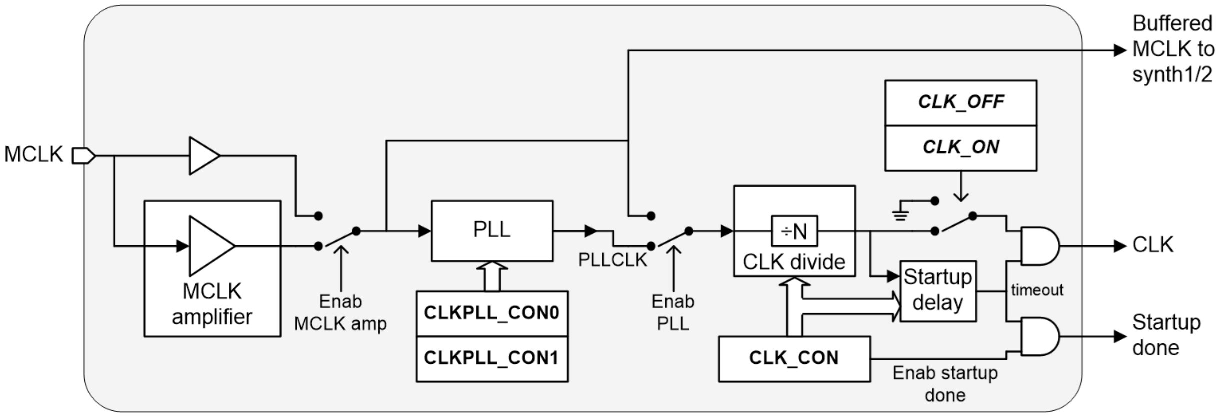


Figure 11 System Clock Generator

The system clock generator buffers the MCLK input signal and generates the internal clocks required by the rest of the CMX983 circuitry. The MCLK pin can optionally be driven by a full swing logic level, or by the low amplitude sinewave or clipped sinewave that is typically produced by an external precision oscillator module. In the latter case, an internal low phase noise amplifier is used to convert the MCLK input signal to a full logic level.

The buffered MCLK signal is used to directly drive the reference dividers in the two fractional-N synthesizers. The main system clock signal CLK is generated either by dividing down the MCLK signal, or by dividing down an internally generated PLL clock signal (if a non-integer related frequency is required).

When the CMX983 comes out of reset the CLK signal is disabled, which prevents all C-BUS accesses except to the three system clock control registers (CLK_CON and CLKPLL_CON0/1), the STATUS register and the GENRESET command. The clock control registers must be configured directly after the CMX983 comes out of reset in order to start the CLK signal. If using the PLL, then CLKPLL_CON0 and CLKPLL_CON1 must be written first. Then when CLK_CON is written, the system clock generator powers up and begins operating. As part of the power up sequence, a startup delay timer ensures that the internal CLK signal is kept inactive until a programmable number of clock pulses have been generated by the clock divider; this gives the MCLK amplifier bias circuit and PLL time to stabilise. A “startup done” status bit can optionally be generated to indicate that the startup delay timer has expired and that the CLK signal is active. Note that once the CLK_CON register has been written, all further changes to the three clock control registers are disabled until the CMX983 is reset again.

CLK_CON - $11: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	Enab startup done	Startup delay					Clock divide							Enab MCLK amp

CLK_CON b15-14: Reserved, set to 0

CLK_CON b13: Enable startup done
Set this bit to 1 to enable the “startup done” status bit. This status bit appears in the main STATUS register, and indicates to the host processor that the startup delay counter has timed out and the internal clock CLK is running.

CLK_CON b12-8: Startup delay
This value determines how long the system clock generator circuits are allowed to stabilise before the internal CLK signal is enabled. The delay time should be sufficient to allow the bias for the AC-coupled MCLK amplifier to settle and for the clock PLL to achieve lock, assuming these circuits are enabled. The startup delay counter is triggered immediately when CLK_CON is written, and counts output clock pulses from the clock divider. When the startup delay counter reaches its programmed endcount, the CLK signal is activated and, if CLK_CON bit 13 = 1, a startup interrupt is generated.

Startup delay	
$00	Delay (cycles) = 2^0 (1)
$01	Delay (cycles) = 2^1 (2)
$02	Delay (cycles) = 2^2 (4)
$03	Delay (cycles) = 2^3 (8)
…	…
$13	Delay (cycles) = 2^{19} (524288)
$14	Delay (cycles) = 2^{20} (1048576)
$15 - $1F	Illegal, do not use

CLK_CON b7-1: Clock divide
Sets the division ratio between MCLK (or PLLCLK) and the system clock CLK. This value can be set to between 1 and 128 (0000000 = 128).

CLK_CON b0: Enable MCLK amplifier
Set to 1 to enable the low-noise MCLK amplifier, for use when the MCLK signal is a low-amplitude sinewave or clipped sinewave. Set to 0 to disable the MCLK amplifier, for use when MCLK is a full swing logic level.

Clock PLL registers
Do not write to these registers if the clock PLL is not being used.

CLKPLL_CON0 - $12: 16-bit Write
Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CLKPLL_CON1 - $13: 16-bit Write
Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

For assistance in using the clock PLL, please contact the Technical Support Team at CML.

CLK_OFF - $14
C-BUS command, no data required

The CLK_OFF command causes the internal system clock CLK to stop, but leaves the MCLK amplifier and PLL (if either are being used) enabled. This command can be used during periods of device inactivity to save power, but allows CLK to be rapidly restarted using the CLK_ON command. After a CLK_OFF command all C-BUS accesses are prevented except for the GENRESET and CLK_ON commands.

CLK_ON - $15
C-BUS command, no data required

The CLK_ON command immediately starts the internal system clock CLK running after previously being stopped by the CLK_OFF command.

9 Receive Channel

9.1 Rx Signal Routing and ADCs

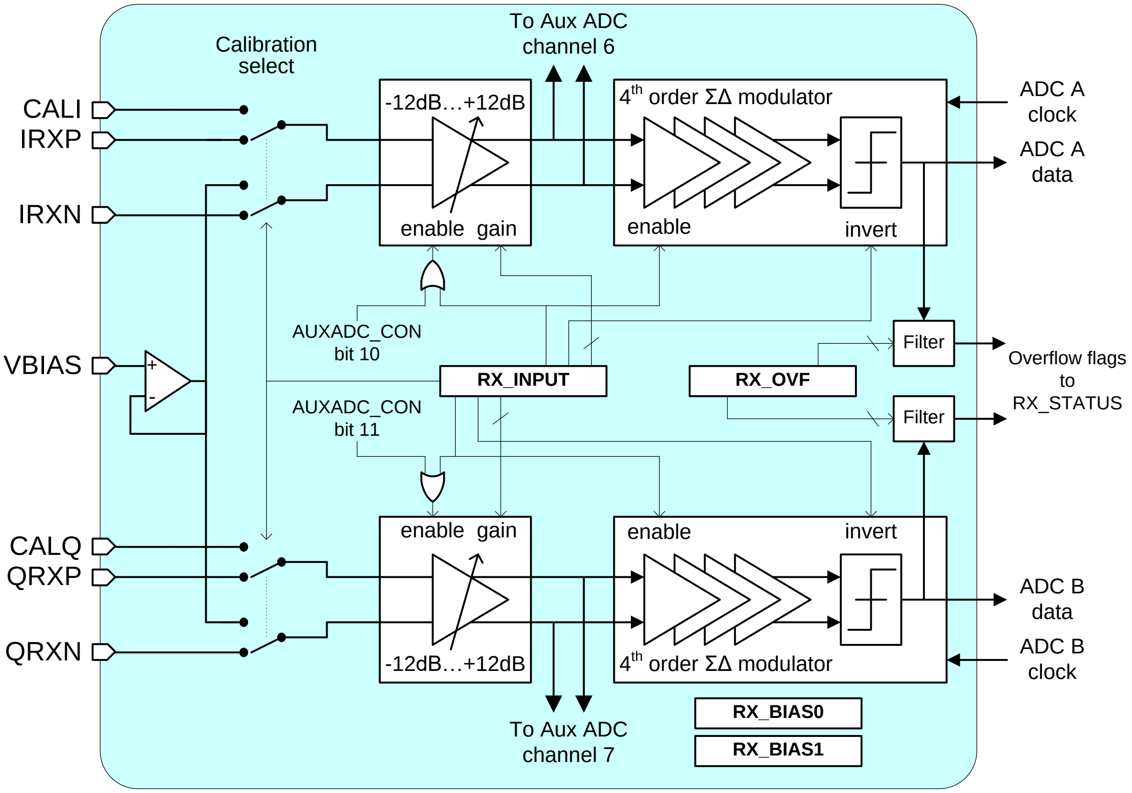


Figure 12 Rx Input Switching

The I/Q inputs connect to the main ADCs through a signal switching block and programmable gain amplifiers as shown in Figure 12. The calibration inputs can be connected to the programmable gain amplifiers to assist with system setup. Each ADC is a fourth-order sigma-delta type that outputs a single-bit pulse density modulated bitstream to the following digital channel filters. Status bits are produced that indicate an input overload; these can be read from the RX_STATUS register (section 0).

Each input requires an external anti-alias filter, although the high oversampling rate typically used by the sigma-delta modulator relaxes the design requirements of these filters. To achieve optimum performance, signals at the sampling frequency (typically around 2.4MHz) should be attenuated to -110dB or lower. Additionally, in order to reduce the complexity of the digital channel filters, the anti-alias filter may be able to usefully suppress signals at the first decimation rate.

The outputs of the Rx A and Rx B programmable gain amplifiers are also connected to channels 6 and 7 of the Auxiliary ADC through differential-to-single-ended converters (see section 13). Enabling either differential-to-single-ended converter in the Auxiliary ADC automatically enables the associated Rx A or Rx B programmable gain amplifier, without also enabling that channel’s sigma-delta modulator. This feature can be used to implement a low power input signal level monitor during periods when the main Rx signal path is not active.

The following C-BUS registers control the signal routing, gain setting and overflow status of the two Rx input circuits:

RX_INPUT - $1D: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cal select	Rx B enable	ADC B invert	Rx B gain (-12dB … 12dB, +mute)					0	Rx A enable	ADC A invert	Rx A gain (-12dB … 12dB, +mute)				

RX_INPUT b15: Calibration select
When this bit is set to 1 the calibration pin CALI is connected to channel A positive input, CALQ is connected to channel B positive input, and the mid-rail reference voltage V_{BIAS} is buffered and applied to both the channel A and B negative inputs. Also, the Rx A and Rx B programmable gain amplifiers are automatically set to 0 dB while the calibration select bit is set to 1.

RX_INPUT b14: Rx B enable
Set to 1 to enable channel B gain stage and ADC modulator.

RX_INPUT b13: ADC B invert
Set to 1 to invert the channel B modulator output.

RX_INPUT b12-8: Rx B gain
Set channel B input gain to between -12dB and +12dB in 1dB steps, or mute the output. The gain setting is in 2’s complement format:

```
10000 = mute
10100 = -12.0dB
10101 = -11.0dB
      ...
11111 = -1.0dB
00000 = 0.0dB
00001 = 1.0dB
      ...
01011 = 11.0dB
01100 = 12.0dB
```

RX_INPUT b7: Reserved, set to 0

RX_INPUT b6: Rx A enable
Set to 1 to enable channel A gain stage and ADC modulator.

RX_INPUT b5: ADC A invert
Set to 1 to invert the channel A modulator output.

RX_INPUT 4-0: Rx A gain
Set channel A input gain, similar in operation to bits 12-8.

RX_OVF - $1E: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	Filt length B	Enab i/p B overflow	Overflow threshold B					0	Filt length A	Enab i/p A overflow	Overflow threshold A				

The input overflow circuit uses a separate running average filter for each of the modulator output bitstreams (channels A and B) using a selectable window width of 32 or 64 bits. The modulator output bits have a weighting of ± ½, so the maximum theoretical filter output value is ± 16 (with a window width of 32) or ± 32 (with a window width of 64). The *average* filter output value, for a dc level of 1V pk-pk at the input of the sigma-delta modulator, is approximately 4.82 (window width = 32) or 9.64 (window width = 64). If at any time the filter output for a channel exceeds the specified threshold value (in either the positive or negative direction), and if the overflow flag is enabled, then the associated status bit in the RX_STATUS register gets set to 1.

RX_OVF b15: Reserved, set to 0

RX_OVF b14: Filter length B
For channel B, set to 1 for a running average filter length of 64, set to 0 for a filter length of 32.

RX_OVF b13: Enable input overflow B
For channel B, set to 1 to enable the filters, set to 0 to disable the filters.

RX_OVF b12-8: Overflow threshold B
If the absolute value of the (enabled) channel B running average filter output is greater than the overflow threshold B value, then an overflow is flagged in the RX_STATUS register.

RX_OVF b7: Reserved, set to 0

RX_OVF b6: Filter length A
For channel A, set to 1 for a running average filter length of 64, set to 0 for a filter length of 32.

RX_OVF b5: Enable input overflow A
For channel A, set to 1 to enable the filters, set to 0 to disable the filters.

RX_OVF b4-0: Overflow threshold A
If the absolute value of the (enabled) channel A running average filter output is greater than the overflow threshold A value, then an overflow is flagged in the RX_STATUS register.

RX_BIAS0 - $CB: 16-bit Write
Reset value = $9333

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Bias 0 value															

RX_BIAS0 is described with RX_BIAS1, below.

RX_BIAS1 - $CC: 16-bit Write
Reset value = $004F

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved, set to 0								Bias 1 value							

Together the RX_BIAS0 and RX_BIAS1 registers control main ADC bias current to support normal or 2x clock rate operation that respectively support Normal or Wide (2x) Bandwidth Modes. For Normal Bandwidth Mode respectively set RX_BIAS0 and RX_BIAS1 to $9333 and $004F. For Wide Bandwidth Mode respectively set RX_BIAS0 and RX_BIAS1 to $6BBB and $00EE.

9.2 Receive Channel Filters

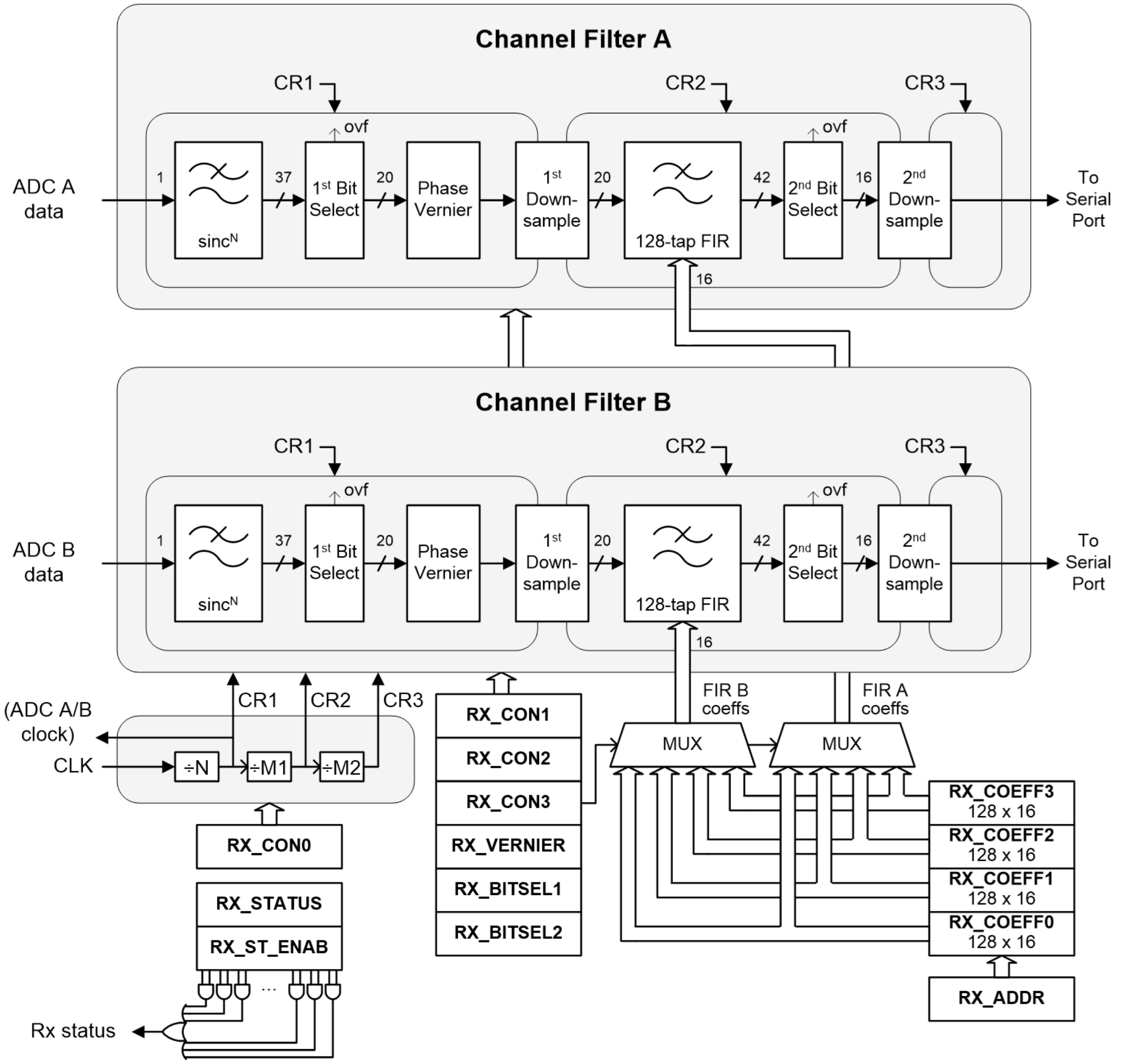


Figure 13 Rx Channel Filters

The CMX983 has two main channel filters (Figure 13). Within each channel filter, the signal is decimated twice. Data from the ADC passes into a sinc^N filter, a bit selector circuit and a vernier phase adjustment before the first downsampler. The resulting signal then passes into a programmable 128-tap FIR filter, a second bit selector and the second downsampler before being passed to the serial port for transmission to the host processor. The serial port data is formatted as 16-bit 2's complement values. Each stage in the Rx channel (sinc filter, bit select, vernier adjust, downsample and FIR filter) is programmable.

Sinc filter

The bitstream produced by each sigma-delta modulator has a characteristic high-pass filtered noise profile. The sinc filter attenuates this quantisation noise, along with any other noise or unwanted signals coming from the device inputs, to prevent aliasing problems in the first downsampler. The number of cascaded stages in the sinc filter can be configured to between 3 and 6, with each stage having a length of up to 64. The length of the sinc filter should be set to an integer multiple (usually 1x) of the first downsample rate M1 so that the zeroes in the filter transfer function appear at multiples of the CR2 clock rate. This minimises the amount of in-band energy in the aliased signals after decimation. Note that the sinc filter transfer characteristic causes droop in the wanted signal, and this droop increases as the number of sinc stages or the sinc length is increased. However, moderate amounts of droop can be compensated for in the following FIR filter. With a nominal power supply voltage (AV_{DD} = 3.3V), the "gain" of the ADC between the input pins and the output of the sinc filter is given by the expression:

$$G = 0.1506\,AL^{N}$$

where L = sinc length, N = number of sinc stages and A = analogue gain (see section 9.1). For example, with an analogue gain of 0dB, a sinc length of 32 and a sinc number of 5, then a dc input signal of 1V (differential) at the input pins will give a nominal output of $5.053\text{x}10^{6}$ (~ $4D1B00) into the first bit selector.

First bit selector

The bit selector at the output of the sinc filter selects which 20 bits of the 37-bit sinc filter accumulator are passed to the following phase vernier and downsampler.

Phase vernier

The phase vernier allows fine adjustment of the signal phase by setting which sinc output sample the first decimator selects. The A and B channel signals can be independently delayed by a programmable number of cycles of the CR1 clock, with the maximum number of delay cycles being one less than the first downsample rate.

First downsampler

The first downsampler reduces the sample rate of the signal by a factor of M1 = f_{CR1}/f_{CR2}. This process causes any residual signal components around multiples of the CR2 clock rate to be aliased, so it is important to make sure that the preceding filters (sinc filter and external anti-alias filter) have adequately attenuated these signal components.

FIR filter

The purpose of the FIR filter is to attenuate out-of band signals and quantisation noise, perform any transfer function shaping that is required by the transmission standard and, if necessary, compensate for the droop caused by the sinc filter. The FIR filter acts as an anti-alias filter for the second downsampler (if used) by ensuring that signal components around multiples of the CR3 frequency are adequately attenuated. The filter operates with 20-bit data samples and 16-bit coefficients, and can be configured with up to 128 taps. There are four banks of programmable coefficients.

Second bit selector

The bit selector at the output of the FIR filter selects which 16 bits of the 42-bit FIR filter accumulator are passed to the following downsampler.

Second downsampler

The second downsampler reduces the sample rate of the signal by a factor of M2 = f_{CR2}/f_{CR3}. This process causes any residual signal components around multiples of the CR3 clock rate to be aliased, so it is important to make sure that the preceding filters have adequately attenuated these signal components.

The following C-BUS registers are used to configure the Rx channels. The settings in RX_CON0/1/2 are applied to both channel filter A and channel filter B:

RX_CON0 - $1F: 8-bit Write
Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	0	ADC clock divide						

RX_CON0 b7: Reserved, set to 0

RX_CON0 b6-0: ADC clock divide
Sets the division ratio between CLK and CR1, where CR1 is the clock for the sigma-delta modulator and sinc filter. This value can be set to between 2 and 128 (0000000 = 128).

RX_CON1 - $20: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	Sinc number		Sinc length						First downsample rate					

RX_CON1 b15-14: Reserved, set to 0

RX_CON1 b13-12: Sinc number
These bits determine the number of cascade stages in the sinc filter: 00 = 3 stages (sinc^3); 01 = 4 stages (sinc^4); 10 = 5 stages (sinc^5); 11 = 6 stages (sinc^6).

RX_CON1 b11-6: Sinc length
Sets the length of each sinc filter section to between 1 and 64 (000000 = 64). This is normally set to the same value as the first downsample rate.

RX_CON1 b5-0: First downsample rate
Sets the division ratio between the CR1 and CR2 clock, which determines the first downsample rate M1. This can be set to between 1 and 64 (000000 = 64).

RX_CON2 - $21: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	FIR bypass	FIR filter length							Second downsample rate					

RX_CON2 b15-14: Reserved, set to 0

RX_CON2 b13: FIR bypass
Set this bit to 1 to disable and bypass the FIR filter. Data from the output of the first downsampler will then be sent directly to the input of the second bit selector (in the 20 most-significant bit positions). When the FIR bypass bit is changed from 0 to 1, an initialisation sequence is performed which resets all data samples in FIR filter to zero (coefficient RAMs are not altered). This initialisation takes 128 CLK cycles to complete.

RX_CON2 b12-6: FIR filter length

Sets the number of taps in the FIR filter to a value between 1 and 128 (0000000 = 128). The filter length is also subject to the following restriction, based on the CLK and ADC sample frequencies:

$$FilterLength \leq \left(\frac{f_{CLK} - f_{CR2}}{f_{CR3}} - 1 \right)$$

RX_CON2 b5-0: Second downsample rate

Sets the division ratio between the CR2 and CR3 clock, which determines the second downsample rate M2. This can be set to between 1 and 64 (000000 = 64). Note that the maximum CR3 clock frequency is further restricted to $f_{CR3} \leq f_{CLK}/4$, i.e. (N × M1 × M2) ≥ 4, although this limit is unlikely to be approached in a typical receive channel configuration.

RX_CON3 - $25: 8-bit Write

Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	0	Rx B enable	FIR B coeff select		0	Rx A enable	FIR A coeff select	

RX_CON3 b7: Reserved, set to 0

RX_CON3 b6: Rx B enable

Set this bit to 1 to enable Rx channel B filter logic (sinc filter onwards) and start the transfer of data to the serial port. When this bit is set to 1, the “channel B idle” bit in RX_STATUS will read as 0. After reset, or when the Rx B enable bit is changed from 1 to 0, the channel B logic ceases operating and an initialisation sequence is performed which resets all data samples in the sinc filter and FIR filter to zero (coefficient RAMs are not altered). This initialisation takes 128 CLK cycles to complete, after which the “channel B idle” bit in RX_STATUS will be set to 1.

It is recommended that the Rx B enable bit should be set to 1 only after the configuration bits in RX_CON0/1/2 have been initialised, and that these registers do not get changed again until the Rx B enable bit is cleared to 0 and channel B is idle.

RX_CON3 b5-4: FIR B coefficient select

Selects which bank of coefficients the channel B FIR filter uses:

00 = RX_COEFF0
01 = RX_COEFF1
10 = RX_COEFF2
11 = RX_COEFF3

RX_CON3 b3: Reserved, set to 0

RX_CON3 b2: Rx A enable

Set this bit to 1 to enable Rx channel A filter logic, similar in operation to bit 6.

RX_CON3 b1-0: FIR A coefficient select

Selects which bank of coefficients the channel A FIR filter uses:

00 = RX_COEFF0
01 = RX_COEFF1
10 = RX_COEFF2

11 = RX_COEFF3

RX_VERNIER - $26: 16-bit Write
Reset value = $0000

Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	Phase vernier B						0	0	Phase vernier A					

RX_VERNIER b15-14: Reserved, set to 0

RX_VERNIER b13-8: Phase vernier B
Allows fine adjustment of the channel B signal phase by setting which sinc output sample the first downsampler selects. Setting the phase vernier value to 0 gives an 'on-time' signal; setting the value to N gives a 'late' signal, delayed by N cycles of the CR1 clock. The phase vernier value must be less than the value set for the first downsample rate in the associated decimator control register RX_CON1.

RX_VERNIER b7-6: Reserved, set to 0

RX_VERNIER b5-0: Phase vernier A
Allows fine adjustment of the channel A signal phase (similar in operation to bits 13-8).

RX_BITSEL1 - $27: 16-bit Write
Reset value = $0000

Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	Rx B first bit select					0	0	0	Rx A first bit select				

RX_BITSEL1 b15-13: Reserved, set to 0

RX_BITSEL1 b12-8: Rx B first bit select
In Rx channel B, selects which 20 bits of the 37-bit sinc filter accumulator are passed to the following phase vernier and downsample stage. This value determines the number of most-significant bits discarded (valid range = 0 to 17): a value of 0 selects the most significant 20 bits of the sinc accumulator, a value of 1 discards the MSB of the accumulator and selects the next most significant 20 bits, and so on. Convergent rounding is applied to the selected bits and the selector output saturates in the case of an overflow (positive saturation = $7FFFF, negative saturation = $80000). To assist with setup, an overflow causes a status bit to be set in the RX_STATUS register.

RX_BITSEL1 b7-5: Reserved, set to 0

RX_BITSEL1 b4-0: Rx A first bit select
Rx channel A first bit select, similar in operation to bits 12-8.

RX_BITSEL2 - $28: 16-bit Write
Reset value = $0000

Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	Rx B second bit select					0	0	0	Rx A second bit select				

RX_BITSEL2 b15-13: Reserved, set to 0

RX_BITSEL2 b12-8: Rx B second bit select

In Rx channel B, selects which 16 bits of the 42-bit FIR filter accumulator are passed to the following downsample stage. This value determines the number of most-significant bits discarded (the valid range = 0 to 26): a value of 0 selects the most significant 16 bits of the FIR accumulator, a value of 1 discards the MSB of the accumulator and selects the next most significant 16 bits, and so on. Convergent rounding is applied to the selected bits and the selector output saturates in the case of an overflow (positive saturation = $7FFF, negative saturation = $8000). To assist with setup, an overflow causes a status bit to be set in the RX_STATUS register.

RX_BITSEL2 b7-5: Reserved, set to 0

RX_BITSEL2 b4-0: Rx A second bit select

Rx channel A second bit select, similar in operation to bits 12-8.

RX_COEFF0 - $29: 16-bit Write, data-streaming
RX_COEFF1 - $2A: 16-bit Write, data-streaming
RX_COEFF2 - $2B: 16-bit Write, data-streaming
RX_COEFF3 - $2C: 16-bit Write, data-streaming

Reset value = undefined

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIR coefficient values (2's complement)															

These four C-BUS locations allow the 128-word FIR coefficient RAMs for the Rx channels to be loaded. Each coefficient RAM can be loaded in ascending order by repeatedly writing data to the same C-BUS location (an internal address pointer automatically increments after each write). To increase the loading rate of the coefficients, data-streaming operation is supported for these four C-BUS addresses. Only as many coefficients as are required (determined by the FIR length) need to be loaded into each RAM, unused RAM locations do not need to be written.

Before loading each of the four coefficient RAMs, the internal address pointer needs to be initialised (usually to address 0); this is done by writing to register RX_ADDR. Note that all four RAMs share this address pointer, so the address needs to be initialised before each coefficient RAM is loaded.

During operation, with a filter length of N, the FIR filter stores the previous N data samples provided by the first downsampler. Then, whenever the second downsampler requires a new sample, the FIR filter generates this by performing a sequence of N multiply/accumulate operations using the selected filter coefficients and the stored data samples. The coefficient at RAM address 0 is multiplied by the most recent data sample from the first downsampler, the coefficient at address 1 is multiplied by the data delayed by one cycle of CR2, the coefficient at address 2 is multiplied by the data delayed by two cycles of CR2, and so on. The accumulated total, after scaling/rounding and downsampling, is sent to the serial port for transmission.

RX_ADDR - $2D: 8-bit Write

Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	0	Rx coefficient address pointer						

The Rx coefficient address pointer determines the address at which data gets written during a C-BUS write to any of the four Rx coefficient RAMs. The Rx coefficient address pointer automatically increments after each 16-bit coefficient value is written, so if the coefficients are written in an ascending sequence the pointer only needs to be initialised once before each bank of coefficients is loaded.

RX_STATUS - $2E: 16-bit Read
Reset value = $0101

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	Ch. B vern. error	Ch. B FIR error	Ch. B 2nd bit select over-flow	Ch. B 1st bit select over-flow	Ch. B input over-flow	Ch. B buffer over-run	Ch. B idle	0	Ch. A vern. error	Ch. A FIR error	Ch. A 2nd bit select over-flow	Ch. A 1st bit select over-flow	Ch. A input over-flow	Ch. A buffer over-run	Ch. A idle

RX_STATUS b15: Reserved, set to 0

RX_STATUS b14: Channel B vernier error
This bit gets set if Rx channel B is enabled and the vernier value is incorrect, i.e. it has been set to a value equal to or greater than first downsample rate in channel B. To clear this bit, first correct the error (or disable the channel) then read the RX_STATUS register again.

RX_STATUS b13: Channel B FIR error
This bit gets set, if Rx channel B is enabled, if a new output value from the FIR filter is not ready when the rising edge of the CR3 clock occurs. This happens if the FIR filter length is too large for the chosen clock and sample rates. When this happens, data sent from the FIR filter to the Rx serial port may become lost or corrupted. This bit gets cleared only when RX_STATUS is read.

RX_STATUS b12: Channel B 2nd bit selector overflow
This bit gets set to 1 when the channel B 2nd bit selector output value saturates to maximum positive ($7FFF) or maximum negative ($8000). This bit gets cleared only when RX_STATUS is read.

RX_STATUS b11: Channel B 1st bit selector overflow
This bit gets set to 1 when the channel B 1st bit selector output value saturates to maximum positive ($7FFFF) or maximum negative ($80000). This bit gets cleared only when RX_STATUS is read.

RX_STATUS b10: Channel B input overflow
This bit gets set to 1 when the channel B input exceeds a pre-programmed limit (see section 9.1). This bit gets cleared only when RX_STATUS is read.

RX_STATUS b9: Channel B buffer overrun
This bit gets set to 1 whenever Rx channel B generates a data sample before the Rx serial port is able to take it, for instance if the Rx serial port gets disabled or is configured to run too slowly. When this happens, the data sample will be lost. The buffer overrun bit gets cleared only when RX_STATUS is read.

RX_STATUS b8: Channel B idle
This bit is a level sensitive signal that is set to 1 whenever channel B is in the idle state.

RX_STATUS b7: Reserved, set to 0

RX_STATUS b6-0: (Channel A status bits)
Similar in operation to bits 14-7.

RX_ST_ENAB - $2F: 16-bit Write
Reset value = $00

Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Rx status enable (channel B)							0	Rx status enable (channel A)						

RX_ST_ENAB b15, b7: Reserved, set to 0

RX_ST_ENAB b14-8, b6-0: Rx status enable
If any of these bits is high while the corresponding bit in the RX_STATUS register is also high, then the Rx status bit (in STATUS register bit 1) gets set to 1.

10 Transmit Channel

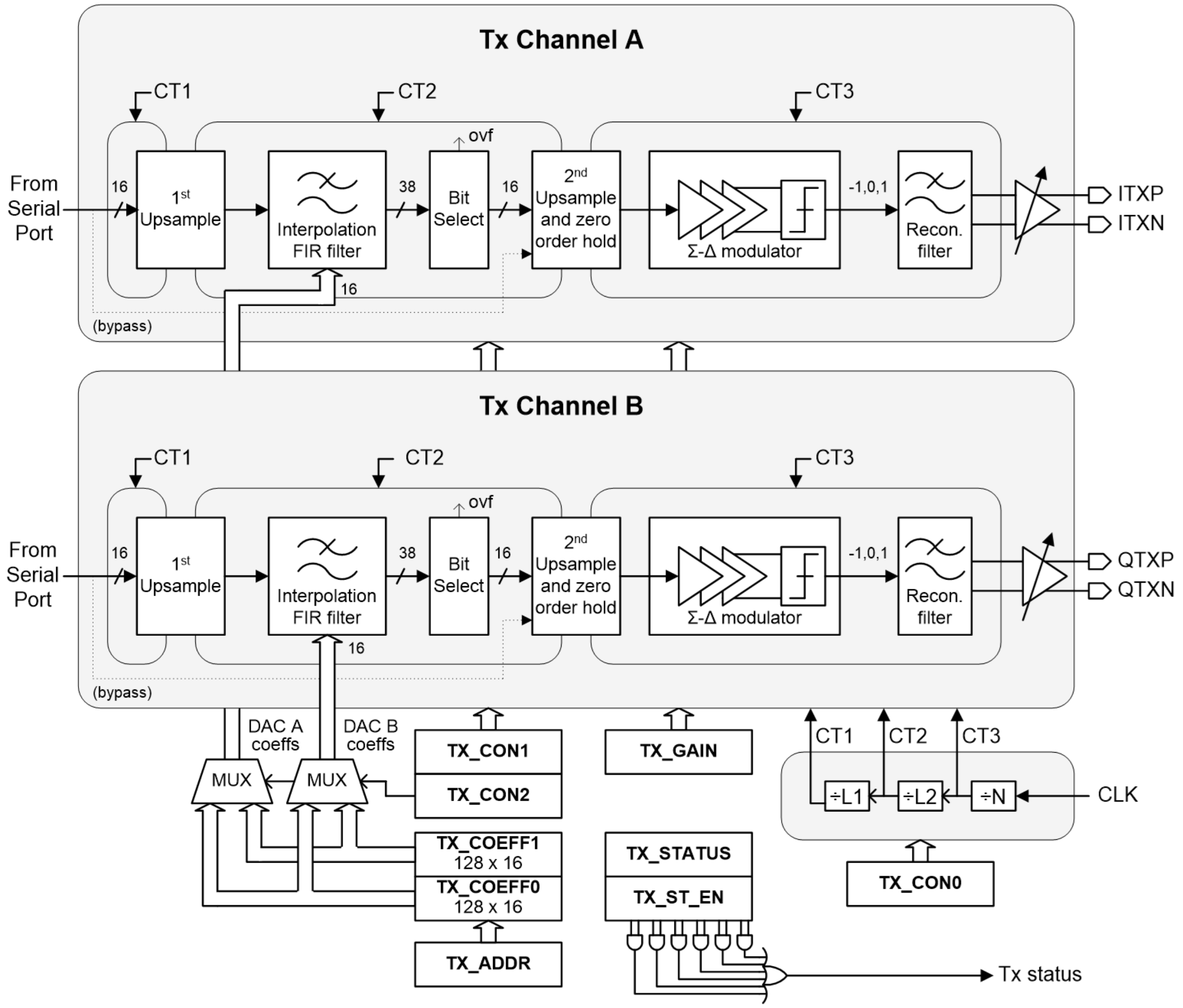


Figure 14 Tx Channel A and B

The CMX983 has two Tx channels (Figure 14). Within each Tx channel, data can be upsampled twice. Data from the serial port, formatted as 16-bit 2’s complement values, optionally passes through a first upsampler, a programmable FIR interpolation filter and a bit selector. The data then passes through a second upsampler and into a sigma-delta modulator. The resulting signal then passes through a reconstruction filter and variable gain block before being driven onto the Tx output pins.

First upsampler

The first upsampler takes 16-bit 2’s complement values from the serial port at a rate equal to CT1 and outputs them to the interpolation filter at a rate equal to CT2. The upsampler uses zero padding, so for an upsample factor of L1 (= f_{CT2}/f_{CT1}) there are L1-1 zero-valued samples inserted between each of the original time samples. This upsampling process results in unwanted spectral images at multiples of the CT1 frequency, but otherwise the original signal remains undistorted. The first upsample factor can be set to any value up to 8. Alternatively, the first upsampler, FIR filter and bit selector can be disabled and bypassed.

Interpolation filter

The interpolation filter is a 128-tap FIR type that takes the zero padded samples from the first upsampler at a frequency equal to CT2. There are two banks of programmable coefficients. The purpose of the low-pass interpolation filter is to attenuate the unwanted spectral images caused by the first upsampler, perform any transfer function shaping that is required by the transmission standard and, if necessary, provide compensation for the droop caused by the second upsampler or the DAC reconstruction filter.

Bit selector

The bit selector at the output of the interpolation filter selects which 16 bits of the 38-bit FIR filter accumulator are passed to the following upsample and hold stage.

Second upsample and hold

The second upsampler includes a zero-order hold function, so for an upsample factor of L2 (= f_{CT3}/f_{CT2}) there are L2-1 repeated sample values inserted after each sample value from the bit selector. This upsampling process results in unwanted spectral images at multiples of the CT2 frequency, but these lie near the nulls in transfer function of the zero-order hold so they are usefully attenuated. The transfer function of the zero-order hold is $H(z)=(1-z^{-L2})/(z-1)$, where the sample period $T=1/f_{CT2}$. This transfer function also causes droop in the wanted signal, but if the CT2 clock rate is high enough this can be made insignificant. If necessary, droop compensation can be performed in the preceding FIR interpolation filter. The second upsample factor can be set to any value up to 32.

Sigma-delta modulator and reconstruction filter

The sigma-delta modulator is a 2nd-order type whose output has a characteristic high-pass filtered noise profile, with the quantisation noise rising at 12dB per octave. The reconstruction filter attenuates this quantisation noise, along with any spectral remnants from the upsamplers. The reconstruction filter comprises a linear-phase switched capacitor filter (with a selectable bandwidth) followed by a linear-phase continuous time filter. These have the following nominal transfer characteristics, with a sample period $T=1/f_{CT3}$:

Switched capacitor section:

$$\begin{cases} \text{Low bandwidth}: H(z) = \dfrac{0.0413510}{1-0.958649\,z^{-1}} \times \dfrac{0.00206664\;z^{-1}}{1-1.93323\,z^{-1}+0.935301\,z^{-2}} \\ \\ \text{High bandwidth}: H(z) = \dfrac{0.0794179}{1-0.920582\,z^{-1}} \times \dfrac{0.00826662\;z^{-1}}{1-1.86233\,z^{-1}+0.870601\,z^{-2}} \end{cases}$$

Continuous time section:

$$H(s) = \frac{1}{1.8684\times10^{-12}s^2+1.9780\times10^{-6}s+1} \times \frac{1}{\underbrace{1.496\times10^{-6}s+1}_{\substack{\text{Recommended} \\ \text{off-chip filter}}}}$$

With a sigma-delta clock frequency of 2.4MHz, the switched capacitor section has a -3dB cutoff frequency of approximately 12.2kHz (low b/w) or 23.9kHz (high b/w). The continuous time filter has a -3dB cutoff frequency of approximately 80kHz.

Output gain stage

The final analogue gain stage between the reconstruction filter and the Tx output pins can be set to a gain of between -11dB and 11dB (in 0.5dB steps). A mute setting is also provided. With the gain set to 0dB and with a fixed digital input of ±32767, the differential output voltage from the Tx DAC is approximately ±0.75 × AV_{DD}.

The following C-BUS registers are used to configure the transmit channels:

TX_CON0 - $30: 8-bit Write
Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	SC high b/w	DAC clock divide						

TX_CON0 b7: SC high b/w
Set to 1 to select the high bandwidth cutoff for the reconstruction filter switched capacitor section. Set to 0 to select the low bandwidth cutoff.

TX_CON0 b6-0: DAC clock divide
Sets the division ratio between CLK and CT3, where CT3 is the clock for the sigma-delta modulator and reconstruction filter. This value can be set to between 2 and 128 (0000000 = 128).

TX_CON1 - $31: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	2nd upsample rate					1st stage bypass	Interpolation filter length							1st upsample rate		

TX_CON1 b15-11: Second upsample rate
Sets the division ratio between the CT3 and CT2 clock, which determines the second upsample rate L2. This can be set to between 1 and 32 (00000 = 32).

TX_CON1 b10: First stage bypass
Set this bit to 1 to disable the first upsampler, interpolation filter and bit selector, and cause Tx data from the serial port to be driven directly into the second upsampler.

TX_CON1 b9-3: Interpolation filter length
Sets the number of taps in the interpolation filter to a value between 1 and 128 (0000000 = 128). This must be set to an integer multiple of the first upsample rate. The filter length is also subject to the following restriction, based on the CLK and DAC sample frequencies:

$$FilterLength \le \frac{f_{CLK} - f_{CT2}}{f_{CT1}}$$

TX_CON1 b2-0: First upsample rate
The first upsample rate can be set to any value between 1 and 8 (000 = 8).

TX_CON2 - $34: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Tx B bit select					Tx B digital enable	Tx B analog enable	FIR B coeff select	Tx A bit select					Tx A digital enable	Tx A analog enable	FIR A coeff select

TX_CON2 b15-11: Tx B bit select
In Tx channel B, selects which 16 bits of the 38-bit FIR filter accumulator are passed to the following upsample stage. This value determines the number of most-significant bits discarded (valid range = 0 to 22): a value of 0 selects the most significant 16 bits of the FIR accumulator, a value of 1 discards the MSB of the accumulator and selects the next most significant 16 bits, and so on. Convergent rounding and saturation are applied to the selected bits. To assist with setup, an overflow causes a status bit to be set in the TX_STATUS register.

TX_CON2 b10: Tx B digital enable
Set this bit to 1 to enable the Tx channel B logic, from the first upsampler to the sigma-delta modulator, and start the transfer of data from the serial port. The channel idle status in the TX_STATUS register immediately goes low when the Tx channel B logic is enabled. The Tx B digital enable bit should be set after all other Tx channel B configuration bits have been initialised. When the Tx B digital enable bit changes from 1 to 0, the Tx channel stops requesting data from the Tx serial port but continues processing the buffered data (up to 2 words) until an underrun occurs. The interpolation filter data RAM is then cleared immediately (all data samples reset to zero) in readiness for the next time the channel is enabled; the coefficient RAMs are not altered. Clearing the data samples takes 128 CLK cycles, after which the channel idle status gets set to 1.

If the Tx B digital enable bit goes high while the associated Tx serial port remains disabled, default data values of $0000 will be fed into the Tx channel. This continues until the Tx serial port becomes enabled, at which point normal data transfers through the Tx serial port will commence. If the Tx serial port is subsequently disabled while the Tx B digital enable bit remains high, then the Tx channel will repeatedly transmit the final 16-bit data value obtained from the Tx serial port, and a buffer underrun will be flagged in TX_STATUS.

TX_CON2 b9: Tx B analogue enable
Set this bit to 1 to enable the Tx channel B reconstruction filter and output gain stage. Normally the analogue enable would be activated in advance of the digital enable and deactivated after all data in a transmission has been processed.

TX_CON2 b8: FIR B coefficient select
Selects which coefficients the channel B interpolation filter uses:
0 = TX_COEFF0
1 = TX_COEFF1

TX_CON2 b7-3: Tx A bit select
Similar in operation to bits 15-11.

TX_CON2 b2: Tx A digital enable
Set this bit to 1 to enable the Tx channel A logic, similar in operation to bit 10.

TX_CON2 b1: Tx A analogue enable
Set this bit to 1 to enable the Tx channel A reconstruction filter and output gain stage, similar in operation to bit 9.

TX_CON2 b0: FIR A coefficient select
Selects which coefficients the channel A interpolation filter uses:
0 = TX_COEFF0
1 = TX_COEFF1

TX_GAIN - $35: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	Tx B clamp	Tx B gain						0	Tx A clamp	Tx A gain					

TX_GAIN b15: Reserved, set to 0

TX_GAIN b14: Tx B clamp
When the Tx analog is disabled (TX_CON2 bit 9 = 0), this bit controls what happens to the DAC output pins QTXP and QTXN: when the Tx B clamp bit is set to 1 then QTXP and QTXN get clamped to a mid-rail reference voltage $AV_{DD}/2$; otherwise, QTXP and QTXN will become high impedance (greater than 100kΩ).

TX_GAIN b13-8: Tx B gain
Set channel B output gain to between -11dB and +11dB in 0.5dB steps, or mute the output. The gain setting is in 2's complement format with an implicit binary point between bit 9 and 8:

```
100000 = mute
101010 = -11.0dB
101011 = -10.5dB
101100 = -10.0dB
      ...
111110 = -1.0dB
111111 = -0.5dB
000000 = 0.0dB
000001 = 0.5dB
000010 = 1.0dB
      ...
010100 = 10.0dB
010101 = 10.5dB
010110 = 11.0dB
```

TX_GAIN b7: Reserved, set to 0

TX_GAIN b6: Tx A clamp
Controls ITXP and ITXN clamp level, similar in operation to bit 14.

TX_GAIN b5-0: Tx A gain
Set channel A output gain, similar in operation to bits 13-8.

TX_COEFF0 - $36: 16-bit Write, data-streaming

TX_COEFF1 - $37: 16-bit Write, data-streaming

Reset value = undefined

Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIR coefficient values (2's complement)															

These two C-BUS locations allow the 128-word FIR coefficient RAMs for Tx channel A and B to be loaded. Each coefficient RAM can be loaded in ascending order by repeatedly writing data to the same C-BUS location (an internal address pointer automatically increments after each write). To increase the loading rate of the coefficients, data-streaming operation is supported for these two C-BUS addresses. Only as many coefficients as are required (determined by the FIR length) need to be loaded into each RAM, unused RAM locations do not need to be written.

The length of the interpolation filter, set in TX_CON2, must be an integer multiple of the first upsample rate L1. Because of the way the interpolation filter is implemented, the coefficients must be scrambled before loading as demonstrated in the following code example:

```
// initialise array of coefficients
int coefficient[FILTER_LENGTH] = {0x0002, 0x0031, 0xFFA3, ...};

// initialise pointer
cbus_write_8bits(TX_ADDR, 0x00);

// load coefficients into CMX983
for (i = 0; i < UPSAMPLE_RATE1; i++)
  for (j = 0; j < FILTER_LENGTH; j += UPSAMPLE_RATE1)
    cbus_write_16bits(TX_COEFF0, coefficient[i + j]);
```

For instance, if the first upsample ratio is set to 5 and the filter length is set to 60, the coefficients C0..C59 would be loaded in the following sequence. Note that C0 is the coefficient that is multiplied by the most recent data sample from the first upsampler:

C0, C5, C10, … C50, C55, C1, C6, C11, … C51, C56, C2, C7, C12, … C52, C57,
C3, C8, C13, … C53, C58, C4, C9, C14, … C54, C59

Before loading either of the two coefficient RAMs, the internal address pointer needs to be initialised (usually to address 0); this is done by writing to register TX_ADDR. Note that both coefficient RAMs share this address pointer, so the address needs to be initialised before each coefficient RAM is loaded.

TX_ADDR - $38: 8-bit Write

Reset value = $00

Bit:

7	6	5	4	3	2	1	0
0	Tx coefficient address pointer						

The Tx coefficient address pointer determines the address at which data gets written during a C-BUS write to either of the Tx coefficient RAMs. The Tx coefficient address pointer automatically increments after each 16-bit coefficient value is written, so if the coefficients are written in an ascending sequence the pointer only needs to be initialised once before each coefficient RAM is loaded.

TX_STATUS - $39: 8-bit Read

Reset value = $11

Bit:

7	6	5	4	3	2	1	0
0	Ch. B bitsel over-flow	Ch. B buffer under-run	Ch. B idle	0	Ch. A bitsel over-flow	Ch. A buffer under-run	Ch. A idle

TX_STATUS b7: Reserved, set to 0

TX_STATUS b6: Channel B bit selector overflow

This bit gets set to 1 when the channel B bit selector output value saturates to maximum positive ($7FFF) or maximum negative ($8000). This bit gets cleared only when TX_STATUS is read.

TX_STATUS b5: Channel B buffer underrun

This bit gets set to 1 whenever Tx channel B runs out of data from the serial port, for instance if the Tx serial port gets disabled or is configured to run too slowly. When this happens, the Tx channel will retransmit the previous data sample. The buffer underrun bit gets cleared only when TX_STATUS is read.

TX_STATUS b4: Channel B idle

This bit is a level sensitive signal that is set to 1 whenever channel B is in the idle state.

TX_STATUS b3: Reserved, set to 0

TX_STATUS b2-0: (Channel A status bits)

Similar in operation to bits 6-4.

TX_ST_ENAB - $3A: 8-bit Write

Reset value = $00

Bit:

7	6	5	4	3	2	1	0
0	Tx status enable (channel B)			0	Tx status enable (channel A)		

TX_ST_ENAB b7, b3: Reserved, set to 0

TX_ST_ENAB b6-4, b2-0: Tx status enable

If any of these bits is high while the corresponding bit in the TX_STATUS register is also high, then the Tx status bit (in STATUS register bit 0) gets set to 1.

11 Serial Ports

11.1 Rx Serial Port

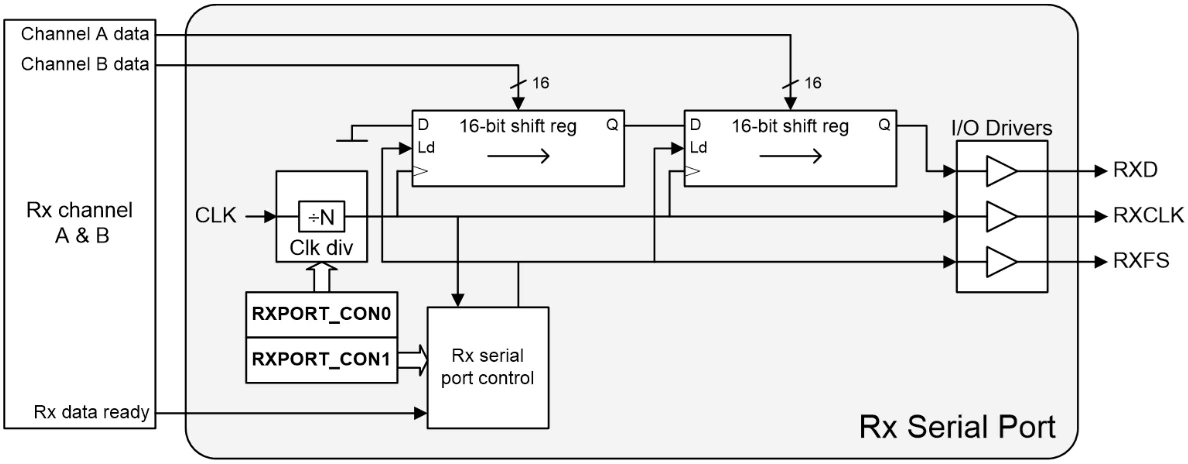


Figure 15 Rx Serial Port

Data from Rx channel A and B are output from the CMX983 through a PCM serial port (Figure 15). PCM data is sent in Short Frame Sync mode with the CMX983 acting as a master. The PCM clock RXCLK is divided down from the system clock CLK, and data transmission is most significant bit first. If both Rx Channel A and B are selected, the two 16-bit data words are multiplexed through a single output data pin RXD as shown in Figure 16. If only one of the two Rx channels is selected, then the 16-bit serial port shift register for the disabled channel is bypassed and only the data for the selected channel is shifted out (example shown in Figure 17).

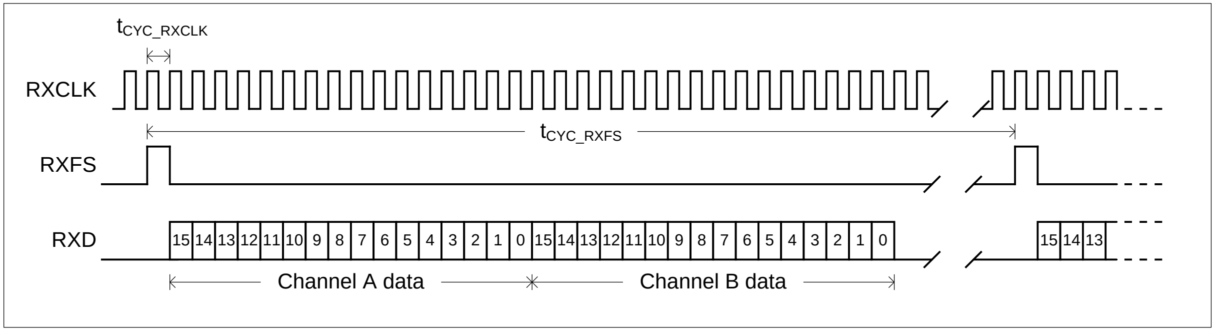


Figure 16 Rx Port Timing (Channel A and B both selected)

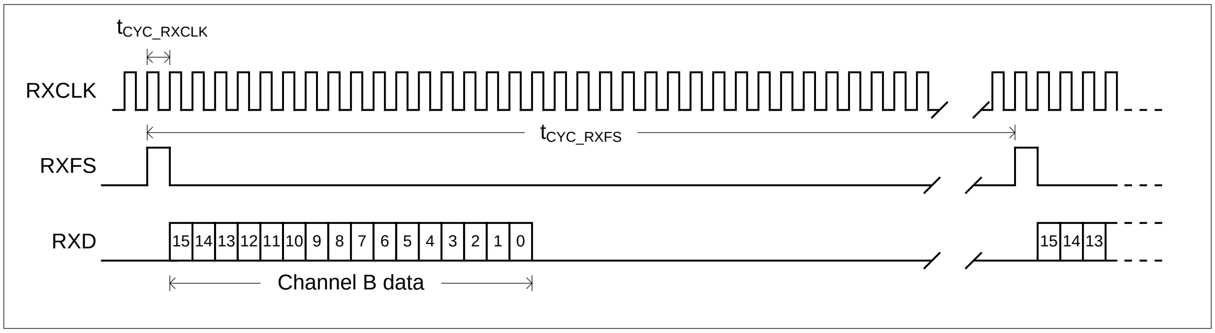


Figure 17 Rx Port Timing (Channel B only selected)

The following C-BUS registers are used to configure the Rx serial port:

RXPORT_CON0 - $40: 8-bit Write
Reset value = $00

Bit:

7	6	5	4	3	2	1	0
RXCLK divide value							

RXPORT_CON0 b7-0: RXCLK divide value
Sets the division ratio between CLK and RXCLK. This value can be set to between 2 and 256 (00000000 = 256). RXCLK has a nominal 50:50 duty cycle, and its frequency must be at least 32 times greater than f_{CR3} (the frequency of the Rx channel second downsample clock, section 0); back-to-back data frames are allowed. Note: To avoid jitter on the RXFS signal, the frequency ratio f_{RXCLK} / f_{CR3} must be an integer.

RXPORT_CON1 - $41: 8-bit Write
Reset value = $00

Bit:

7	6	5	4	3	2	1	0
0	0	0	Chan. B data select	Chan. A data select	RXD hi-Z	Rx port enable	Invert RX-CLK

RXPORT_CON1 b7-5: Reserved, set to 0

RXPORT_CON1 b4: Channel B data select
Set to 1 to cause the 16-bit channel B data to be output on the RXD pin. Set to 0 to prevent channel B data from being output on the RXD pin (this bypasses the 16-bit channel B shift register).

RXPORT_CON1 b3: Channel A data select
Set to 1 to cause the 16-bit channel A data to be output on the RXD pin. Set to 0 to prevent channel A data from being output on the RXD pin (this bypasses the 16-bit channel A shift register).

RXPORT_CON1 b2: RXD hi-Z
Set to 1 to cause the RXD pin to go high impedance between data packets. Set to 0 to cause the RXD pin to be driven low between data packets.

RXPORT_CON1 b1: Rx port enable
Set to 1 to enable the Rx serial port and start the RXCLK pin oscillating. The frame sync pulse RXFS will only be generated when the Rx serial port is enabled, *and* the Rx channel(s) are enabled.

Set to 0 to disable the Rx serial port and drive RXCLK low. When the Rx serial port is disabled, data samples generated by the Rx channels will be discarded. If the Rx port enable bit changes from 1 to 0 during a data frame, the frame will complete before the RXCLK pin stops oscillating.

RXPORT_CON1 b0: Invert RXCLK
Set this bit to 1 to invert the RXCLK signal. This bit should not be changed if the Rx port enable bit has already been set to 1.

11.2 Tx Serial Port

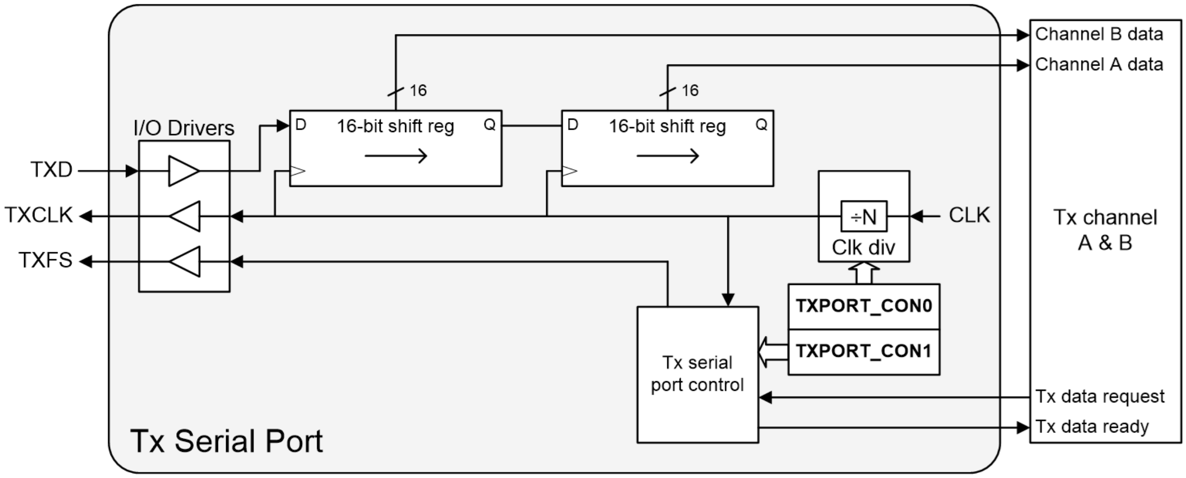


Figure 18 Tx Serial Port

Data for Tx channel A and B are input to the CMX983 through a PCM serial port (Figure 18). PCM data is sent in Short Frame Sync mode with the CMX983 acting as a master. The PCM clock TXCLK is divided down from the system clock CLK, and data transmission is most significant bit first. If both Tx Channel A and B are selected, the two 16-bit data words are multiplexed through a single input data pin TXD as shown in Figure 19. If only one of the two Tx channels is selected, then the 16-bit serial port shift register for the disabled channel is bypassed and only the data for the selected channel is shifted in (example shown in Figure 20).

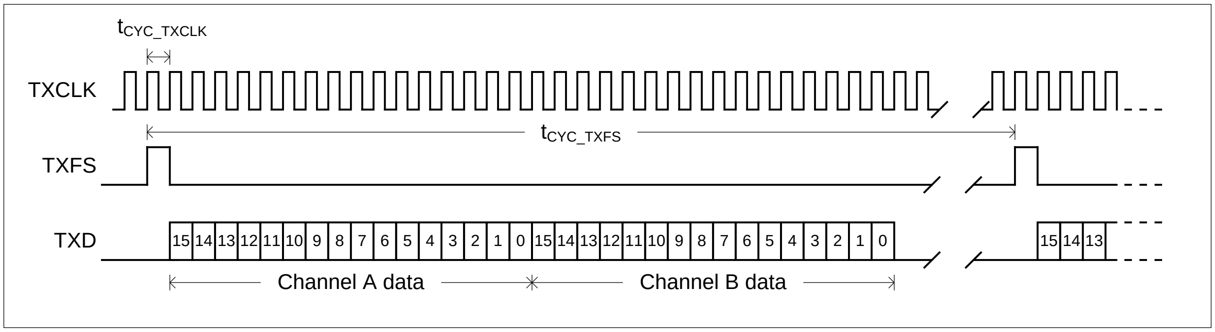


Figure 19 Tx Port Timing (Channel A and B both selected)

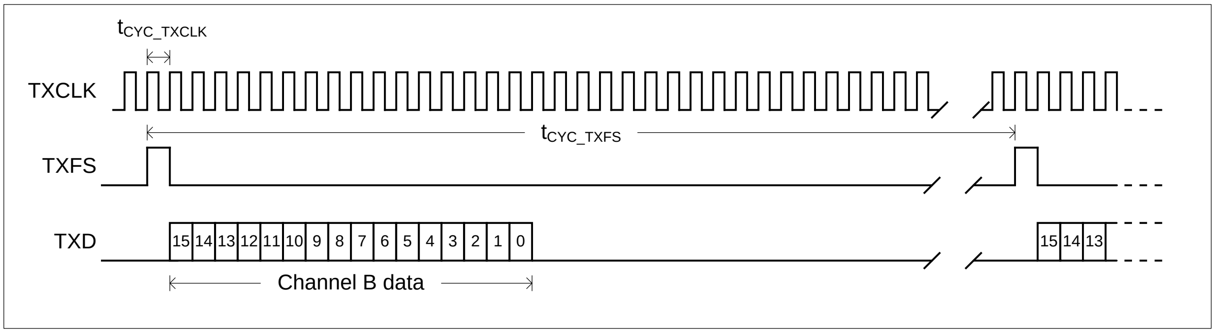


Figure 20 Tx Port Timing (Channel B only selected)

The following C-BUS registers are used to configure the Tx serial port:

TXPORT_CON0 - $48: 8-bit Write-only
Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	TXCLK divide value							

TXPORT_CON0 b7-0: TXCLK divide value
Sets the division ratio between CLK and TXCLK. This value can be set to between 2 and 256 (00000000 = 256). TXCLK has a nominal 50:50 duty cycle, and its frequency must be at least 32 times greater than f_{CT1} (the frequency of the Tx channel first upsample clock, section 10); back-to-back data frames are allowed. Note: to avoid jitter on the TXFS signal, the frequency ratio f_{TXCLK} / f_{CT1} must be an integer.

TXPORT_CON1 - $49: 8-bit Write
Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	0	0	0	0	Chan. B data select	Chan. A data select	Tx port enable	Invert TX-CLK

TXPORT_CON1 b7-4: Reserved, set to 0

TXPORT_CON1 b3: Channel B data select
Set to 1 to cause the 16-bit channel B data to be input on the TXD pin. Set to 0 to prevent channel B data from being input on the TXD pin (this bypasses the 16-bit channel B shift register).

TXPORT_CON1 b2: Channel A data select
Set to 1 to cause the 16-bit channel A data to be input on the TXD pin. Set to 0 to prevent channel A data from being input on the TXD pin (this bypasses the 16-bit channel A shift register).

TXPORT_CON1 b1: Tx port enable
Set to 1 to enable the Tx serial port and start the TXCLK pin oscillating. The frame sync pulse TXFS will only be generated when the Tx serial port is enabled, *and* the Tx channel(s) are enabled.

Set to 0 to disable the Tx serial port and drive TXCLK low. If the Tx port enable bit changes from 1 to 0 during a data frame, the frame will complete before the TXCLK pin stops oscillating.

TXPORT_CON1 b0: Invert TXCLK
Set this bit to 1 to invert the TXCLK signal. This bit should not be changed if the Tx port enable bit has already been set to 1.

12 Fractional-N Frequency Synthesizers

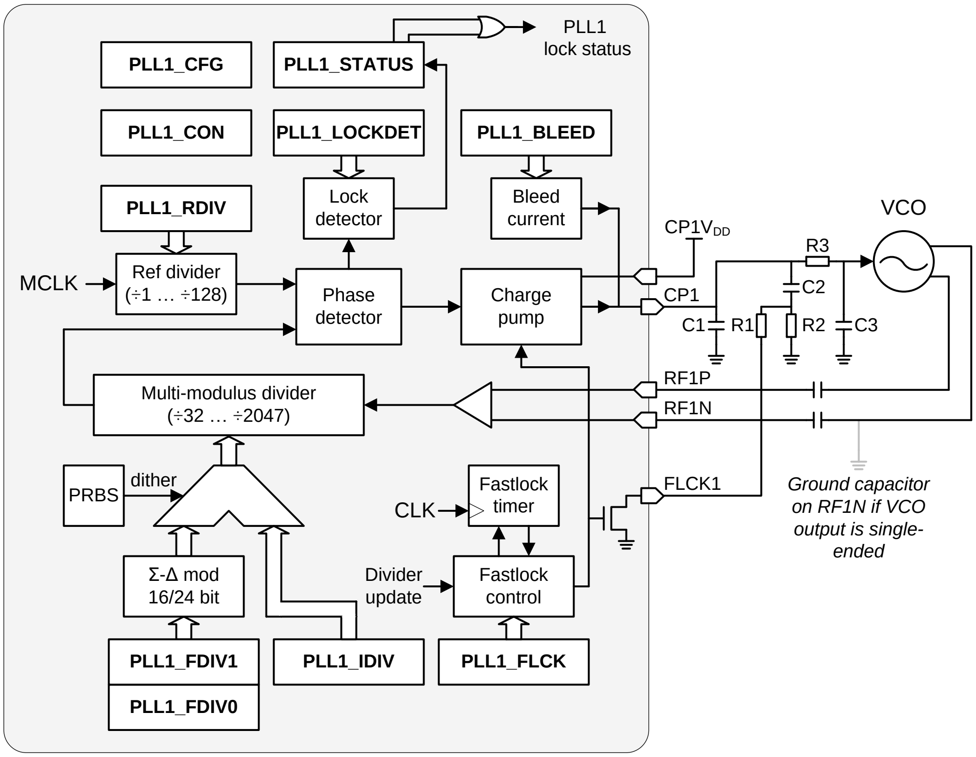


Figure 21 Fractional-N Frequency Synthesizer

The CMX983 has two identical, independently programmable, 2.1 GHz fractional-N frequency synthesizers; a block diagram of one of the synthesizers is shown in Figure 21. Note that a single ended VCO signal can be used; if so, it should be ac-coupled to RF1P and the capacitor connected to RF1N should be grounded.

The synthesizers use a sigma-delta modulation technique that allows use of a high reference frequency, thus providing rapid frequency switching and low phase noise performance. The 24-bit fractional divider resolution provides an ultra-fine step size for narrowband applications, and can be used to compensate for crystal oscillator frequency drift or Doppler shift.

A fast locking mechanism is provided that increases the transition rate when changing to a new operating frequency. This is done by temporarily modifying the loop filter characteristics and charge pump gain whenever the main divider settings are updated, allowing the responsiveness of the closed loop system to be increased without compromising the loop stability. The fast lock mode automatically turns off after a predetermined delay, thus reverting the PLL to its standard, low noise mode of operation.

Each synthesizer has a programmable lock detector circuit that indicates when the loop is in lock. The lock detectors can be configured for analogue or digital operation, and no external components are required.

The PLL synthesizers are configured through a number of C-BUS registers. The registers for PLL1 and PLL2 operate in an identical way:

PLL1_CON - $4E; 16-bit Write

PLL2_CON - $57: 16-bit Write
Reset value = $0000

Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Resol-ution	Mode select			Enab dither	0	Enab PLL	Inv. CP	In-lock status enab	Out-of-lock status enab	Lock status edge trigger	Charge pump current			

PLL1[2]_CON b15: Reserved, set to 0

PLL1[2]_CON b14: Resolution
Set this bit to 0 to select a 24-bit fractional value for the main divider (using registers PLL1[2]_FDIV1 and PLL1[2]_FDIV0). Set this bit to 1 to select a 16-bit fractional value for the main divider (using register PLL1[2]_FDIV0 only).

PLL1[2]_CON b13-11: Mode select
Sets the main divider operating mode:

000	Integer mode (sigma-delta disabled)
001	Fractional-N divider with 3rd order modulator
110	Fractional-N divider with alternative 3rd order modulator

Other values should not be used.

The two types of Fractional-N modulator offer different noise characteristics. The type ‘110’ generally has the best close-in noise (characterised by 1 Hz Normalised Phase Noise), whereas the type ‘001’ has lower sigma-delta noise at offset around 1 MHz. Note: the exact characteristics of the PLL noise will depend on the overall PLL design including the VCO gain and loop filter.

PLL1[2]_CON b10: Enable dither
Set this bit to 1 to add a “dither” to the LSB of the fractional divide value. This helps to suppress idle tones from the sigma-delta modulator output. Set this bit to 0 to disable the dither.

PLL1[2]_CON b8: Enable PLL
Set to 1 to enable the PLL circuit (sigma-delta modulator, multi-modulus divider, reference divider, phase detector and charge pump). Set to 0 to disable and powersave the PLL circuit.

PLL1[2]_CON b7: Invert charge pump
With this bit set to 0 the charge pump will sink current when the main divider output frequency f_{main} is a higher frequency than the reference clock f_{ref}. Set this bit to 1 to invert the charge pump output, so that it sources current when $f_{main} > f_{ref}$.

PLL1[2]_CON b6: In-lock status enable
Set to 1 to allow the in-lock status bit in the PLL1[2]_STATUS register to be set when lock is detected. The in-lock status bit can either be edge-triggered or level-triggered, depending on the state of PLL1[2]_CON bit 4.

PLL1[2]_CON b5: Out-of-lock status enable
Set to 1 to allow the out-of-lock status bit in the PLL1[2]_STATUS register to be set when lock is lost. The out-of-lock status bit can either be edge-triggered or level-triggered, depending on the state of PLL1[2]_CON bit 4.

PLL1[2]_CON b4: Lock status edge trigger
When this bit is set to 1, the in-lock status bit and out-of-lock status bit (PLL1[2]_STATUS bits 1-0) will be edge triggered. This means that the in-lock status bit gets set each time the lock signal transitions from 0 to 1, and the out-of-lock status bit gets set each time the lock signal transitions from 1 to 0.

When this bit is set to 0, the in-lock status bit and out-of-lock status bit (PLL1[2]_STATUS bits 1-0) will be level triggered. This means that the in-lock status bit will be continuously set high as long as lock = 1 and PLL1[2]_CON bit 6 = 1, and the out-of-lock status bit will be continuously set high as long as lock = 0 and PLL1[2]_CON bit 5 = 1.

PLL1[2]_CON b3-0: Charge pump current
Sets the value of the charge pump output current pulses. The value can be set in increments of 25µA, from 25µA (0000) to 400µA (1111).

PLL_CFG - $CE: 16-bit Write
Reset value = $2000

All bits in this register should be cleared to zero for optimum performance.

PLL1_LOCKDET - $4F: 16-bit Write
PLL2_LOCKDET - $58: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Lock detect enab	Lock mode	Reset lock	*Analogue configuration bits:*												
				0	0	0	0	0	Lock discharge rate				Lock charge rate			
				Digital configuration bits:												
				Loss-of-lock window		Lock window			Loss-of-lock threshold			Lock threshold				

PLL1[2]_LOCKDET b15: Lock detect enable
Set this bit to 1 to enable the lock detector circuit. Set this bit to 0 to disable and powersave the lock detector circuit.

PLL1[2]_LOCKDET b14: Lock mode
Set this bit to 1 to use the analogue lock detector. Set this bit to 0 to use the digital lock detector.

PLL1[2]_LOCKDET b13: Reset lock
Writing a 1 to this bit generates a short pulse that resets the lock detector (either analogue or digital) to an out-of-lock condition and clears PLL1[2]_STATUS bits 1-0. Immediately after writing a 1 to the reset lock bit, it is cleared back to 0 and the lock detector and lock status bits resume normal operation.

Analogue configuration bits

These bits are active when PLL1[2]_LOCKDET bit 14 = 1:

PLL1[2]_LOCKDET b12-8: Reserved, set to 0

PLL1[2]_LOCKDET b7-4: Lock discharge rate

These bits control the discharge rate of the analogue lock detector capacitor, which determines the time taken for the lock signal to go inactive when lock is lost. The discharge rate is specified as a multiple of the selected charge rate (see bits 3-0).

PLL1[2]_LOCKDET bits 7-4	Discharge rate	PLL1[2]_LOCKDET bits 7-4	Discharge rate
$0	5 x charge rate	$8	100 x charge rate
$1	7 x charge rate	$9	150 x charge rate
$2	10 x charge rate	$A	200 x charge rate
$3	15 x charge rate	$B	300 x charge rate
$4	20 x charge rate	$C	500 x charge rate
$5	30 x charge rate	$D-$F	Do not use
$6	50 x charge rate		
$7	70 x charge rate		

PLL1[2]_LOCKDET b3-0: Lock charge rate

These bits control the charge rate of the analogue lock-detector capacitor, and so determine the time taken for the lock signal to become active when the phase detector inputs are in phase. The nominal time taken for the capacitor to fully charge from a reset state, assuming no discharge pulses occur, is shown below:

PLL1[2]_LOCKDET bits 3-0	Charge time	PLL1[2]_LOCKDET bits 3-0	Charge time
$0	5 µs	$8	100 µs
$1	7 µs	$9	150 µs
$2	10 µs	$A	200 µs
$3	15 µs	$B	300 µs
$4	20 µs	$C	500 µs
$5	30 µs	$D	700 µs
$6	50 µs	$E	1 ms
$7	70 µs	$F	Do not use

Digital configuration bits

These bits are active when PLL1[2]_LOCKDET bit 14 = 0:

PLL1[2]_LOCKDET b12-11: Loss-of-lock window

While the loss-of-lock counter is active (i.e. lock = 1), these bits determine the phase detector error window: if the difference in arrival time of the phase detector inputs is outside this window, they are deemed to be “out of phase”. When enough consecutive “out of phase” pulses occur (determined by PLL1[2]_LOCKDET bits 7-5) then the lock signal gets set to 0. The loss-of-lock window is specified as a multiple of the lock window value, which in turn is determined by PLL1[2]_LOCKDET bits 10-8.

PLL1[2]_LOCKDET bits 12-11	Loss-of-lock window
00	1.0× lock window
01	1.5× lock window
10	2.0× lock window
11	Illegal, do not use

PLL1[2]_LOCKDET b10-8: Lock window

While the lock counter is active (i.e. lock = 0), these bits determine the phase detector error window: if the difference in arrival time of the phase detector inputs is within this window, they are deemed to be "in phase". When sufficient consecutive "in phase" pulses occur (determined by PLL1[2]_LOCKDET bits 4-0) then the lock signal gets set to 1. The nominal value of the error window is shown in the following table:

PLL1[2]_LOCKDET bits 10-8	Lock window	PLL1[2]_LOCKDET bits 10-8	Lock window
$0	±7 ns	$4	±30 ns
$1	±10 ns	$5	±50 ns
$2	±15 ns	$6	±70 ns
$3	±20 ns	$7	±100 ns

PLL1[2]_LOCKDET b7-5: Loss-of-lock threshold

While the lock indicator is active (lock = 1), these bits determine how many consecutive "out of phase" signals must occur at the phase detector before loss-of-lock is detected, causing the lock indicator to go inactive (lock = 0). The loss-of-lock threshold can be set to between 1 and 8 (000 = 8).

PLL1[2]_LOCKDET b4-0: Lock threshold

While the lock indicator is inactive (lock = 0), these bits determine how many consecutive "in phase" signals must occur at the phase detector before lock is detected, causing the lock indicator to go active (lock = 1). The lock threshold can be set to between 1 and 32 (00000 = 32).

PLL1_FLCK - $50: 16-bit Write
PLL2_FLCK - $59: 16-bit Write

Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	Enab fastlck	Fastlock timer coarse divide			Fastlock timer fine divide							Fastlock current	

PLL1[2]_FLCK b15-13: Reserved, set to 0

PLL1[2]_FLCK b12: Enable fastlock

Set to 1 to enable fastlock. Then each time the main divider registers are updated, the associated fastlock pin (FLCK1 or FLCK2) is pulled to ground, the charge pump current changes to the value set by PLL1[2]_FLCK bits 1-0, and the fastlock timer is started. The fastlock state continues until the timer expires, at which point the fastlock pin returns to a high impedance state and the charge pump current reverts to the value determined by PLL1[2]_CON bits 3-0.

PLL1[2]_FLCK b11-9: Fastlock timer coarse divide
PLL1[2]_FLCK b8-2: Fastlock timer fine divide

These bits control the duration of the fastlock mode. The coarse divide can be set to a value between 0 and 7, and the fine divide can be set to between 1 and 128 (0000000 = 128). The fastlock timer is clocked by the internal system clock CLK, and its period is given by the following expression:

$$T_{FASTLOCK} = \frac{4^{CoarseDivide} \times FineDivide}{f_{CLK}}$$

PLL1[2]_FLCK b1-0: Fastlock current

Sets the value of the charge pump output current pulses in fastlock mode. The value is set as a multiple of the nominal charge pump current:

PLL1[2]_FLCK bits 1-0	Charge pump current multiplier M
00	4x
01	8x
10	12x
11	16x

To maintain loop stability with fastlock active the resistor R1 shown in Figure 21 will typically need to be set to the following value:

$$R1 \approx \frac{R2}{\sqrt{M} - 1}$$

With fastlock active, the PLL lock time is decreased by a factor of approximately $\sqrt{M}$. In practice, an even greater reduction is often achieved because fastlock can reduce or eliminate “cycle slipping” in the phase detector.

PLL1_BLEED - $51: 8-bit Write
PLL2_BLEED - $5A: 8-bit Write

Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	0	0	Enab bleed	Bleed coarse			Bleed fine	

These registers can be used to add a bleed current to the charge pump output of the PLLs. The bleed current adds a phase shift to the PLL loop and can help reduce spurious products associated with the sigma-delta modulator by operating the charge pump in a more linear region. Excessive bleed current should be avoided because it can increase the PLL reference spurs and phase noise. A good initial compromise is to set the phase shift to about four VCO cycles:

$$I_{bleed} \approx \frac{4 f_{MCLK} I_{CP}}{R_{div} f_{VCO}}$$

where R_{div} is the reference divider value and I_{CP} is the charge pump current setting. This bleed current value can then be adjusted to optimise performance. Any leakage current from external components on the CP1 or CP2 pins must also be considered as this will alter the effective bleed current.

PLL1[2]_BLEED b7-6: Reserved, set to 0

PLL1[2]_BLEED b5: Enable bleed

Set to 1 to enable a constant bleed current to be sourced into the associated charge pump output pin.

PLL1[2]_BLEED b4-2: Bleed current (coarse)
PLL1[2]_BLEED b1-0: Bleed current (fine)

These bits control the nominal bleed current sourced into the charge pump output pin, according to the following formula:

$$I_{bleed} = 0.5\mu A \times 2^{bleed\_coarse} \times \left(1 + \frac{bleed\_fine}{4}\right)$$

The bleed current can therefore be set to a value within the range 0.5µA … 112µA. For instance, if PLL1[2]_BLEED bits 4-2 = 101_2 and PLL1[2]_BLEED bits 1-0 = 11_2, the resulting nominal bleed current will be 0.5µA×2^5×1.75 = 28µA.

Note that during fastlock, the bleed current is scaled up in the same proportion as the main charge pump current, as determined by PLL1[2]_FLCK bits 1-0.

PLL1_RDIV - $52: 8-bit Write
PLL2_RDIV - $5B: 8-bit Write
Reset value = $00

Bit:

7	6	5	4	3	2	1	0
0	Reference divider						

PLL1[2]_RDIV b7: Reserved, set to 0

PLL1[2]_RDIV b6-0:
Sets the division ratio between the master clock MCLK and the PLL reference clock. This value can be set to between 1 and 128 (0000000 = 128).

PLL1_IDIV - $53: 16-bit Write
PLL2_IDIV - $5C: 16-bit Write
Reset value = $0020

Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	Main divider integer value										

PLL1[2]_IDIV b15-11: Reserved, set to 0

PLL1[2]_IDIV b10-0: Main divider integer value
These bits represent the integer portion closest to the desired fractional-N divider value. The integer value is combined with the fractional value from registers PLL1[2]_FDIV1 and PLL1[2]_FDIV0 (which represent a fractional offset of between approximately +0.5 and –0.5) to allow selection of the desired VCO frequency. The valid range for the main divider integer value is from 32 to 2047 (in integer-N mode), or from 36 to 2043 (in fractional-N mode).

PLL1_FDIV0 - $54: 16-bit Write
PLL2_FDIV0 - $5D: 16-bit Write
Reset value = $0000

Bit:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Main divider fractional value (LSB)															

PLL1_FDIV1 - $55: 8-bit Write
PLL2_FDIV1 - $5E: 8-bit Write
Reset value = $00

Bit:

7	6	5	4	3	2	1	0
Main divider fractional value (MSB)							

PLL1[2]_FDIV0 b15-0: Main divider fractional value (LSB)
PLL1[2]_FDIV1 b7-0: Main divider fractional value (MSB)
In fractional-N mode, the fractional divide value ranges between approximately –0.5 and +0.5 as determined by the PLL1[2]_FDIV1 and PLL1[2]_FDIV0 registers:

With fractional resolution set to 24 bits, the registers are concatenated to form a 24-bit 2's complement number *fdiv*. The resulting fractional divide value is equal to (fdiv ÷ 2^{24}), which is in the range –0.5 to +0.49999994…

With fractional resolution set to 16 bits, the PLL1[2]_FDIV1 register is ignored and the value in the PLL1[2]_FDIV0 register is treated as a 16-bit 2's complement number *fdiv*. The fractional divide value is equal to (fdiv ÷ 2^{16}), which is in the range –0.5 to +0.49998474…

In integer-N mode, both the PLL1[2]_FDIV1 and PLL1[2]_FDIV0 registers are ignored.

PLL1_STATUS - $56: 8-bit Read
PLL2_STATUS - $5F: 8-bit Read
Reset value = $00

Bit:

7	6	5	4	3	2	1	0
Lock	0	0	0	0	0	In-lock status	Out-of-lock status

PLL1[2]_STATUS b7: Lock
This bit shows the current state of the lock detector output: lock = 1 indicates "in lock", and lock = 0 indicates "out of lock". This bit can be polled by the host processor.

PLL1[2]_STATUS b6-2: Reserved, set to 0

PLL1[2]_STATUS b1: In-lock status
To enable this status bit, the in-lock status enable bit (PLL1[2]_CON bit 6) must be set to 1. The in-lock status bit can be either edge triggered or level triggered, depending on the state of PLL1[2]_CON bit 4. When configured as edge triggered, the in-lock status bit gets set high after each 0 to 1 transition of the lock signal, and gets cleared when the PLL1[2]_STATUS register is read. When configured as level triggered, the in-lock status bit gets continuously set high as long as lock = 1 and PLL1[2]_CON bit 6 = 1, and gets cleared when the PLL1[2]_STATUS register is read *and* either lock = 0 or PLL1[2]_CON bit 6 = 0. Note: the in-lock status bit is also cleared to 0 when a "reset lock" operation is performed (see description of PLL1[2]_LOCKDET bit 13).

PLL1[2]_STATUS b0: Out-of-lock status
To enable this status bit, the out-of-lock status enable bit (PLL1[2]_CON bit 5) must be set to 1. The out-of-lock status bit can be either edge triggered or level triggered, depending on the state of PLL1[2]_CON bit 4. When configured as edge triggered, the out-of-lock status bit gets set high after

each 1 to 0 transition of the lock signal, and gets cleared when the PLL1[2]_STATUS register is read. When configured as level triggered, the out-of-lock status bit gets continuously set high as long as lock = 0 and PLL1[2]_CON bit 5 = 1, and gets cleared when the PLL1[2]_STATUS register is read *and* either lock = 1 or PLL1[2]_CON bit 5 = 0. Note: the out-of-lock status bit is also cleared to 0 when a “reset lock” operation is performed (see description of PLL1[2]_LOCKDET bit 13).

The in-lock status (bit 1) and the out-of-lock status (bit 0) are ORed together and the resulting signal is passed to the main STATUS register (section 6.4). This can be used to generate an interrupt signal.

12.1 Register Loading Order

To use the PLL1 synthesiser, the registers must be loaded in the order specified below. Similar rules apply to PLL2.

Registers PLL1_CON, PLL1_LOCKDET, PLL1_FLCK and PLL1_RDIV should be initialised before the main divider registers are loaded for the first time. The PLL enable bit (PLL1_CON bit 8) should be set during this process to power up the synthesizer circuit, but the charge pump output will remain in a high impedance state until the main divider registers are loaded.

After the main divider registers are loaded in the correct order (which depends on the operating mode), the PLL synthesizer begins operating. The main divider registers can be changed at any subsequent time, but must always be updated in the specified order:

Integer-N mode: Load PLL1_IDIV with the desired value, at which point the new divide ratio will take effect.

Fractional-N mode, 16-bit fractional resolution: Load PLL1_IDIV (if necessary), then load PLL1_FDIV0. The new divide ratio only takes effect when PLL1_FDIV0 is loaded.

Fractional-N mode, 24-bit fractional resolution: Load PLL1_IDIV and PLL1_FDIV1 (if necessary), then load PLL1_FDIV0. The new divide ratio only takes effect when PLL1_FDIV0 is loaded.

Each time the main divide registers are updated, at the point when the new divide ratio takes effect, a fastlock sequence in the associated PLL is triggered (if enabled).

12.2 Fractional-N Programming Examples

Example 1: To operate PLL1 in 16-bit fractional mode (modulator type 001) with a VCO frequency f_{VCO} = 803.125MHz, a master clock frequency f_{MCLK} = 19.2MHz, and a PLL comparison frequency f_{REF} = 2.4MHz.

Rdiv = $f_{MCLK} \div f_{REF}$ = 19.2MHz ÷ 2.4MHz = 8

Ndiv = $f_{VCO} \div f_{REF}$ = 803.125MHz ÷ 2.4MHz = 334.6354167

Split the N divider value into integer and fractional parts:

Idiv = Round(334.6354167) = 335 (decimal) = 0x014F (hex)

Fdiv (16-bit mode) = Round(2^{16} × (Ndiv – Idiv)) = -23893 (decimal) = 0xA2AB (hex)

Load C-BUS registers:

PLL1_CON bit 14 = 1 (16-bit), bit 13-11 = 001, bit 8 = 1 (enable), set other bits as desired
Set PLL1_LOCKDET and PLL1_FLCK as desired
PLL1_RDIV = 0x08
PLL1_IDIV = 0x014F
PLL1_FDIV0 = 0xA2AB

At this point, the charge pump is enabled and PLL1 begins to acquire lock. There is no need to load PLL1_FDIV1 in 16-bit mode. The frequency step size in this example is 2.4MHz ÷ 2^{16} ≈ 36.621Hz.

Example 2: To operate PLL2 in 24-bit fractional mode (modulator type 110) with a VCO frequency f_{VCO} = 1721.386MHz, a master clock frequency f_{MCLK} = 20.736MHz, and a PLL comparison frequency f_{REF} = 1.728MHz.

Rdiv = f_{MCLK} ÷ f_{REF} = 20.736MHz ÷ 1.728MHz = 12 (decimal) = 0x0C (hex)

Ndiv = f_{VCO} ÷ f_{REF} = 1721.386MHz ÷ 1.728MHz = 996.1724537

Split the N divider value into integer and fractional parts:

Idiv = Round(996.1724537) = 996 (decimal) = 0x03E4 (hex)

Fdiv (24-bit mode) = Round(2^{24} × (Ndiv – Idiv)) = 2893293 (decimal) = 0x2C25ED (hex)

Load C-BUS registers:

PLL2_CON bit 14 = 0 (24-bit), bit 13-11 = 110 , bit 8 = 1 (enable), set other bits as desired
Set PLL2_LOCKDET and PLL2_FLCK as desired
PLL2_RDIV = 0x0C
PLL2_IDIV = 0x03E4
PLL2_FDIV1 = 0x2C
PLL2_FDIV0 = 0x25ED

At this point, the charge pump is enabled and PLL2 begins to acquire lock. The frequency step size in this example is 1.728MHz ÷ 2^{24} ≈ 0.103Hz.

Note: if the calculated fractional part of Ndiv is exactly equal to 0.5 then the Idiv value should be rounded up. For example, if Ndiv = 312.5000 then Idiv = Round(Ndiv) = 313.

12.3 Lock Detector Configuration Guidelines

The CMX983 Fractional-N PLL synthesizers each contain an analogue lock detector and a digital lock detector. Both types of lock detector use the phase error in the PLL loop to determine whether the PLL is in lock. The user can select the type of lock detector that works best in their particular application. The following guidelines describe how the lock detectors in PLL1 operate; the lock detectors in PLL2 are identical.

12.3.1 Digital Lock Detector

Digital lock detector mode is selected when PLL1_LOCKDET bit 14 = 0. The digital lock detector is connected to the R-divider and N-divider output clocks as shown in Figure 22. The PLL phase error is measured as the time difference between the positive edges of those clock signals. If the phase error is within the specified error window then an “in_phase” pulse is generated, otherwise an “out_of_phase” pulse is generated. The number of consecutive “in_phase” and “out_of_phase” pulses is accumulated and is used to determine the state of the lock output.

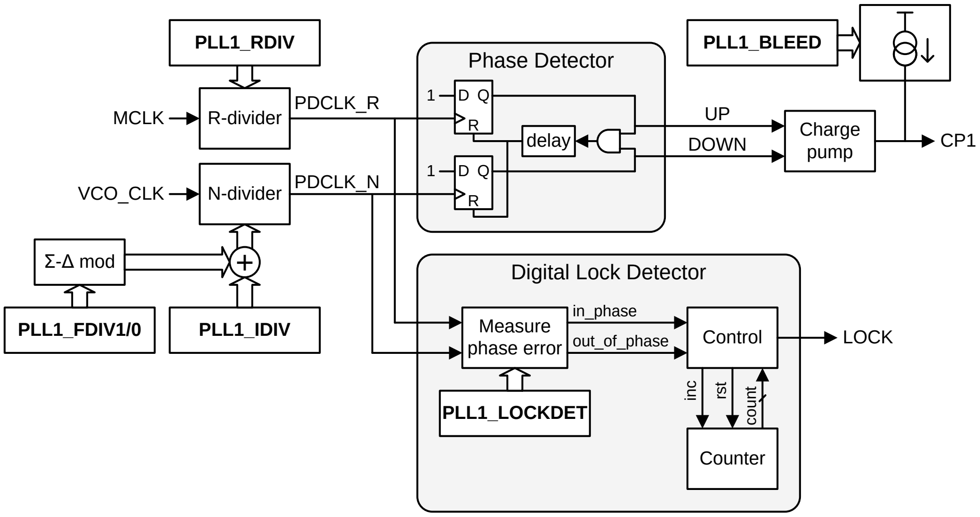


Figure 22 Digital Lock Detector

Typical R-divider and N-divider output waveforms are shown in Figure 23, along with the state transition diagram of the lock detector. The lock signal goes active after a number of consecutive “in_phase” pulses are received by the control logic, defined by the lock threshold (PLL1_LOCKDET bits 4-0); the lock signal subsequently goes inactive after a number of consecutive “out_of_phase” pulses are received, defined by the loss-of-lock threshold (PLL1_LOCKDET bits 7-5). Note that when the PLL is out of lock (lock = 0), the error window width is set by the “lock window” value in PLL1_LOCKDET bits 10-8; when the PLL is in lock (lock = 1), the error window width is increased by the “loss-of-lock” multiplier in PLL1_LOCKDET bits 12-11.The lock status is communicated to the host µC through the PLL1_STATUS register.

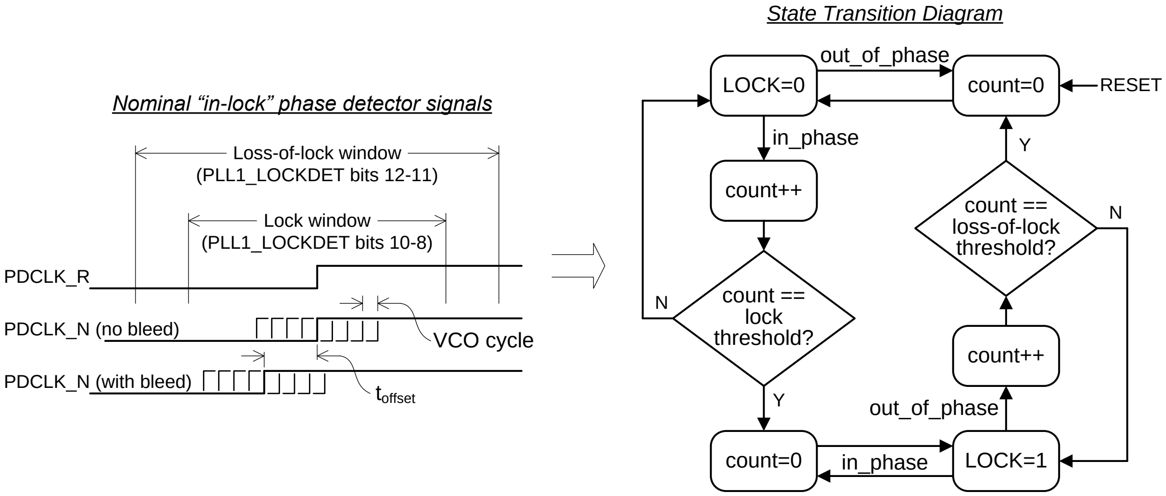


Figure 23 Digital Lock Detector State Transitions

In fractional-N mode the position of the N-divider edge varies by up to ±4 cycles of the VCO clock due to the action of the sigma-delta modulator, and has an extra offset of up to ±0.5 cycles depending on the fractional division value in PLL1_FDIV1/0. The effect of applying a bleed current is also shown in Figure 23; this increases the phase error as the PLL feedback loop compensates for the extra charge added on each phase detector cycle. Any leakage current on the CP1 pin adds a further shift – the source of this leakage current may be the on-chip current sources in the charge pump, or the off-chip loop filter or VCO components. The total amount of shift caused by the bleed and leakage current is given by

$$t_{offset} \cong \frac{R_{div}\left(I_{bleed}+I_{leakage}\right)}{f_{MCLK} I_{CP1}} \quad \text{(units are Amps, Hertz, seconds)}$$

where R_{div} is the MCLK division value set by PLL1_RDIV, $I_{leakage}$ is the leakage current being sourced into the CP1 pin and I_{CP1} is the charge pump current setting. When configuring the digital lock detector the selected lock window width must be large enough to encompass the maximum expected phase error, with sufficient margin to give reliable lock detector operation in the presence of noise. A good starting point is to set the lock window approximately 50% larger than the maximum expected phase error:

$$Lock\ window \approx \pm 1.5 \times \left(\frac{4.5}{f_{VCO}} + \left|t_{offset}\right|\right)$$

The digital lock detector performance can be further optimised by adjusting the lock threshold and loss-of-lock threshold in PLL1_LOCKDET bits 7-0. Setting these thresholds to larger values makes the lock signal less liable to glitch as the PLL acquires lock and less sensitive to noise when in lock, but less responsive if the PLL loses lock.

12.3.2 Analogue Lock Detector

Analogue lock detector mode is selected when PLL1_LOCKDET bit 14 = 1. The analogue lock detector is connected to the phase detector outputs as shown in Figure 24.

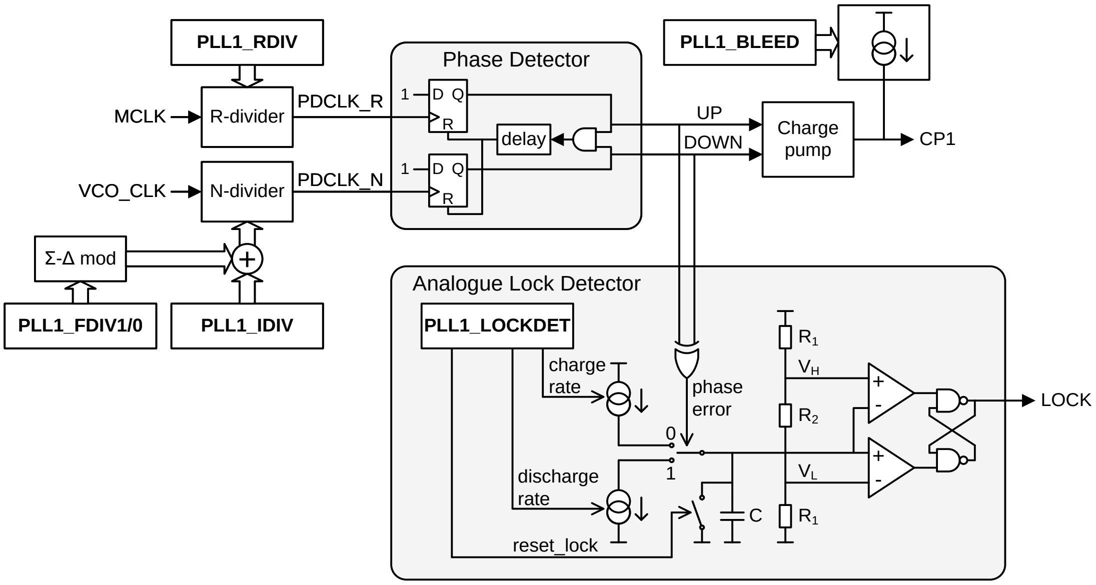


Figure 24 Analogue Lock Detector

The output of the exclusive-OR gate in the analogue lock detector pulses every phase detector cycle, going high on the first rising edge of the two phase detector inputs and going low on the rising edge of the other input. The duration of this pulse is a measure of the phase error of the PLL loop, with a longer pulse representing a larger phase error. The pulse is used to control the charging and discharging of a capacitor. When the phase error signal is low the capacitor charges up, and when the phase error signal is high the capacitor discharges.

When the average PLL phase error is small enough, an overall positive current is supplied to the capacitor and its voltage increases. When the capacitor voltage eventually exceeds the upper threshold in the comparator circuit the lock output goes high, indicating that the PLL is in lock. If the PLL phase error subsequently increases so that the overall current supplied to the capacitor is negative, the capacitor voltage decreases. When the capacitor voltage falls below the lower threshold in the comparator circuit the lock output goes low, indicating that lock has been lost.

PLL1_LOCKDET bits 3-0 control the time taken for the capacitor voltage to increase from 0V to the upper comparator threshold voltage V_H, in the absence of any phase error in the PLL loop. This value should be set to some multiple of the phase detector cycle time

$$t_{charge} \approx L\left(\frac{R_{div}}{f_{MCLK}}\right)$$

where L is approximately the number of "in phase" cycles that must occur before the lock signal goes high. Typically, the value of L will be set to greater than 30 in order to achieve reliable lock detector operation.

PLL1_LOCKDET bits 6-4 set the discharge rate of the capacitor, which is defined as a multiple of the charge rate. Note that the discharge current is always higher than the charge current. To configure the discharge rate the average "in-lock" phase error must be calculated. In the absence of any bleed or leakage current the phase error varies from cycle to cycle (in fractional-N mode) by an average of approximately 2 VCO cycles. Adding bleed or leakage current will increase the average phase error because it shifts the position of the N-divider output with respect to the R-divider output. The amount of this shift, as described in section 12.3.1, is

$$t_{offset} \cong \frac{R_{div}\left(I_{bleed} + I_{leakage}\right)}{f_{MCLK} I_{CP1}}$$

For the purpose of calculations the average "in-lock" phase error is approximately

$$t_{phase\_err} \cong \sqrt{{t_{offset}}^2 + \left(\frac{2}{f_{VCO}}\right)^2}$$

In order for the lock detector capacitor voltage to increase when the PLL is in lock, the average charge supplied to the capacitor on each cycle must be greater than the charge removed. To achieve this, the discharge factor set by PLL1_LOCKDET bits 6-4 should set to:

$$discharge\_rate \approx \frac{1}{M}\left(\frac{R_{div}}{f_{MCLK} t_{phase\_err}} - 1\right)$$

where M is the ratio (when in lock) of the charge added to the capacitor to the charge removed on each phase detector cycle. A good rule is to make M=4, although this value can be adjusted to achieve reliable lock detector operation.

13 Auxiliary ADC and Comparators

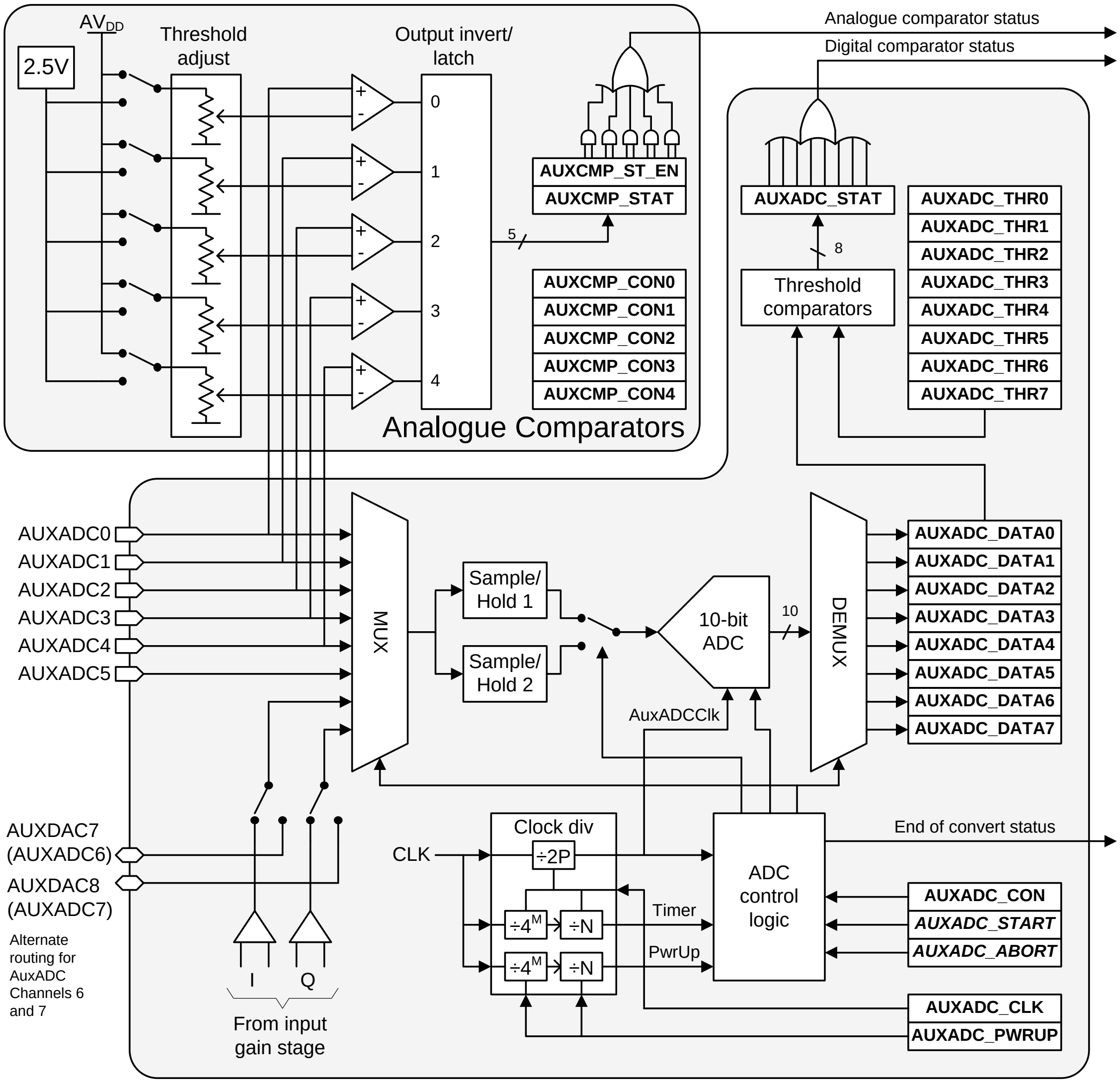


Figure 25 Auxiliary ADC and Comparators

The Auxiliary ADC and Comparators circuit is shown in Figure 25. The auxiliary analogue to digital converter is a 10-bit successive approximation ADC with eight multiplexed inputs. Six of the ADC inputs connect to dedicated input pins and the remaining two inputs can be individually configured to connect to either the main Rx channel I/Q gain stages, or to the AUXDAC7/8 pins. If using either of the AUXDAC7/8 pins as an ADC input, the associated auxiliary DAC cannot be used and must be disabled. Each auxiliary ADC channel includes a digital threshold comparator that can assert a status flag when any of the programmed threshold levels have been exceeded.

There are five analogue comparators that share five of the ADC input signals. Each of these five inputs is compared against an internal voltage threshold, and a separate flag is set when any of the programmed threshold levels have been crossed.

13.1 Auxiliary ADC

Each of the eight auxiliary ADC inputs can be disabled if required. Two sample-and-hold (S/H) circuits are used so that while one channel is being converted, the next channel is charging the opposite S/H. The ADC uses the analogue supply AV_{DD} as a reference – an input value of 0V gives a nominal digital output of 0, and an input value equal to AV_{DD} gives a nominal output of 1023 ($3FF).

The ADC can be configured into one of two conversion modes:

1. **Single shot convert** – when the host processor issues an AUXADC_START command through the C-BUS, a convert sequence (an A/D conversion on each enabled input in ascending order) is performed.
2. **Continuous convert** – Convert sequences are performed repeatedly, under control of an internal timer.

In both conversion modes a status bit is generated at the end of each convert sequence. A separate status bit is generated when the converted data values of selected channels cross a high or low preset threshold value. These status bits appear in the main STATUS register (section 6.4) and can either be polled by the host µC, or used to generate an interrupt signal.

Note that although the “end of convert” indication occurs when all selected channels have completed conversion, each individual result register is updated when the respective channel finishes a conversion. Therefore, in continuous convert mode, there is a limited time after the “end of convert” is asserted before the next convert sequence overwrites the first enabled channel’s data register. The host µC must read the result within this time otherwise the data will be lost.

To save power, the Aux ADC can be configured to automatically power down parts of its analogue circuitry when no channels are selected for conversion, or after a convert sequence (in either single shot or continuous convert mode) has completed on all enabled channels. When this power down mode is selected, a programmable delay must be added at the start of each new convert sequence before the ADC starts doing its first conversion. This delay, controlled by the AUXADC_PWRUP register, allows time for circuits to power up and stabilise and adds directly to the time taken to complete a convert sequence.

The following C-BUS registers are used to configure the auxiliary ADCs:

AUXADC_START - $60

C-BUS command, no data required

When the Aux ADC is configured in single shot convert mode, a convert sequence is initiated when the host µC issues an AUXADC_START command. This command is ignored it the ADC is configured in continuous convert mode, or in single shot mode if a convert sequence is already in progress.

AUXADC_ABORT - $61

C-BUS command, no data required

Issuing this command immediately terminates an active single shot or continuous convert sequence, without generating an “end of convert” status bit. The AUXADC_ABORT command also resets the convert mode to single shot (AUXADC_CON bit 8 = 0), but leaves all other configuration bits unaltered.

AUXADC_CLK - $62: 16-bit Write

Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Aux ADC clock divide						Timer coarse divide			Timer fine divide						

AUXADC_CLK b15-10: Aux ADC clock divide

These bits hold the prescaler value P that determines the frequency of the auxiliary ADC clock 'AuxADCClk'. The value P can be set to between 1 and 64 (000000 = 64); the frequency of AuxADCClk is given by:

$$f_{AuxADCClk} = \frac{f_{CLK}}{2 \times P}$$

AuxADCClk should be set to a frequency of between 1MHz and 2MHz. Each individual ADC conversion takes 11 periods of AuxADCClk to perform, plus one extra cycle of CLK to store the result in the associated data register. If there are N channels enabled, then a convert sequence will be completed in a time given by:

$$T_{Convert} = N\left(\frac{11}{f_{AuxADCClk}} + \frac{1}{f_{CLK}}\right)$$

If the AUXADC_PWRUP register is loaded with a value greater than 0, the convert sequence will be lengthened by that number of AuxADCClk cycles.

AUXADC_CLK b9-7: Timer coarse divide
AUXADC_CLK b6-0: Timer fine divide

The Aux ADC timer determines the rate at which convert sequences are started in continuous convert mode. The coarse divide can be set to a value between 0 and 7, and the fine divide can be set to between 1 and 128 (0000000 = 128). The timer interval is given by this expression:

$$T_{Timer} = \frac{4^{CoarseDivide} \times FineDivide}{f_{CLK}}$$

For instance, if the coarse divide is set to 2 (010) and the fine divide is set to 125 (1111101) then the timer interval will equal 4^2x125 = 2000 cycles of CLK.

The timer period T_{Timer} should be set to a value greater than the convert sequence time $T_{Convert}$, unless back-to-back convert sequences are required in which case set the coarse divide to 000 and the fine divide to 0000001.

AUXADC_PWRUP - $63: 16-bit Write

Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Pwrup enab	0	0	0	0	Power up coarse divide			Power up fine divide							

AUXADC_PWRUP b15: Power-up enable
Set to 1 to enable the power-up delay timer. Set to 0 to disable the delay timer – this should only be done if the sample/hold circuits remain powered up between convert sequences (see register AUXADC_CON).

AUXADC_PWRUP b14-11: Reserved, set to 0
AUXADC_PWRUP b10-8: Timer coarse divide

AUXADC_PWRUP b7-0: Timer fine divide
The power-up divider determines how many cycles of AuxADCClk will occur after a convert sequence is initiated before the conversion actually begins. The coarse divide can be set to a value between 0 and 7, and the fine divide can be set to between 1 and 256 (00000000 = 256). The delay is given by:

$$T_{Powerup} = \frac{4^{CoarseDivide} \times FineDivide}{f_{CLK}}$$

With the Aux ADC configured to power down the sample/hold circuits between convert sequences, this delay allows the circuits time to power back up and stabilise when a new convert sequence begins. The required delay time is defined in section 14.3 (Operating Characteristics).

AUXADC_CON - $64: 16-bit Write
Reset value = $0200

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	ADC7 alt input	ADC6 alt input	“Q” single ended enable	“I” single ended enable	S/H auto-power	Conv. mode	Channel Enable							

AUXADC_CON Register b15-14: Reserved, set to 0

AUXADC_CON Register b13: ADC7 Alternative Input
Set to 1 to connect pin AUXDAC8 to ADC input 7, set to 0 to connect the Q channel differential to single-ended converter to ADC input 7. When using pin AUXDAC8 as an input to the aux ADC, the pin must be put into a high impedance state by disabling Aux DAC8 (clear AUXDAC_DATA8 register b15).

AUXADC_CON Register b12: ADC6 Alternative Input
Set to 1 to connect pin AUXDAC7 to ADC input 6, set to 0 to connect the I channel differential to single-ended converter to ADC input 6. When using pin AUXDAC7 as an input to the aux ADC, the pin must be put into a high impedance state by disabling Aux DAC7 (clear AUXDAC_DATA7 register b15).

AUXADC_CON Register b11: Q-channel single-ended converter enable
Set to 1 to enable the differential to single-ended converter on ADC input 7, and enable the associated input gain stage in Rx channel B (if not already enabled). Set this bit to 0 to power down the converter. The converter cannot automatically power down between convert sequences.

AUXADC_CON Register b10: I-channel single-ended converter enable
Set to 1 to enable the differential to single-ended converter on ADC input 6, and enable the associated input gain stage in Rx channel A (if not already enabled). Set this bit to 0 to power down the converter. The converter cannot automatically power down between convert sequences.

AUXADC_CON Register b9: Sample/hold auto-power
Set this bit to 1 to automatically power down the sample/hold circuits between conversion sequences. Set this bit to 0 to keep the sample/hold circuits powered up between conversion sequences.

AUXADC_CON Register b8: Convert Mode
Set to 0 for single shot convert mode. Issuing a C-BUS AUXADC_START command then starts a single conversion on each of the enabled ADC inputs from lowest to highest.
Set to 1 for continuous convert mode. Convert sequences (A/D conversions on each enabled input from lowest to highest) are then automatically performed at regular intervals, as determined by the timer divide value in the AUXADC_CLK register.

This bit is automatically cleared to 0 by an AUXADC_ABORT command.

AUXADC_CON Register b7-0: Channel Enable
When any of the channel enable bits is set to 1 it causes the corresponding input to be selected for analogue to digital conversion. During a convert sequence each enabled channel, from the lowest to the highest, is converted in turn. The channel enable bits should not be modified while a convert sequence is underway.

Channel Select	*Analogue Input*	*Channel Select*	*Analogue Input*
7	Q input	3	AUXADC3
6	I input	2	AUXADC2
5	AUXADC5	1	AUXADC1
4	AUXADC4	0	AUXADC0

Aux ADC inputs 0 to 5 (pins AUXADC0 to AUXADC5) are direct inputs to the ADC multiplexer.

Inputs 6 and 7 connect to the main ADC channel A and B signal inputs through differential to single-ended converters. These have good common mode rejection and a differential gain of 0.5, so that an input signal of, say, 5V pk-pk differential will give a 2.5V pk-pk output, centred on $AV_{DD}/2$.

AUXADC_THR0 - $65 16-bit Write
AUXADC_THR1 - $66 16-bit Write
AUXADC_THR2 - $67 16-bit Write
AUXADC_THR3 - $68 16-bit Write
AUXADC_THR4 - $69 16-bit Write
AUXADC_THR5 - $6A 16-bit Write
AUXADC_THR6 - $6B 16-bit Write
AUXADC_THR7 - $6C 16-bit Write
Reset value = $00FF

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Aux ADC threshold value B								Aux ADC threshold value A							

At the end of a convert sequence the result for each enabled ADC channel is compared against the two threshold values in the associated threshold register, and a bit in the AUXADC_STAT register is set accordingly (this may also set a bit in the main STATUS register). Only the most significant 8 bits of the Aux ADC values are used for the comparison. Using Aux ADC channel 0 as an example, the associated ADC status bit (AUXADC_STAT bit 0) is set as shown below. The other seven channels operate in a similar way.

(a) If threshold value A (AUXADC_THR0 b7-0) ≥ threshold value B (AUXADC_THR0 b15-8), then the status bit (AUXADC_STAT b0) is set if:
ADC result (AUXADC_DATA0 b9-2) > threshold value A (AUXADC_THR0 b7-0), *OR*
ADC result (AUXADC_DATA0 b9-2) < threshold value B (AUXADC_THR0 b15-8)

(b) If threshold value A (AUXADC_THR0 b7-0) < threshold value B (AUXADC_THR0 b15-8), then the status bit (AUXADC_STAT b0) is set if:
ADC result (AUXADC_DATA0 b9-2) > threshold value A (AUXADC_THR0 b7-0), *AND*
ADC result (AUXADC_DATA0 b9-2) < threshold value B (AUXADC_THR0 b15-8)

Setting the threshold values according to (a) is useful for testing if the ADC value has exceeded a particular range, and (b) is useful for testing if the ADC value has entered a particular range, as shown in Figure 26.

Note that the threshold comparison for a particular channel can be disabled by setting threshold value A = $FF and threshold value B = $00.

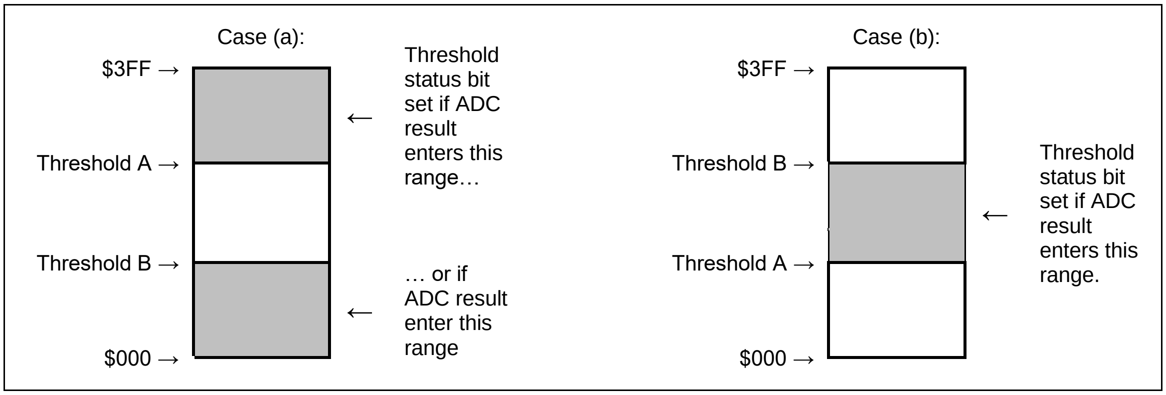

Case (a):
$3FF →
Threshold A →
Threshold B →
$000 →
Threshold status bit set if ADC result enters this range…
… or if ADC result enter this range
Case (b):
$3FF →
Threshold B →
Threshold A →
$000 →
Threshold status bit set if ADC result enters this range.

Figure 26 Auxiliary ADC Threshold Trigger Range

AUXADC_STAT - $6D: 8-bit Read

Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	ADC thresh flag 7	ADC thresh flag 6	ADC thresh flag 5	ADC thresh flag 4	ADC thresh flag 3	ADC thresh flag 2	ADC thresh flag 1	ADC thresh flag 0

AUXADC_STAT Register b7-0: ADC threshold flag 7..0

Each ADC threshold flag gets set to 1 at the end of a convert sequence if the associated ADC channel is enabled and the corresponding ADC result is within the programmed threshold range (see description of AUXADC_THR7..0 registers). The threshold flags in AUXADC_STAT are sticky: once they are set to 1 they remain in that state until a C-BUS read of AUXADC_STAT is performed, after which they are automatically cleared to 0. The eight threshold flags in the AUXADC_STAT register are ORed together and the resulting digital comparator status bit is passed to the CMX983 STATUS register (section 6.4).

AUXADC_DATA0 - $6E: 16-bit Read

AUXADC_DATA1 - $6F; 16-bit Read
AUXADC_DATA2 - $70: 16-bit Read
AUXADC_DATA3 - $71: 16-bit Read
AUXADC_DATA4 - $72; 16-bit Read
AUXADC_DATA5 - $73: 16-bit Read
AUXADC_DATA6 - $74: 16-bit Read
AUXADC_DATA7 - $75: 16-bit Read

Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Aux ADC data									

Each of the eight registers AUXADC_DATA0 to AUXADC_DATA7 holds the 10-bit result of the last conversion performed on the associated channel. The data is updated immediately that the respective channel has been converted during a convert sequence, the registers are *not* updated simultaneously when the convert sequence has completed.

13.2 Auxiliary Comparators

Each of the five analogue comparator channels can be individually enabled. The positive inputs of the comparators are connected to the AUXADC4..0 input pins, and the negative inputs of the comparators are connected to a set of five programmable threshold voltages. The threshold voltages can be individually set in nominal 100mV increments, and can use either the analogue supply or an on-chip bandgap voltage as a reference. Each comparator output can be optionally inverted so that a status flag is generated either when the input pin is above or below the associated threshold voltage.

The following C-BUS registers are used to configure the auxiliary ADCs:

AUXCMP_CON0 - $76: 8-bit Write
AUXCMP_CON1 - $77: 8-bit Write
AUXCMP_CON2 - $78; 8-bit Write
AUXCMP_CON3 - $79: 8-bit Write
AUXCMP_CON4 - $7A: 8-bit Write
Reset value = $00

Bit:	7	6	5	4	3	2	1	0
CON0:	Enab cmp 0	Invert cmp 0	Vref sel 0	Comparator 0 threshold voltage				
CON1:	Enab cmp 1	Invert cmp 1	Vref sel 1	Comparator 1 threshold voltage				
CON2:	Enab cmp 2	Invert cmp 2	Vref sel 2	Comparator 2 threshold voltage				
CON3:	Enab cmp 3	Invert cmp 3	Vref sel 3	Comparator 3 threshold voltage				
CON4:	Enab cmp 4	Invert cmp 4	Vref sel 4	Comparator 4 threshold voltage				

Each of these five registers controls one of the analogue comparators:

AUXCMP_CON0-4 Register b7: Enable comparator
Set to 1 to enable the associated channel (comparator + threshold voltage generator), or set to 0 to disable and powersave the channel. The comparators are prevented from setting their associated flag bit in the AUXCMP_STAT register when they are disabled, or for 256 CLK cycles after the enable bit changes from 0 to 1 (to allow time for the internal threshold voltage to settle).

AUXCMP_CON0-4 Register b6: Invert comparator
Set to 1 to invert the comparator output, causing the comparator flag to be set when the input voltage goes *below* the comparator threshold voltage. Set to 0 to cause the comparator flag to be set when the input voltage goes *above* the comparator threshold voltage.

AUXCMP_CON0-4 Register b5: Voltage reference select
Set to 1 to use the internally generated voltage (2.5V) as the reference for the comparator threshold. Set to 0 to use the analogue supply voltage (AV_{DD}) as the reference for the comparator threshold.

AUXCMP_CON0-4 Register b4-0: Comparator threshold voltage
Set the comparator threshold voltage from 0.0V (00000_2) to 3.1V (11111_2) in 100mV (nominal) steps. Note: if using the internally generated voltage as a reference, then the maximum voltage that can be set is 2.5V (11001_2).

AUXCMP_STAT - $7B: 8-bit Read

Reset value = $00

Bit:

7	6	5	4	3	2	1	0
0	0	0	Cmp thresh flag 4	Cmp thresh flag 3	Cmp thresh flag 2	Cmp thresh flag 1	Cmp thresh flag 0

AUXCMP_STAT Register b7-5: Reserved, set to 0

AUXCMP_STAT Register b4-0: Comparator threshold flag 4..0

Each comparator threshold flag gets set to 1 if the associated comparator is enabled and the corresponding input pin voltage is greater than the programmed threshold voltage (or less than, if the comparator output is inverted). The threshold flags in AUXCMP_STAT are sticky: once they are set to 1 they remain in that state until a C-BUS read of AUXCMP_STAT is performed, after which they are automatically cleared to 0. When AUXCMP_STAT is read, if the comparator input is still greater than the programmed threshold voltage then the flag will stay high. Disabling a comparator channel does not automatically clear the associated flag bit.

The five threshold flags in the AUXCMP_STAT register are gated with the enable bits in the AUXCMP_STAT_EN register and are then ORed together. The resulting analogue comparator status bit is passed to the CMX983 STATUS register (section 6.4), as shown in Figure 27. This can be used to generate an interrupt signal.

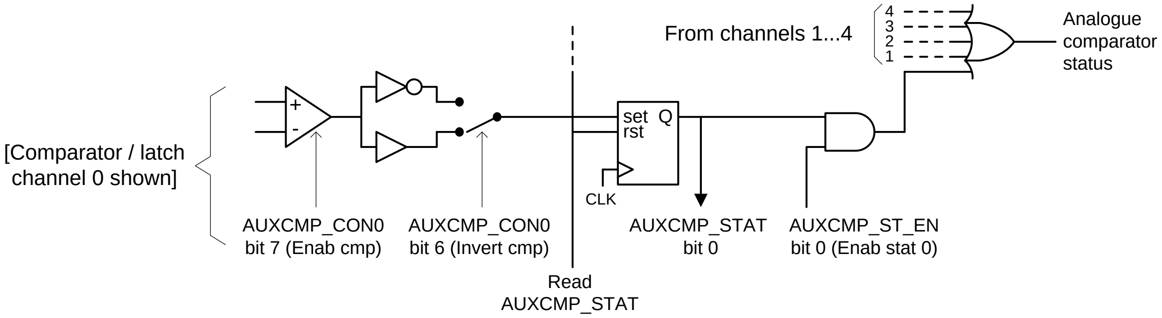

From channels 1...4
4
3
2
1
Analogue comparator status
[Comparator / latch channel 0 shown]
+
-
set Q
rst
CLK
AUXCMP_CON0 bit 7 (Enab cmp)
AUXCMP_CON0 bit 6 (Invert cmp)
AUXCMP_STAT bit 0
AUXCMP_ST_EN bit 0 (Enab stat 0)
Read AUXCMP_STAT

Figure 27 Comparator and Threshold Status Flag

AUXCMP_ST_EN - $7C: 8-bit Write

Reset value = $00

Bit:

7	6	5	4	3	2	1	0
0	0	0	Enab stat 4	Enab stat 3	Enab stat 2	Enab stat 1	Enab stat 0

AUXCMP_ST_EN Register b7-5: Reserved, set to 0

AUXCMP_ST_EN Register b4-0: Enable status 4..0

Setting any of the enable status bits to 1 allows the associated comparator threshold flag in the AUXCMP_STAT register to drive the analogue comparator status bit to 1.

14 Auxiliary DACs

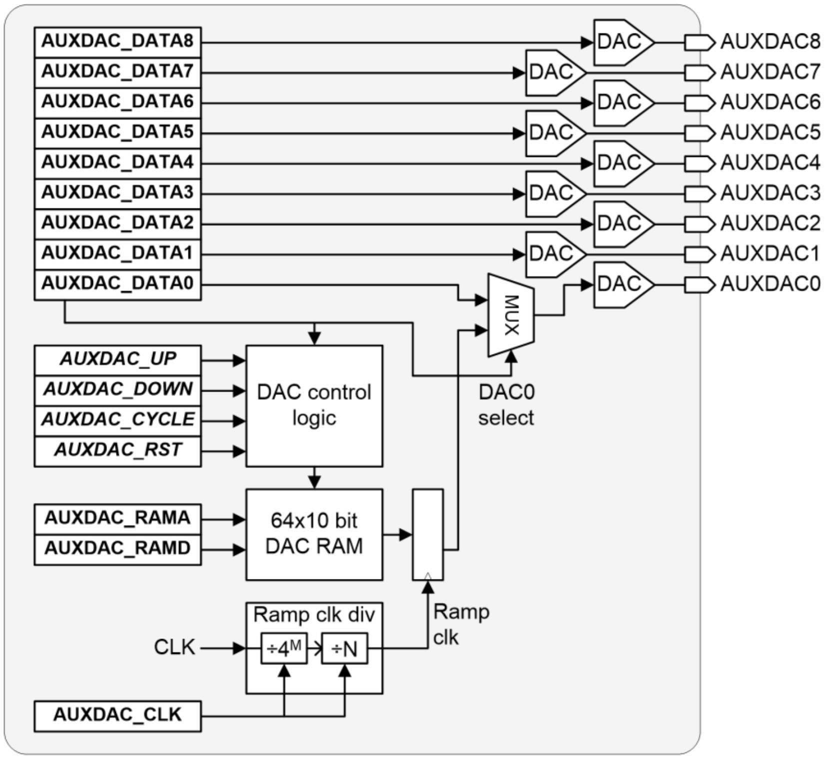


Figure 28 Auxiliary DACs

The CMX983 has nine general-purpose 10-bit D/A converters (pins AUXDAC0 – AUXDAC8) to assist in a variety of control functions (Figure 28). These Aux DACs operate independently, and can be individually enabled or powered down. The Aux DACs are designed to provide an output as a proportion of the analogue supply voltage, depending on the Aux DACs data register setting: a value of 0 drives that Aux DACs output to AV_{SS}; a value of 1023 ($3FF) drives the output to AV_{DD}.

DAC0 has an additional ramping feature where the contents of an internal 64 word × 10 bit DAC RAM can be transferred in ascending order to DAC0 (ramp up), in descending order (ramp down) or repeatedly up and down (cyclical ramping) at a programmable rate. The DAC0 ramp up and ramp down facility is particularly useful for controlling the power of an RF transmitter at the beginning and end of a transmit slot, in order to minimise adjacent-channel splatter.

The following C-BUS registers are used to configure the auxiliary DACs:

AUXDAC_CLK - $82: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	Ramp coarse divide			Ramp fine divide							

AUXDAC_CLK Register b15-11: Reserved, set to 0

AUXDAC_CLK Register b10-8: Ramp coarse divide
AUXDAC_CLK Register b7-0: Ramp fine divide

The ramp clock divider determines the rate at which the 64-word DAC RAM is read during a ramp sequence – a total of 63 reads are performed during a ramp up or ramp down. The coarse divide can be set to a value between 0 and 7, and the fine divide can be set to between 1 and 256 (00000000 = 256). The total time taken to ramp up or down is given by the following expression:

$$T_{RAMP} = 63 \times \left(\frac{4^{CoarseDivide} \times FineDivide}{f_{CLK}} \right)$$

For instance, if the coarse divide is set to 2 (010) and the fine divide is set to 15 (00001111) then the ramp time will equal $63 \text{x} 4^2 \text{x} 15$ = 15120 cycles of CLK. During cyclical ramping, the period of a single output cycle will equal $2\text{x}T_{RAMP}$.

AUXDAC_RAMD - $83: 16-bit Write, data-streaming

Reset value = undefined

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Aux DAC RAM values									

The 64-word DAC RAM can be loaded in ascending order by repeatedly writing data to this C-BUS location (the internal address pointer automatically increments after each write). To increase the loading rate of the coefficients, data-streaming operation is supported for this C-BUS address. Before loading the DAC RAM, the internal address pointer needs to be initialised (usually to address 0); this is done by writing to register AUXDAC_RAMA. Note that writes to the DAC RAM are disabled if the DAC0 select bit (AUXDAC_DATA0 bit 12) is set to 1.

AUXDAC_RAMA - $84: 8-bit Write

Reset value = $00

Bit:	7	6	5	4	3	2	1	0
	0	0	DAC RAM address pointer					

The DAC RAM address pointer determines the address at which data gets written during a C-BUS write to the auxiliary DAC RAM. The address pointer automatically increments after each 16-bit value is written, so if the RAM is written in an ascending sequence the pointer only needs to be initialised once before the RAM is loaded.

AUXDAC_UP - $85

C-BUS command, no data required

When the AUXDAC_UP command is issued, the DAC RAM contents are read out in an ascending sequence and applied to DAC0 (through the multiplexer) until the final location at address 63 has been read.

AUXDAC_DOWN - $86
C-BUS command, no data required

When the AUXDAC_DOWN command is issued, the DAC RAM contents are read out in a descending sequence and applied to DAC0 (through the multiplexer) until the final location at address 0 has been read.

AUXDAC_CYCLE - $87
C-BUS command, no data required

When the AUXDAC_CYCLE command is issued, the DAC RAM will cycle continuously between ramp up and ramp down operations. The rate at which data is read from the RAM is determined by the AUXDAC_CLK register, and it takes a total of 2x63 = 126 reads to complete one up + one down cycle. This mode of operation can be terminated by issuing an AUXDAC_UP, AUXDAC_DOWN or AUXDAC_RST command, or by setting AUXDAC_DATA0 bit 12 to 0.

AUXDAC_RST - $88
C-BUS command, no data required

When the AUXDAC_RST command is issued, any active ramp operation is immediately terminated and the DAC RAM pointer is reset to 0.

AUXDAC_DATA0 - $89: 16-bit Write
AUXDAC_DATA1 - $8A: 16-bit Write
AUXDAC_DATA2 - $8B: 16-bit Write
AUXDAC_DATA3 - $8C: 16-bit Write
AUXDAC_DATA4 - $8D: 16-bit Write
AUXDAC_DATA5 - $8E: 16-bit Write
AUXDAC_DATA6 - $8F: 16-bit Write
AUXDAC_DATA7 - $90: 16-bit Write
AUXDAC_DATA8 - $91: 16-bit Write
Reset value = $0000

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC0:	Enab	Ramp revrse	Ramp hold	DAC0 select	0	0	Aux DAC0 data									
DAC1:	Enab	0	0	0	0	0	Aux DAC1 data									
DAC2:	Enab	0	0	0	0	0	Aux DAC2 data									
DAC3:	Enab	0	0	0	0	0	Aux DAC3 data									
DAC4:	Enab	0	0	0	0	0	Aux DAC4 data									
DAC5:	Enab	0	0	0	0	0	Aux DAC5 data									
DAC6:	Enab	0	0	0	0	0	Aux DAC6 data									
DAC7:	Enab	0	0	0	0	0	Aux DAC7 data									
DAC8:	Enab	0	0	0	0	0	Aux DAC8 data									

AUXDAC_DATA0-8 Register b15: DAC0 – DAC8 enable
Setting any of these bits to 1 enables the corresponding Aux DAC circuit and causes a voltage to be driven onto its output pin. When a DAC is disabled it goes into a zero-power state and its output pin goes high impedance.

AUXDAC_DATA0 Register b14: Ramp reverse mode
This bit determines the behaviour of DAC0 during a ramp operation when a C-BUS command is issued that reverses the direction of the ramp, i.e. issuing an AUXDAC_UP command while the DAC is ramping down, or issuing an AUXDAC_DOWN command while the DAC is ramping up. When this bit is set to 0, the ramp direction reverses immediately. When this bit is set to 1, the ramp direction reverses only when the currently active ramp operation completes.

AUXDAC_DATA0 Register b13: Ramp hold
Set this bit to 1 to pause a ramp up, ramp down or ramp cycle operation; this freezes the DAC0 output. Setting this bit to 0 allows the ramp function to continue.

AUXDAC_DATA0 Register b12: DAC0 select
This bit controls the multiplexer at the input to DAC0. When this bit is set to 1, the DAC RAM output register is connected to DAC0 and C-BUS writes to the DAC RAM are disabled. When this bit is set to 0, the AUXDAC_DATA0 register is connected to DAC0 and any active ramp operation is immediately terminated.

AUXDAC_DATA1-8 Register b14-12: Reserved, set to 0

AUXDAC_DATA0-8 Register b11-10: Reserved, set to 0

AUXDAC_DATA0-8 Register b9-0: DAC0 – DAC8 data
The least significant 10-bit values in registers AUXDAC_DATA1 – AUXDAC_DATA8 are driven directly into the corresponding auxiliary DACs. The least significant 10-bit value in register AUXDAC_DATA0 is multiplexed with the output of the ramp circuit before being driven into DAC0; this multiplexer is controlled by AUXDAC_DATA0 bit 12.

An example of the Aux DAC RAM contents for a raised cosine ramp profile is shown in Figure **29**.

Aux DAC RAM contents (hexadecimal)															
0 000	1 001	2 003	3 006	4 00A	5 010	6 017	7 01F	8 028	9 033	10 03E	11 04B	12 059	13 068	14 078	15 089
16 09A	17 0AD	18 0C1	19 0D5	20 0EA	21 100	22 116	23 12D	24 145	25 15D	26 175	27 18E	28 1A7	29 1C0	30 1D9	31 1F3
32 20C	33 226	34 23F	35 258	36 271	37 28A	38 2A2	39 2BA	40 2D2	41 2E9	42 2FF	43 315	44 32A	45 33E	46 352	47 365
48 376	49 387	50 397	51 3A6	52 3B4	53 3C1	54 3CC	55 3D7	56 3E0	57 3E8	58 3EF	59 3F5	60 3F9	61 3FC	62 3FE	63 3FF

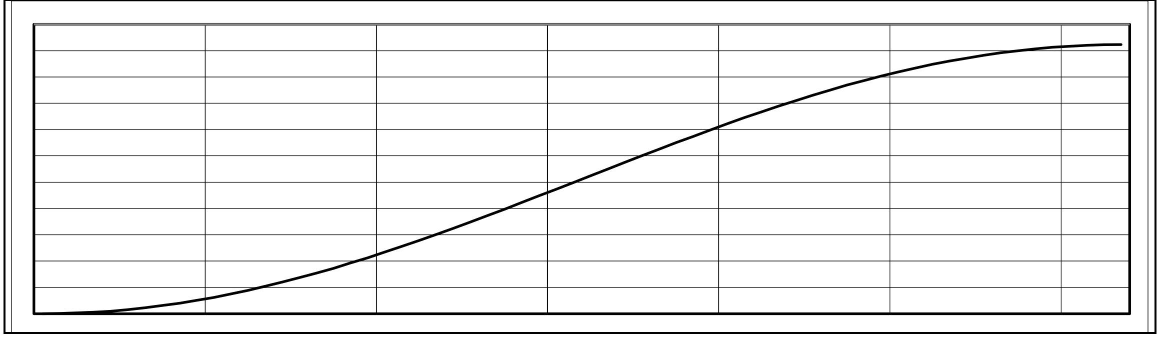


Figure 29 **Aux DAC RAM contents example**

15 Performance Specification

15.1 Absolute Maximum Ratings

Exceeding these maximum ratings can result in damage to the device.

	Min.	Max.	Units
Power Supplies			
IOV_{DD} - IOV_{SS}	-0.3	4.0	V
DV_{DD} - DV_{SS}	-0.3	2.16	V
AV_{DD} - AV_{SS}	-0.3	4.0	V
$RF1V_{DD}$ - AV_{SS}	-0.3	2.16	V
$RF2V_{DD}$ - AV_{SS}	-0.3	2.16	V
$CP1V_{DD}$ - AV_{SS}	-0.3	6.0	V
$CP2V_{DD}$ - AV_{SS}	-0.3	6.0	V
Voltage differential between power supplies:			
IOV_{SS}, DV_{SS} and AV_{SS}	0	50	mV
Voltage on any pin to DV_{SS}	-0.3	IOV_{DD} + 0.3	V
Current into or out of any pin, except power supply pins	-20	+20	mA

Q1 Package (64-pin VQFN)	Min.	Max.	Units
Total Allowable Power Dissipation at T_{AMB} = 25 C		3500	mW
... Derating		35.0	mW/°C
Storage Temperature	-55	+125	C
Operating Temperature	-40	+85	°C

15.2 Operating Limits

Correct operation outside these limits is not implied.

	Min	Typ	Max.	Units
IOV_{DD} - IOV_{SS}	3.0	3.3	3.6	V
DV_{DD} - DV_{SS}	1.7	1.8	1.9	V
AV_{DD} - AV_{SS}	3.0	3.3	3.6	V
$RF1V_{DD}$ - AV_{SS}	1.7	1.8	1.9	V
$RF2V_{DD}$ - AV_{SS}	1.7	1.8	1.9	V
$CP1V_{DD}$ - AV_{SS}	3.0	5.0	5.25	V
$CP2V_{DD}$ - AV_{SS}	3.0	5.0	5.25	V
Voltage differential between power supplies:				
IOV_{SS}, DV_{SS} and AV_{SS}	0	–	50	mV
Operating Temperature	-40	–	+85	°C
Master Clock Frequency (MCLK)	5.0	–	30	MHz
Internal Clock Frequency (CLK)	5.0	–	50	MHz

It is recommended that the 1.8V supplies (DV_{DD}, $RF1V_{DD}$ and $RF2V_{DD}$) be brought up and the RESETN pin driven low before the analogue supply (AV_{DD}) is brought up. The charge pump supplies ($CP1V_{DD}$, $CP2V_{DD}$) can be brought up either before or after the other supplies. If this sequence is not followed then the analogue circuits may power up in an indeterminate or active state, and this state may persist until the 1.8V supplies are brought up and the RESETN pin is driven low. The CMX983 will not suffer physical damage or reliability degradation if the recommended power up sequence is not followed.

15.3 Operating Characteristics

For the conditions in Table 4 below unless otherwise specified:

Note: “NBM” and ”WBM” refer to the Main ADCs or Main DACs operating in respective Normal Bandwidth Mode and Wide Bandwidth Mode and, for the Main ADCs, with RX_BIAS0 and RX_BIAS1 register values set for each of those modes as described in section 9.1.

Table 4 Operational Characteristics - Test Conditions

Power supplies			*General*		
AV_{DD}, IOV_{DD}	3.3	V	f_{MCLK}, f_{CLK}	19.2	MHz
$CP1V_{DD}$, $CP2V_{DD}$	5.0	V	T_{AMB}	25	C
DV_{DD}, $RF1V_{DD}$, $RF2V_{DD}$	1.8	V			
Main ADC and Rx channel configuration			*Main DAC and Tx channel configuration*		
Signal tone	1010	Hz	Signal tone	1010	Hz
Signal amplitude, differential	3.0	Vp-p	Input code level, peak	±30178	
Rx input gain	0	dB	Input sample rate (CT1)		
Σ-Δ clock rate (CR1)			NBM	150	kHz
NBM	2.4	MHz	WBM	300	kHz
WBM	4.8	MHz	Upsampler 1 rate	Bypass	
Sinc filter length	16		FIR filter	Bypass	
Sinc filter no.	5		Bit selector	Bypass	
Bit selector 1	17		Upsampler 2 rate	16	
Decimator 1 rate	16		Σ-Δ clock rate (CT3)		
FIR filter	Bypass		NBM	2.4	MHz
Bit selector 2	1		WBM	4.8	MHz
Decimator 2 rate	1		SC filter bandwidth	Low	
Output sample rate (CR3)	150	kHz	Tx output gain	0	dB
Measurement bandwidth			External RC filter -3dB	100	kHz
NBM	9	kHz	Measurement bandwidth		
WBM	18	kHz	NBM	9	kHz
			WBM	18	kHz

DC Parameters	**Notes**	**Min.**	**Typ.**	**Max.**	**Unit**
Supply Current – Powersave	1				
IOI$_{DD}$		–	10	–	µA
DI$_{DD}$		–	100	–	µA
AI$_{DD}$		–	20	–	µA
RF1I$_{DD}$ or RF2I$_{DD}$		–	10	–	µA
CP1I$_{DD}$ or CP2I$_{DD}$		–	10	–	µA
Supply Current – Idle (CLK running)					
DI$_{DD}$		–	0.83	–	mA
AI$_{DD}$	6	–	0.52	–	mA
Supply Current – I and Q Tx Channels Active					
IOI$_{DD}$	1, 2	–	1.7	–	mA
DI$_{DD}$	1, 2	–	6.5	–	mA
DI$_{DD}$	1, 5	–	5.7	–	mA
AI$_{DD}$	1, 2	–	6.5	–	mA
Supply Current – I and Q Rx Channels Active					
IOI$_{DD}$	1, 2	–	0.8	–	mA
DI$_{DD}$	1, 2	–	7.8	–	mA
DI$_{DD}$	1, 5	–	7.0	–	mA
AI$_{DD}$ – NBM	1, 2	–	7.2	–	mA
AI$_{DD}$ – WBM	1, 2, 13	–	16.0	–	mA
Additional Currents from Idle Mode					
One RF Synthesiser active:					
DI$_{DD}$		–	0.09	–	mA
AI$_{DD}$		–	1.2	–	mA
RF1I$_{DD}$ or RF2I$_{DD}$	3	–	9.2	–	mA
CP1I$_{DD}$ or CP2I$_{DD}$	4	–	0.75	–	mA
Digital PLL active:					
DI$_{DD}$		–	0.47	–	mA
One AuxADC input active:					
DI$_{DD}$		–	0.06	–	mA
AI$_{DD}$		–	0.45	–	mA
One AuxDAC output active:					
DI$_{DD}$		–	0.02	–	mA
AI$_{DD}$		–	0.26	–	mA
V_{BIAS} Buffer active:					
AI$_{DD}$		–	0.11	–	mA

DC Parameters	Notes	Min.	Typ.	Max.	Unit
Digital Interfaces					
Input Logic ‘1’		70%	–	–	IOV_{DD}
Input Logic ‘0’		–	–	30%	IOV_{DD}
Input Leakage Current (Logic ‘1’ or ‘0’)		-1.0	–	1.0	µA
Input Capacitance		–	–	7.5	pF
Output Logic ‘1’ (I_{OH} = 2mA)		90%	–	–	IOV_{DD}
Output Logic ‘0’ (I_{OL} = -5mA)		–	–	10%	IOV_{DD}
“Off” State Leakage Current		-1.0	–	1.0	µA
V_{BIAS}					
Output voltage offset wrt $AV_{DD}/2$ (I_{OL} < 1µA)		–	±2%	–	AV_{DD}
Output impedance		–	50	–	kΩ
Start-up time	7	–	10.5	–	ms
V_{BBUF} (V_{BIAS} Buffer)					
Output load current		–	–	50	µA

Notes:
1. T_{AMB} = 25°C, not including any current drawn from the device pins by external circuitry.
2. Includes 2 x RF Synthesisers, digital PLL, AuxADC and AuxDAC.
3. Lock detect active.
4. Per charge pump, with an output current of 100µA.
5. No digital PLL, RF synthesisers or auxiliary circuits.
6. MCLK amplifier active.
7. With 100nF load.

Operating Characteristics (continued)

AC Parameters	Notes	Min.	Typ.	Max.	Unit
Clocks					
MCLK frequency (f_{MCLK})		5	–	30	MHz
MCLK sensitivity (AC-coupled)		0.2	–	0.8	Vpk-pk
MCLK slew rate (AC-coupled)		10	–	–	V/µs
MCLK amplifier phase noise	10	–	-83	–	dBc/Hz
CLK frequency (f_{CLK})		5	–	50	MHz
Synthesiser 1 and 2					
RF input frequency ($f_{RF1,2}$)		100	–	2100	MHz
RF input sensitivity	12	50	–	500	mVpk
RF input slew rate		150	–	–	V/µs
Charge pump sink/source (programmable)		–	25-400	–	µA
Charge pump absolute accuracy		-20	–	20	%
Charge pump matching		-4	–	4	%
Charge pump compliance range		0.5	–	CP1/2 V_{DD} - 0.5	V
PD comparison frequency (f_{COMP})		–	–	30	MHz
N-Divider range (Integer mode)		32	–	2047	
N-Divider range (Fractional mode)		36	–	2043	
1Hz normalised phase noise floor	11	–	-205	–	dBc/Hz
Main ADCs					
Gain stage error		-0.3	0	0.3	dB
Sigma-delta clock rate (CR1) – NBM		–	2.4	2.6	MHz
Sigma-delta clock rate (CR1) – WBM	13	–	4.8	5.0	MHz
SINAD – NBM		87	88	–	dB
SINAD – WBM	13	–	85	–	dB
Input voltage (after gain stage)		20	–	80	%AV_{DD}
SFDR – NBM		–	95	–	dB
SFDR – WBM	13	–	91	–	dB
Zero error (offset)		–	±5.5	±10	mV
Main DACs					
Gain stage error		-0.3	0	0.3	dB
Sigma-delta clock rate (CT3) – NBM		–	2.4	2.6	MHz
Sigma-delta clock rate (CT3) – WBM	13	–	4.8	5.0	MHz
Output voltage		–	–	10 to 90	%AV_{DD}
Resolution (gain = 0dB)		–	75.5	–	µV/bit
Output load (per output, to AV_{DD}/2)		20	–	–	kΩ
SINAD – NBM		72	76	–	dB
SINAD – WBM	13	–	73	–	dB
SFDR – NBM		–	85	–	dB
SFDR – WBM	13	–	85	–	dB
Zero error (offset)		–	±3.0	±10	mV

Notes: 10. MCLK = 19.2MHz sinewave, 400mVpk-pk, measured at 1kHz offset.

11. 1Hz Normalised Phase Noise Floor (PN1Hz) can be used to calculate the phase noise within the PLL loop bandwidth by:

Measured Phase Noise (in 1Hz) = -PN1Hz - 20log$_{10}$(N) - 10log$_{10}$(f$_{comparison}$);
f$_{comparison}$ = Frequency at the output of the reference divider; N = main divider ratio.
Measured with f$_{comparison}$ = 2.4MHz, charge pump current = 400uA, modulator setting = 001.
Value will vary depending on PLL settings.

12. An input sinewave below 477MHz must be greater than 50mVpk amplitude to meet the minimum slew rate specification.
13. The main ADCs operating in Wide Bandwidth Mode with RX_BIAS0 and RX_BIAS1 register values set as described in section 9.1

Operating Characteristics (continued)

AC Parameters	Notes	Min.	Typ.	Max.	Unit
Aux ADC					
Sample/hold startup time		30	–	–	µs
Resolution		–	10	–	bits
Aux ADC clock period		500	–	1000	ns
Integral non-linearity	20	–	–	±2	bits
Differential non-linearity	20, 21	–	–	±1	bits
Zero error (offset)		–	–	±10	mV
Input capacitance		–	–	5	pF
Aux Comparators					
Internal 2.5V reference voltage		2.3	2.5	2.7	V
Comparator offset voltage		–	±6.5	±10	mV
Aux DACs					
Capacitive load on Aux DAC output		–	–	100	pF
Resolution		–	10	–	bits
Settling time to 0.5 LSB	22	–	–	10	µs
Integral non-linearity		–	–	±4	bits
Differential non-linearity	24	–	–	±1	bits
Zero error (offset)		–	±5.5	±12	mV
Output current (individual Aux DAC output)	23	–	–	±2.3	mA
Total output current (sum of all Aux DACs)		–	–	±10	mA
Output noise voltage (100 Hz – 30kHz)		–	30	–	µV rms

Notes: 20. Non-linearity is specified between 0.5% and 99.5% of full scale.
21. Guaranteed monotonic (no missing codes).
22. Worst case large signal transition.
23. Aux DAC output voltage in the range 0.2V to AV$_{DD}$ – 0.2V at maximum output current
24. Not applicable in alternate routing mode

Operating Characteristics (continued)

AC Parameters		**Notes**	**Min.**	**Typ.**	**Max.**	**Unit**
C-BUS Timing						
Input pin rise/fall time (10% - 90% of IOV$_{DD}$)			–	–	3	ns
Capacitive load on RDATA and IRQN			–	–	30	pF
t_{CSE}	CSN enable to SCLK high time		40	–	–	ns
t_{CSH}	Last SCLK high to CSN high time		40	–	–	ns
t_{LOZ}	SCLK low to RDATA output enable time	30	0	–	–	ns
t_{HIZ}	CSN high to RDATA high impedance	30	–	–	30	ns
t_{CSOFF}	CSN high time between transactions		40	–	–	ns
t_{CK}	SCLK cycle time		100	–	–	ns
t_{CH}	SCLK high time		40	–	–	ns
t_{CL}	SCLK low time		40	–	–	ns
t_{CDS}	CDATA setup time		25	–	–	ns
t_{CDH}	CDATA hold time		25	–	–	ns
t_{RDV}	SCLK low to RDATA valid time	31	0	–	35	ns
Serial Port Timing						
Input pin rise/fall time (10% - 90% of IOV$_{DD}$)			–	–	3	ns
Capacitive load on Rx and Tx port output pins			–	–	30	pF
t_{RCK}	RXCLK cycle time		100	–	–	ns
t_{RCKH}	RXCLK high time		40	–	–	ns
t_{RCKL}	RXCLK low time		40	–	–	ns
t_{RCHD}	RXCLK high to RXD or RXFS delay	31	-20	–	20	ns
t_{RCLOZ}	RXCLK high to RXD output enable time	30	0	–	–	ns
t_{RXDS}	RXD or RXFS setup time to RXCLK low		20	–	–	ns
t_{RXDH}	RXD or RXFS hold time from RXCLK low		20	–	–	ns
t_{TCK}	TXCLK cycle time		100	–	–	ns
t_{TCKH}	TXCLK high time		40	–	–	ns
t_{TCKL}	TXCLK low time		40	–	–0	ns
t_{TCHD}	TXCLK high to TXFS delay	31	-20	–	20	ns
t_{TXDS}	TXD or TXFS setup time to TXCLK low		20	–	–	ns
t_{TXDH}	TXD or TXFS hold time from TXCLK low		20	–	–	ns

Notes: 30. Measured with Test Load B on output pin
31. Measured with Test Load A on output pin

15.4 Timing Diagrams

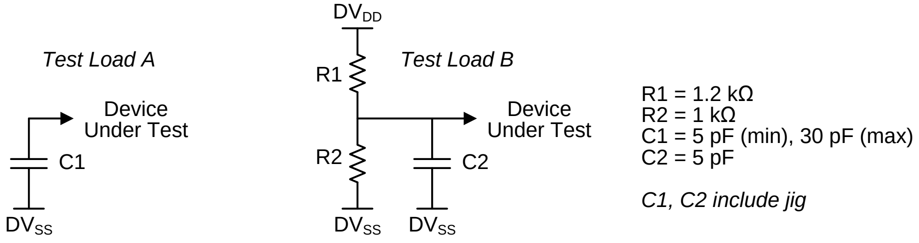


Figure 30 AC Test Load for Digital Outputs

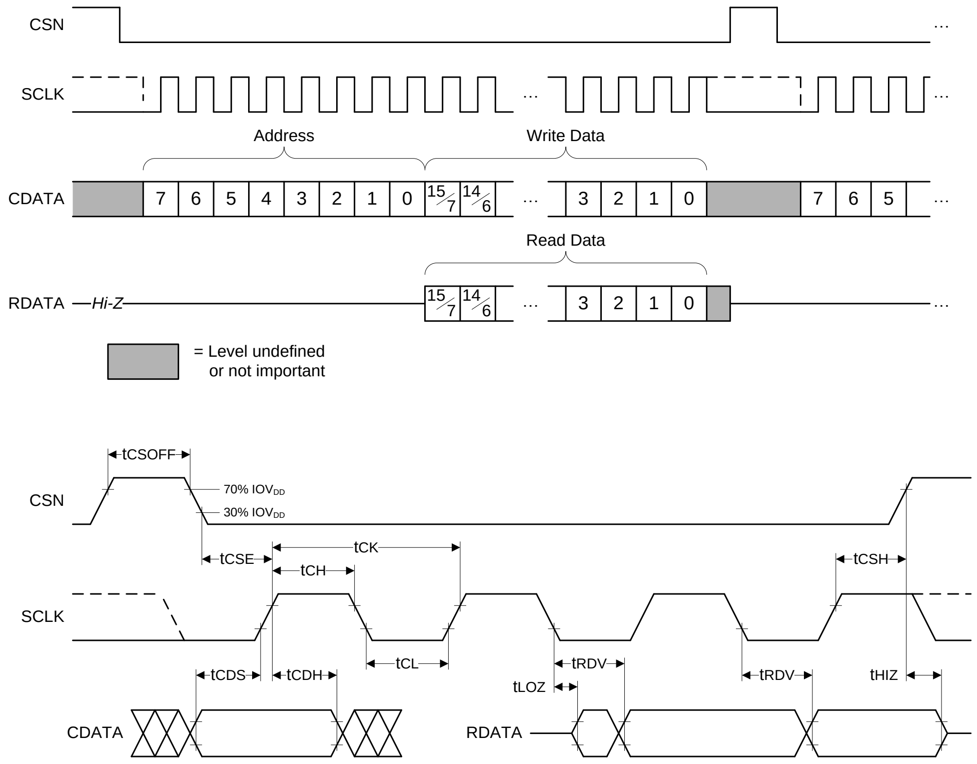


Figure 31 C-BUS Timings

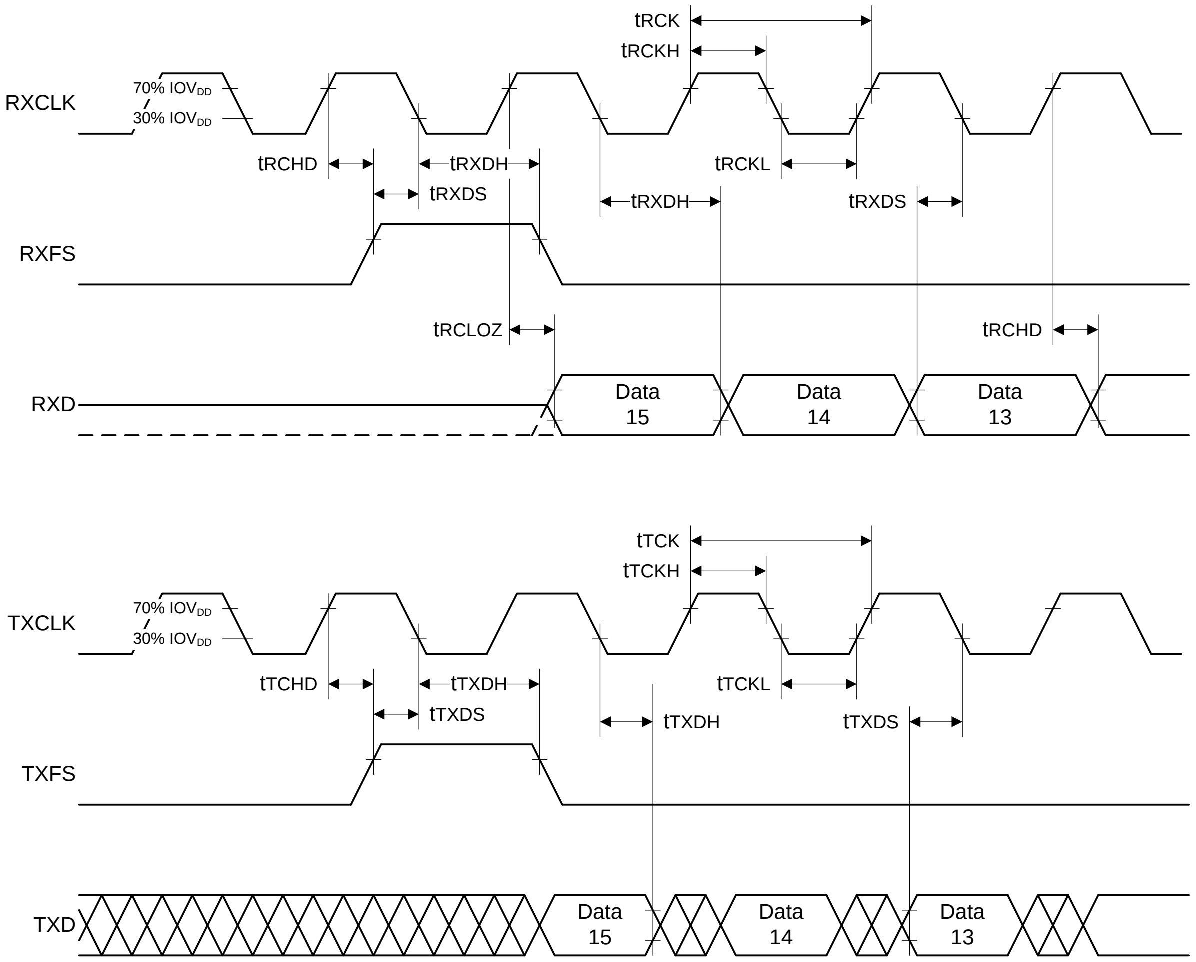


Figure 32 Serial Port Timings

15.5 Typical Performance Characteristics

Refer to Table 4 for details of the test conditions that apply to the following graphs.

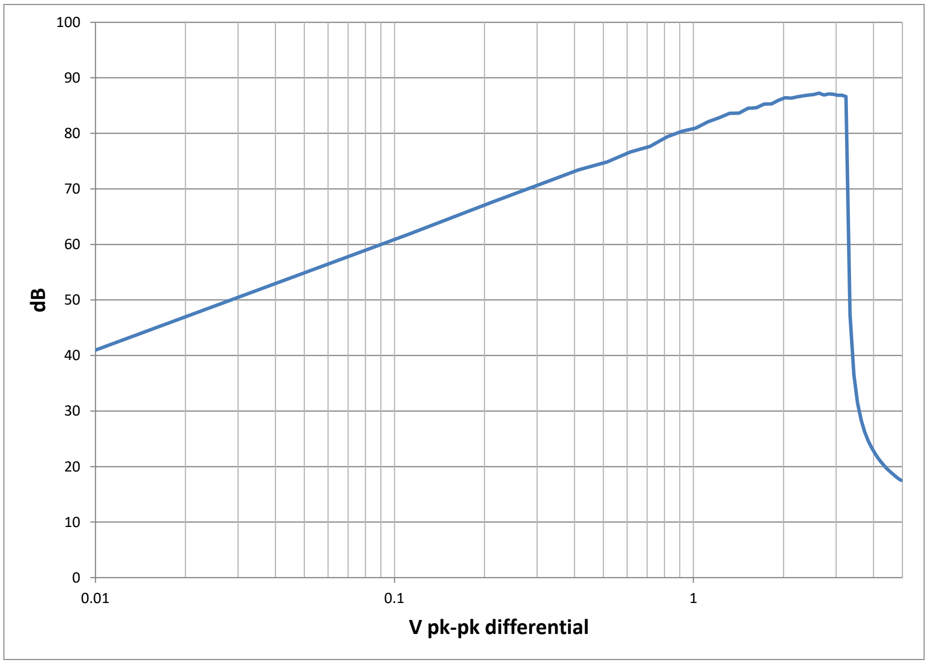


Figure 33 Main ADC SINAD vs. Input Level – NBM

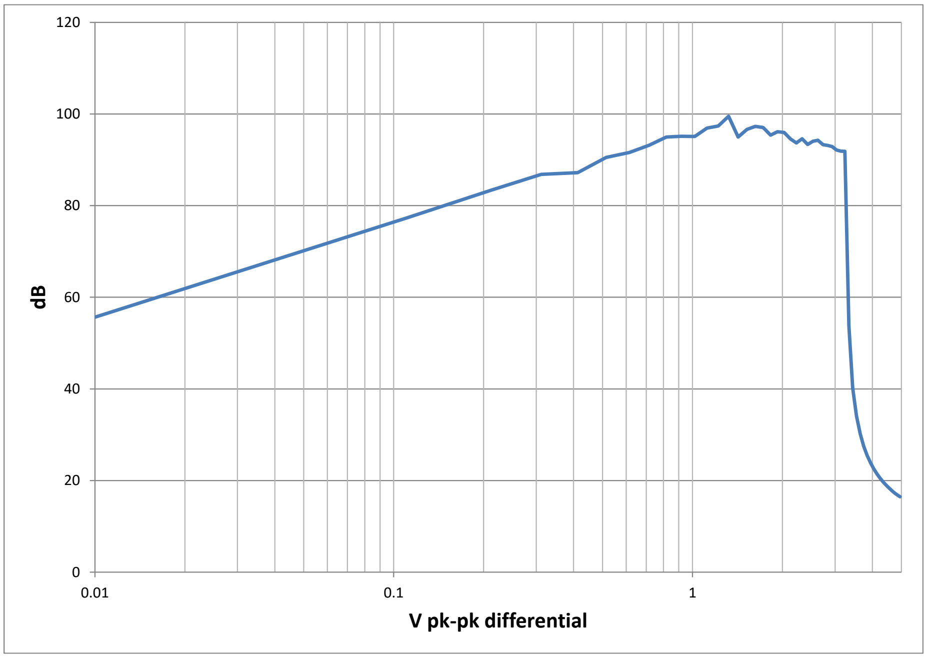


Figure 34 Main ADC SFDR vs. Input Level – NBM

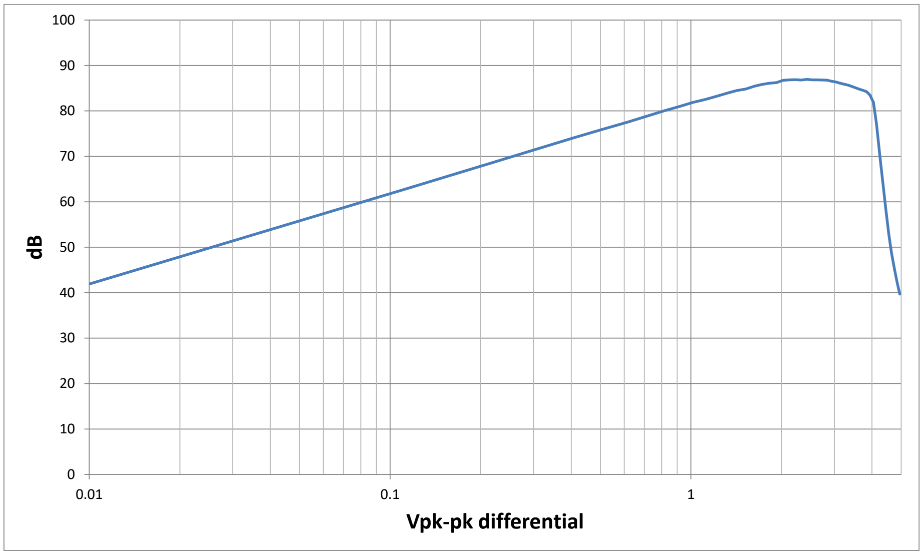


Figure 35 Main ADC SINAD vs. Input Level – WBM

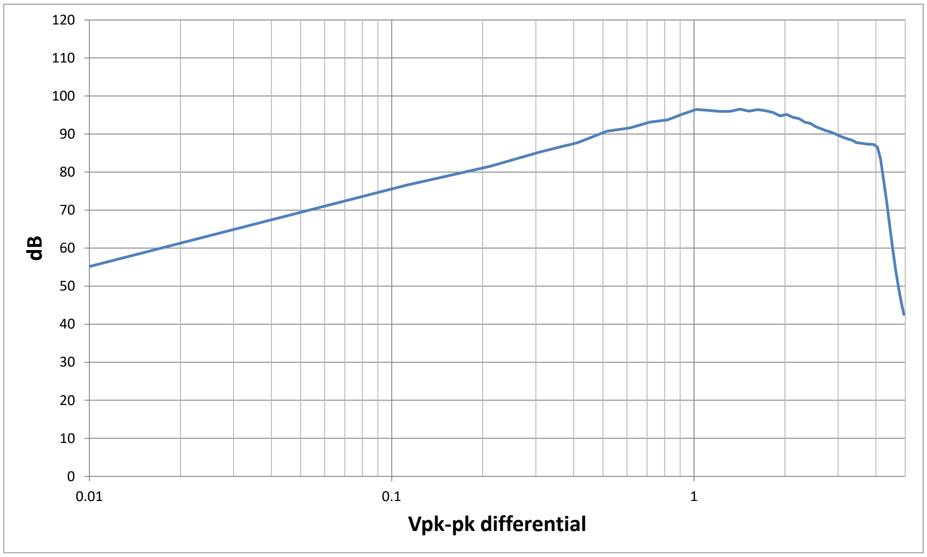


Figure 36 Main ADC SFDR vs. Input Level – WBM

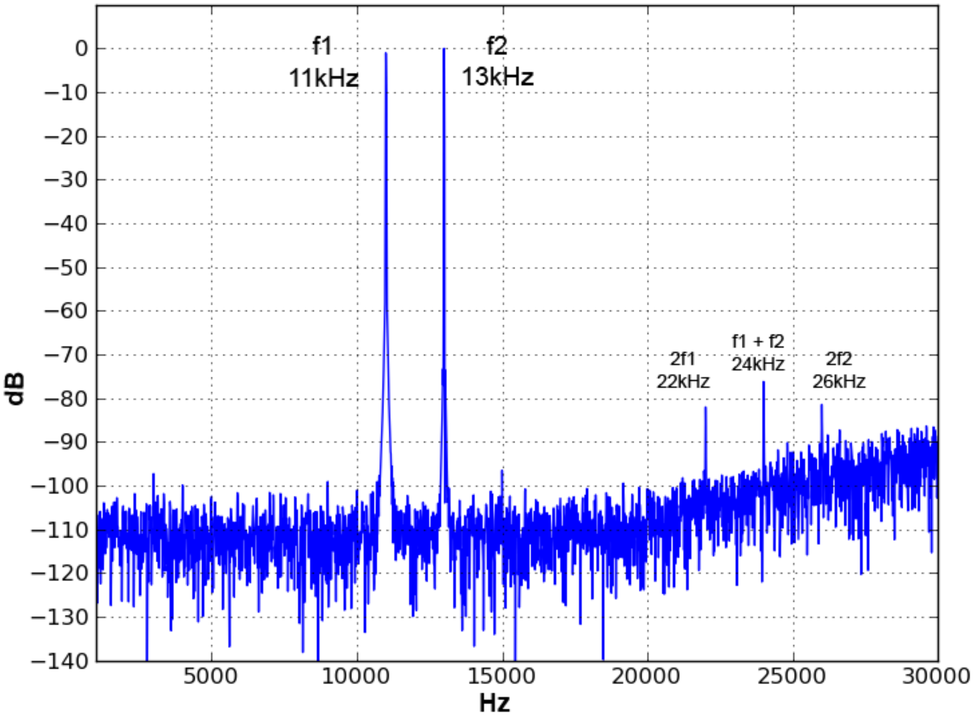


Figure 37 Main ADC Two Tone Test – NBM

Note: For the graph shown in Figure 37, input level was 1.5V pk-pk for each of the two tones.

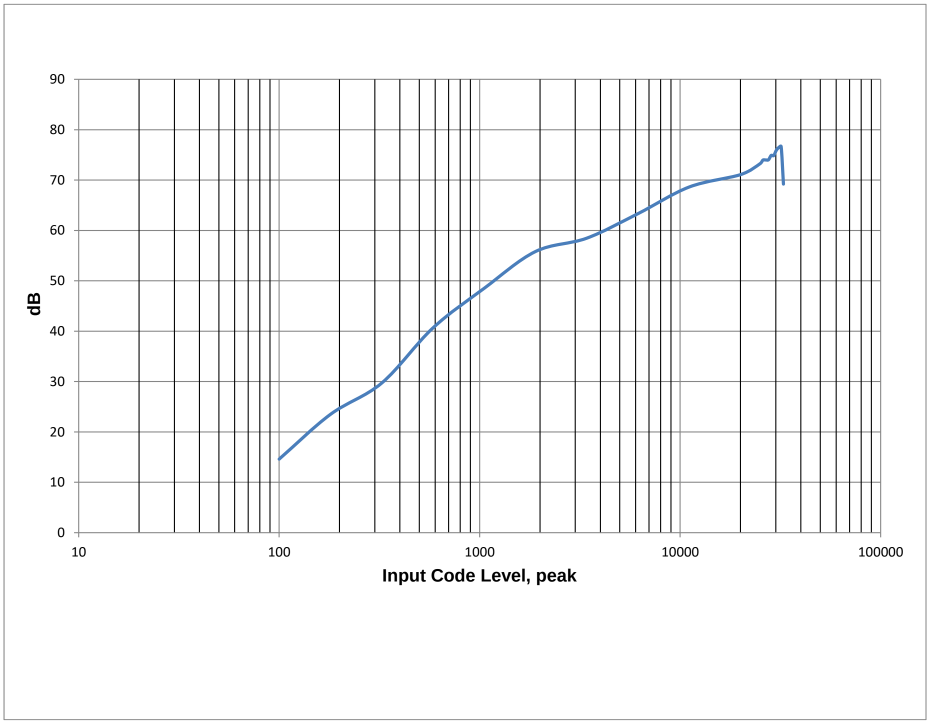


Figure 38 Main DAC SINAD vs. Input Code Level – NBM

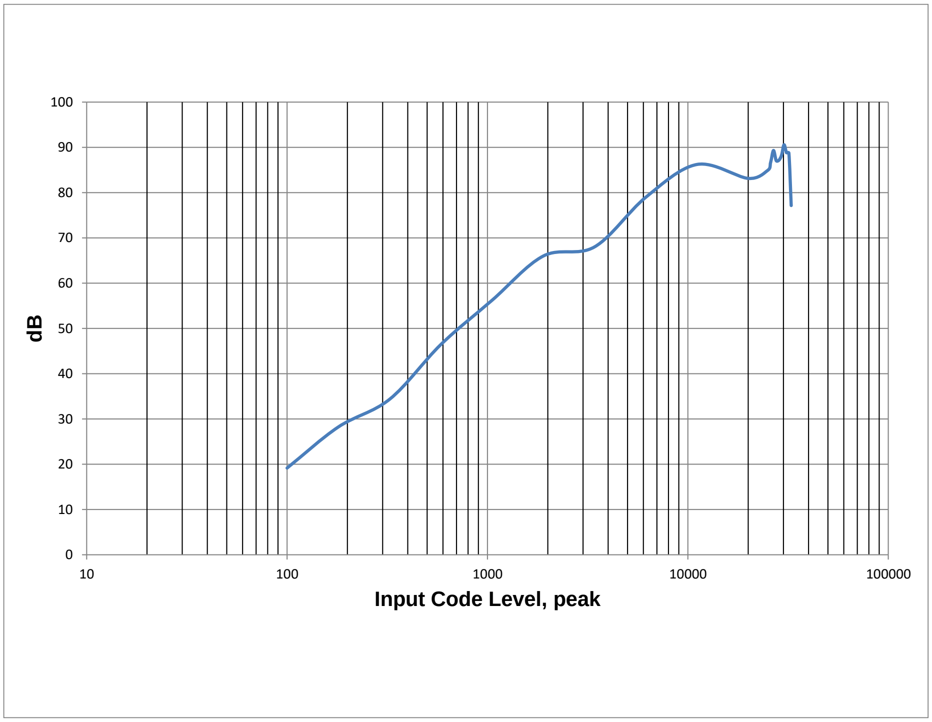


Figure 39 Main DAC SFDR vs. Input Code Level – NBM

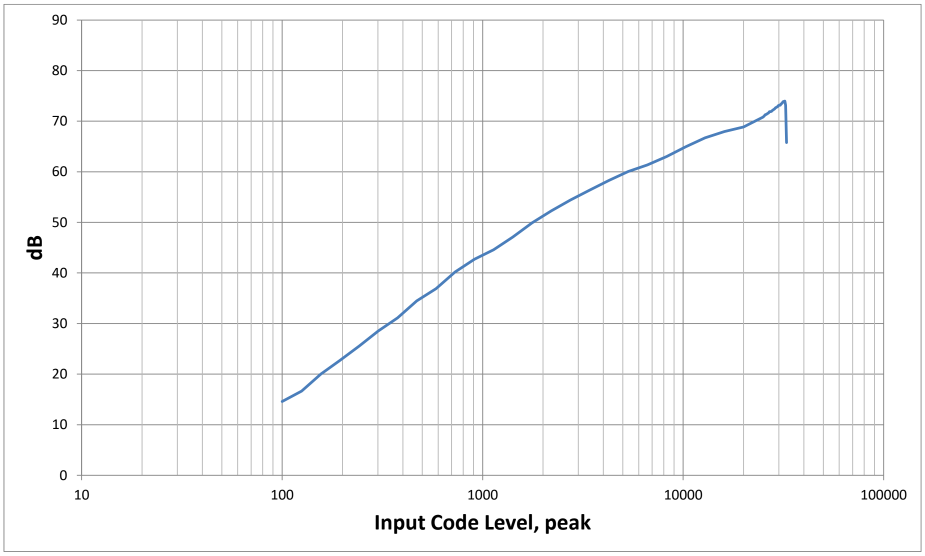


Figure 40 Main DAC SINAD vs. Input Code Level – WBM

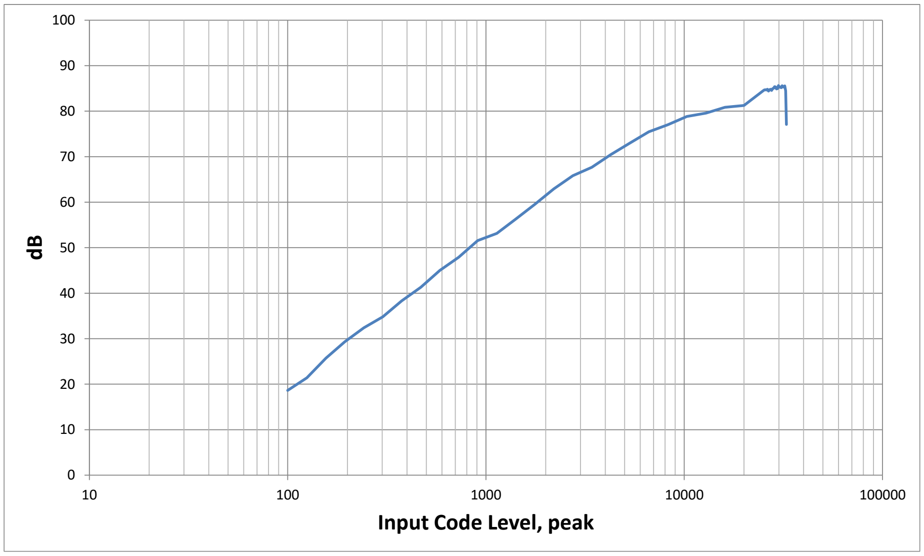


Figure 41 Main DAC SFDR vs. Input Code Level – WBM

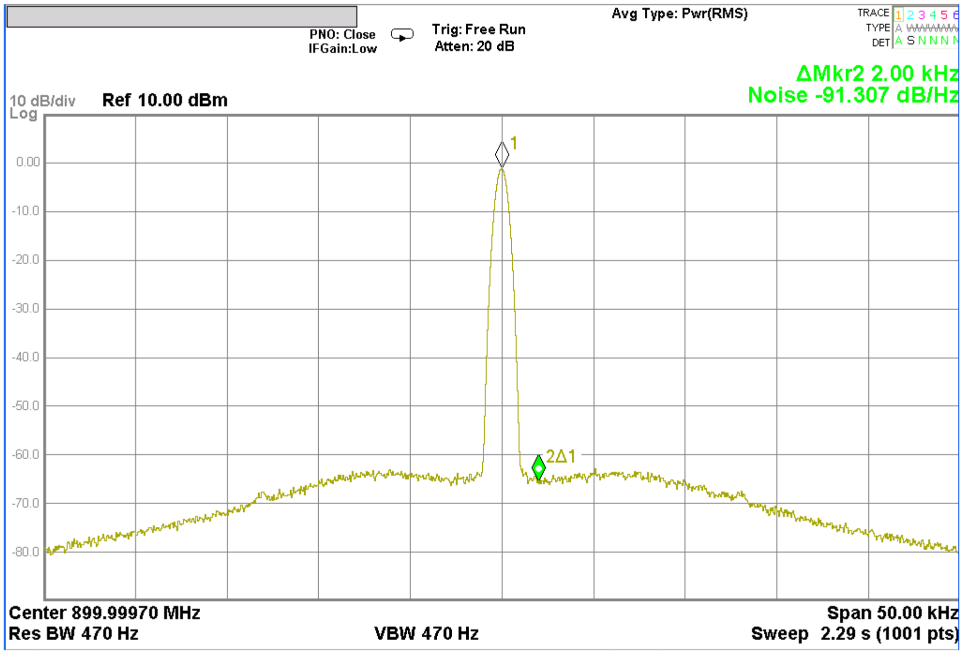


Figure 42 PLL1 Output Spectra

Note: The graph shown in Figure 42 shows the output spectra for f_{COMP} = 4.8MHz (RDIV = $4) and 250μA charge pump current. Without fast lock: Loop Filter (reference design as shown in Figure 21)
C1 = 6n8F / C2 = 27nF / R2 = 1k5ohms / R3 = 470ohms / C3 = 470pF

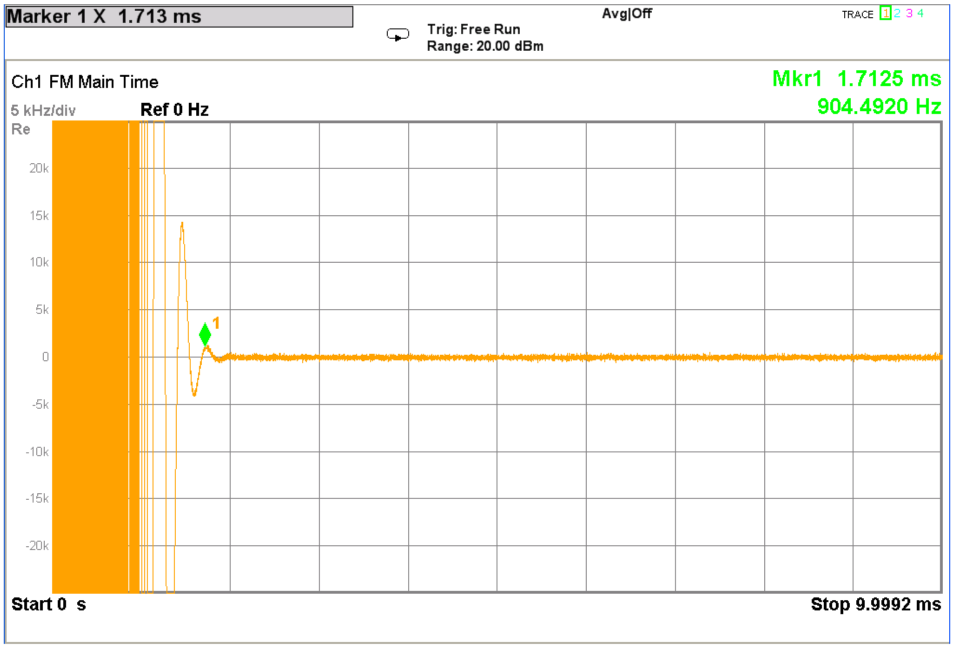


Figure 43 PLL1 Lock Time

Note: The graph shown in Figure 43 shows the lock times for a change in f_{VCO} from 875MHz (IDIV = 182, FDIV = $4AAAAA) to 965MHz (IDIV = 201, FDIV = $0AAAAA) for f_{COMP} = 4.8MHz (RDIV = $4) and 250µA charge pump current. Without fast lock: Loop Filter (reference design as shown in Figure 21)
C1 = 6n8F / C2 = 27nF / R2 = 1k5ohms / R3 = 470ohms / C3 = 470pF

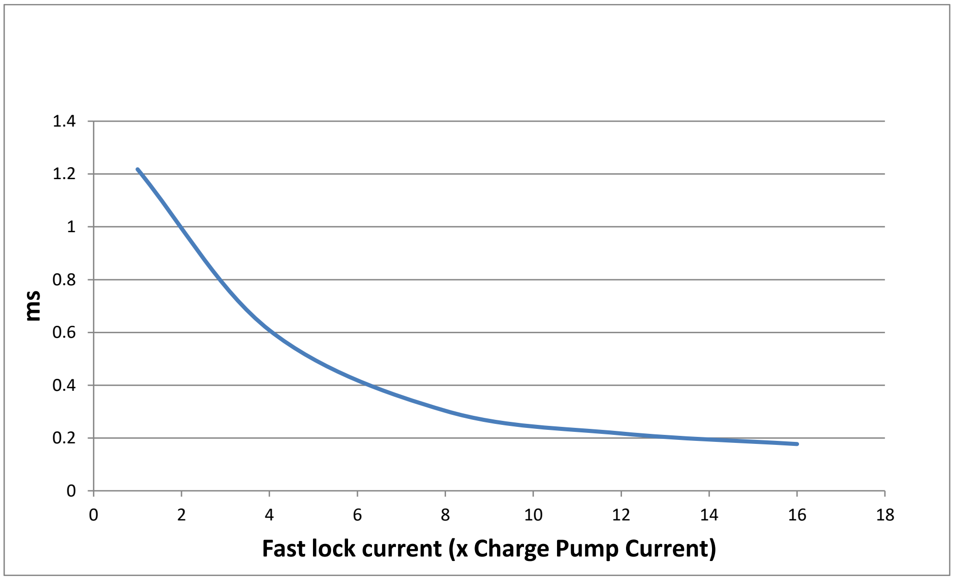


Figure 44 PLL1 Indicated Lock Time vs. Fast Lock Current (965 to 875.525 MHz)

Note: Fast lock timer, coarse = 4, fine = 80, giving 1.067ms

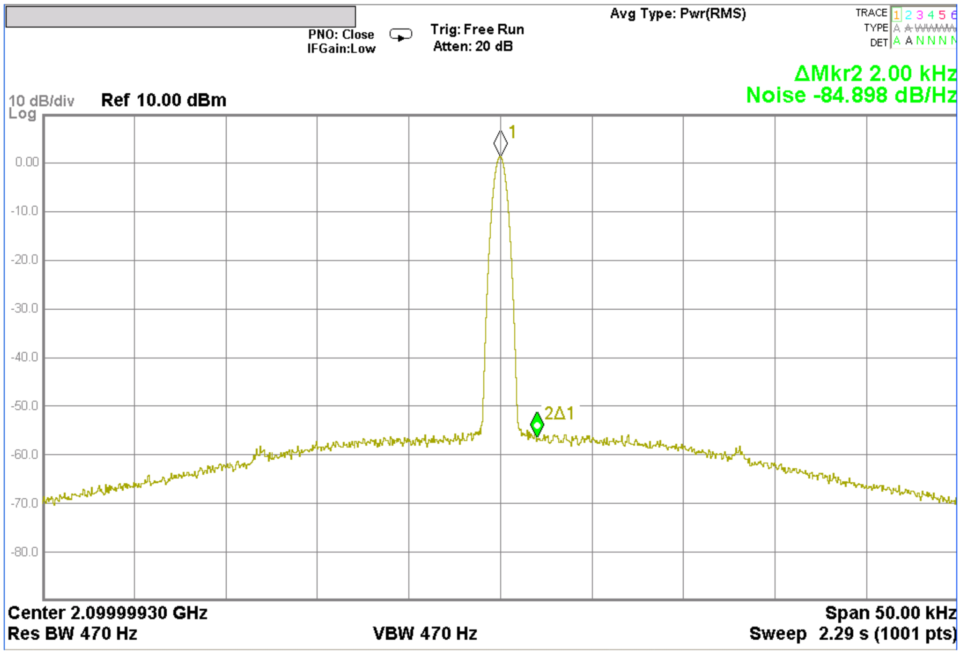
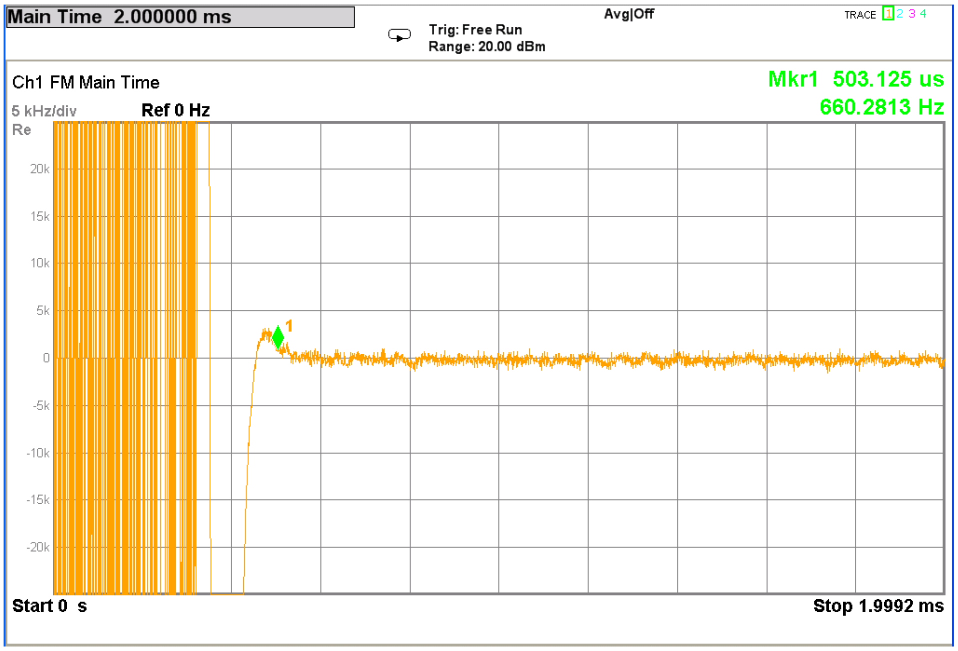


Figure 45 PLL2 Output Spectra

Note: The above graph shows the output spectra for f_{COMP} = 4.8MHz (RDIV = $4) and 250µA charge pump current. Without fast lock: Loop Filter (reference design as shown in Figure 21)
C1 = 1n5F / C2 = 9n4F (4n7F//4n7F) / R2 = 4k7ohms / R3 = 1k2ohms / C3 = 180pF



Main Time 2.000000 ms

Trig: Free Run
Range: 20.00 dBm

Avg|Off

Mkr1 503.125 us
660.2813 Hz

Figure 46 PLL2 Lock Time

Note: The above graph shows the lock times for a change in f_{VCO} from 2044MHz (IDIV = 426, FDIV = $D55555) to 2125MHz (IDIV = 443, FDIV = $B55555) for f_{COMP} = 4.8MHz (RDIV = $4) and 250µA charge pump current.
Without fast lock: Loop Filter (reference design as shown in Figure 21)
C1 = 1n5F / C2 = 9n4F (4n7F//4n7F) / R2 = 4k7ohms / R3 = 1k2ohms / C3 = 180pF

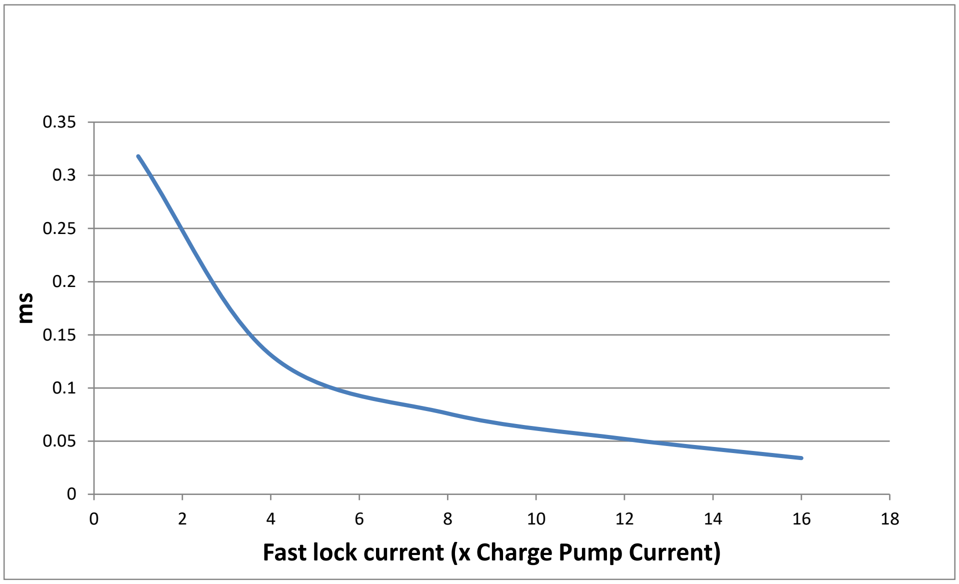


Figure 47 PLL2 Indicated Lock Time vs. Fast Lock Current (2125 to 2043 MHz)

Note: Fast lock timer, coarse = 6, fine = 1, giving 213µs

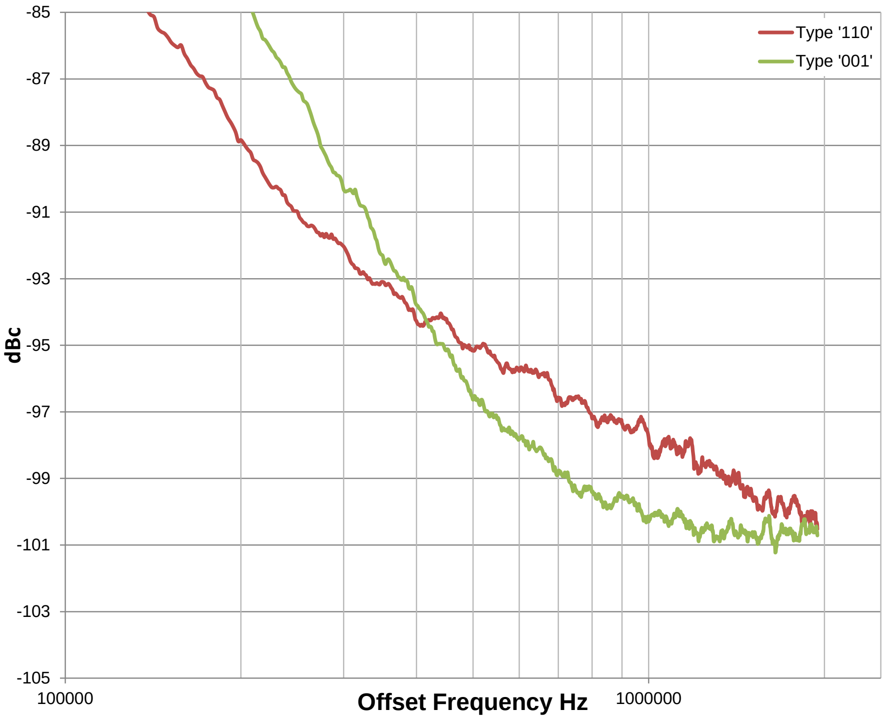


Figure 48 Relative Performance of PLL Options

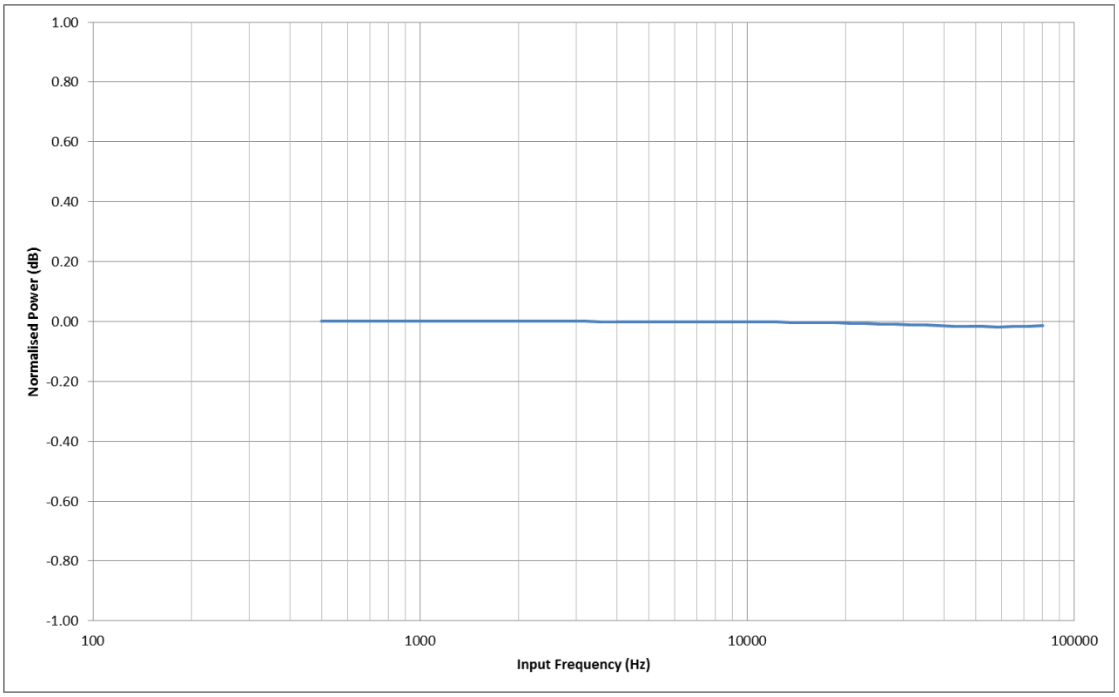


Figure 49 ADC Modulator Frequency Response (Σ-Δ clock=2.4MHz)

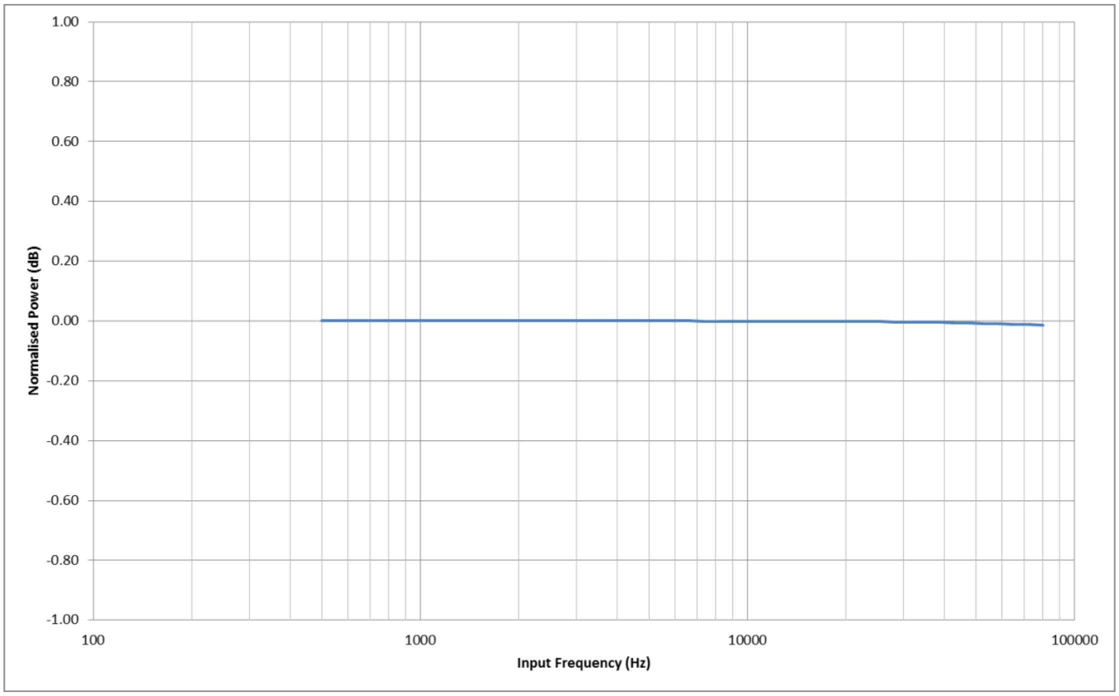


Figure 50 ADC Modulator Frequency Response (Σ-Δ clock=4.8MHz)

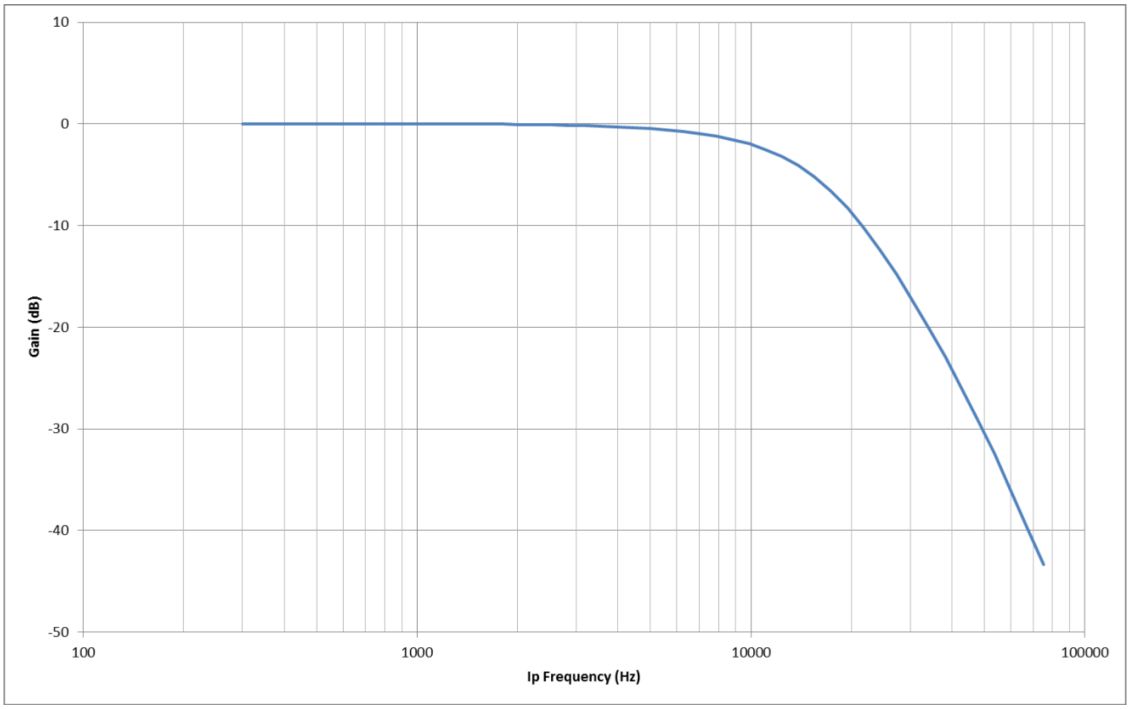


Figure 51 DAC Frequency Response (Σ-Δ clock=2.4MHz, SC Filter Bandwidth=Low)

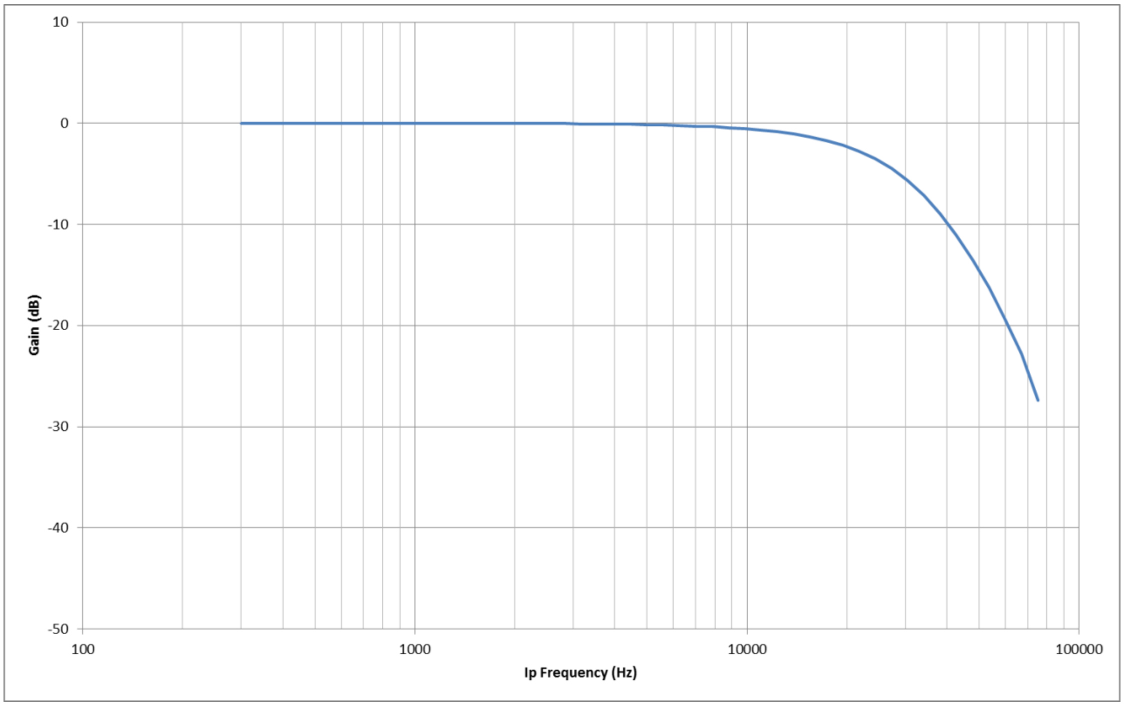


Figure 52 DAC Frequency Response (Σ-Δ clock=2.4MHz, SC Filter Bandwidth=High)

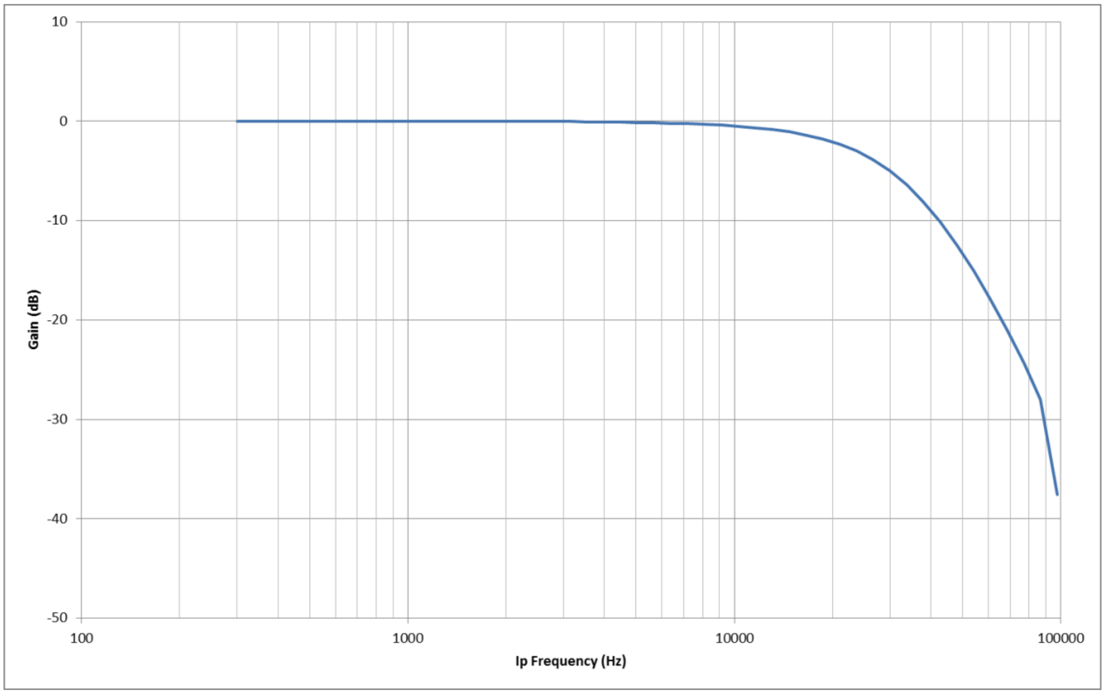


Figure 53 DAC Frequency Response (Σ-Δ clock=4.8MHz, SC Filter Bandwidth=Low)

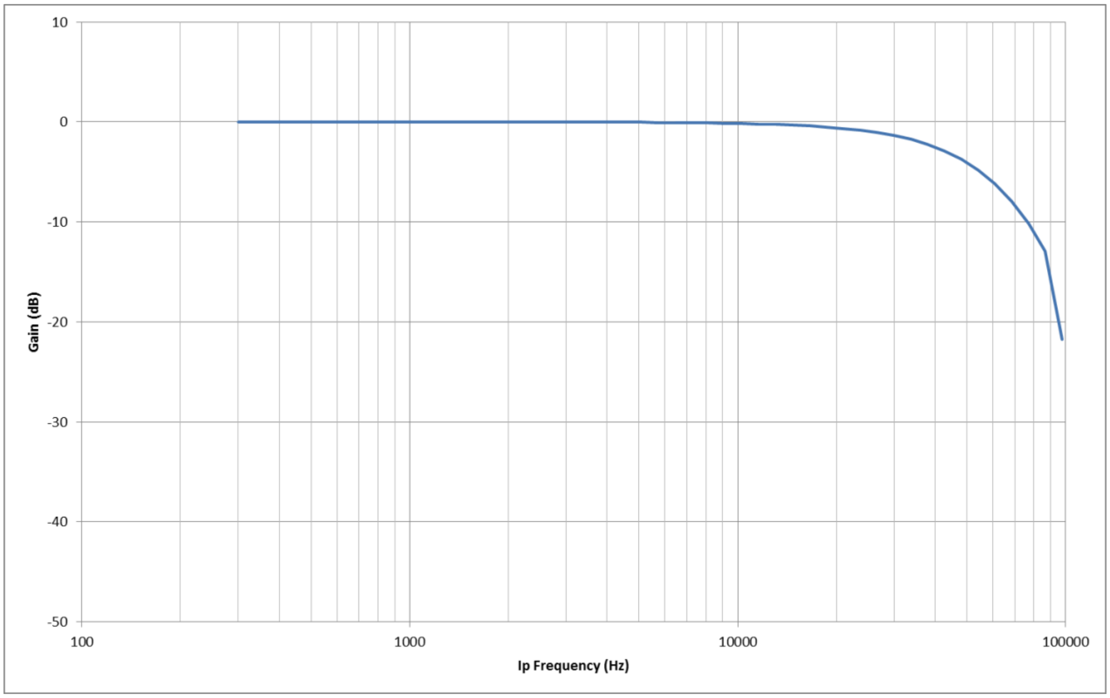


Figure 54 DAC Frequency Response (Σ-Δ clock=4.8MHz, SC Filter Bandwidth=High)

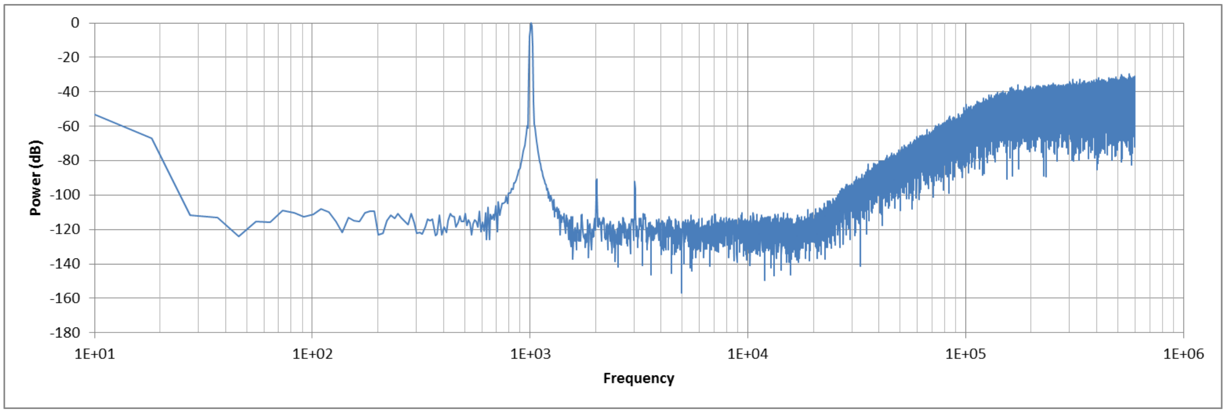


Figure 55 ADC Modulator Spectrum (Clock Rate = 2.4MHz)

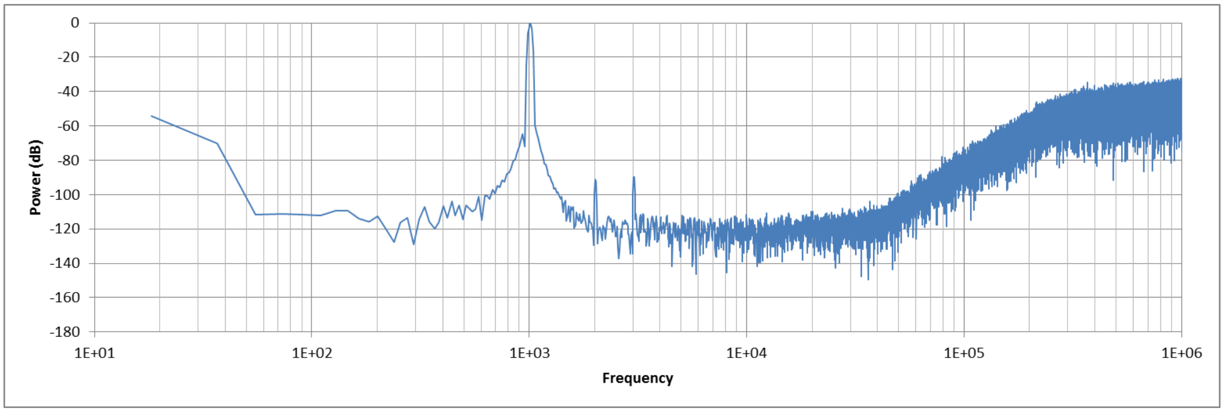
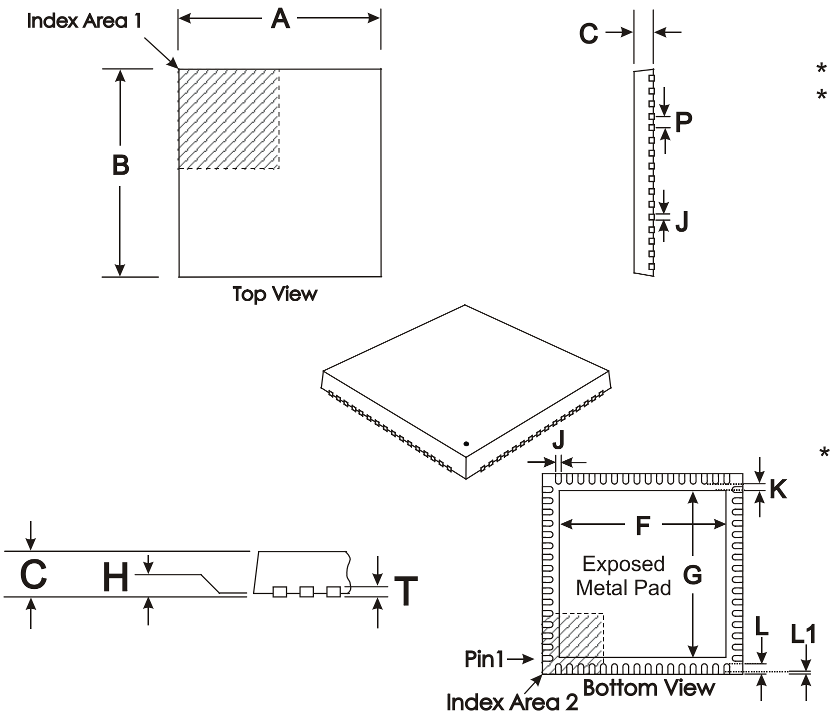


Figure 56 ADC Modulator Spectrum (Clock Rate = 4.8MHz)

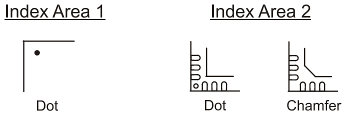
16 Packaging



DIM.	MIN.	TYP.	MAX.
* A		9.00 BSC	
* B		9.00 BSC	
C	0.80	0.90	1.00
F	7.00		7.80
G	7.00		7.80
H	0.00		0.05
J	0.18	0.25	0.30
K	0.20		
L	0.30	0.40	0.50
L1	0		0.15
P		0.50	
T		0.20	

NOTE :
* A & B are reference data and do not include mold deflash or protrusions.

All dimensions in mm
Angles are in degrees



Index Area 1 is located directly above Index Area 2

Depending on the method of lead termination at the edge of the package, pull back (L1) may be present.
L minus L1 to be equal to, or greater than 0.3mm
The underside of the package has an exposed metal pad which should ideally be soldered to the pcb to enhance the thermal conductivity and mechanical strength of the package fixing. Where advised, an electrical connection to this metal pad may also be required

Figure 57 64-lead VQFN Mechanical Outline: *Order as part no. CMX983Q1*

Handling precautions: This product includes input protection, however, precautions should be taken to prevent device damage from electro-static discharge. CML does not assume any responsibility for the use of any circuitry described. No IPR or circuit patent licences are implied. CML reserves the right at any time without notice to change the said circuitry and this product specification. CML has a policy of testing every product shipped using calibrated test equipment to ensure compliance with this product specification. Specific testing of all circuit parameters is not necessarily performed.

United Kingdom p: +44 (0) 1621 875500 e: sales@cmlmicro.com
techsupport@cmlmicro.com
Singapore p: +65 62888129 e: sg.sales@cmlmicro.com
sg.techsupport@cmlmicro.com
United States p: +1 336 744 5050 e: us.sales@cmlmicro.com
us.techsupport@cmlmicro.com

www.cmlmicro.com