MX-COM, INC.
COMMUNICATION ICs

# CMX980A

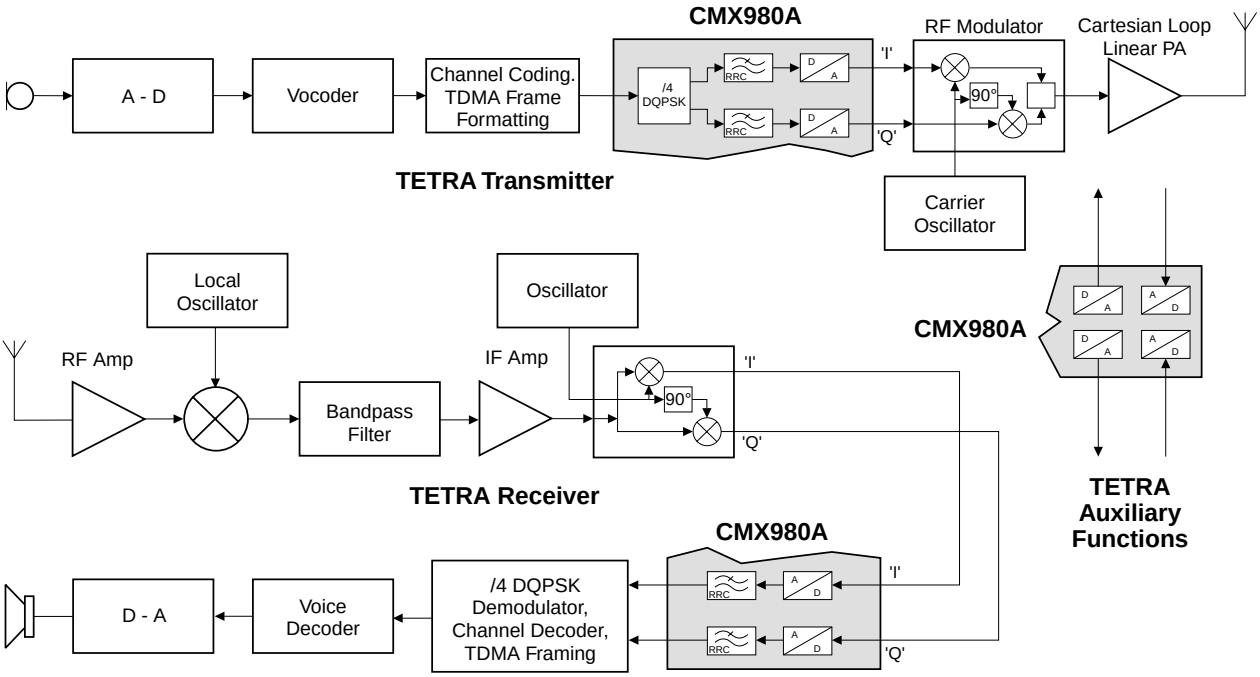
## DATA BULLETIN

## Digital Radio Baseband Processor

**Advance Information**

## Features

- **RRC Filters for both Tx and Rx**
- **π/4 DQPSK Modulation**
- **2 x 14-Bit Resolution Sigma Delta D-A**
- **2 x 16-Bit Resolution Sigma Delta A-D**
- **4 x 10-Bit D-A and 4 Input 10-Bit A-D**
- **Transmit Output Power Control**
- **Low Power 3.0 - 5.5Volt Operation**
- **Effective Power down Modes**




***Example: CMX980A in a TETRA System Application***

This device is intended to act as an interface between the analog and digital sections of a Digital Radio System, and performs many critical and DSP-intensive functions. The chip is designed with the necessary capability to meet the requirements for use in both mobile and base station applications in Terrestrial Trunked Radio (TETRA) systems, but the architecture is sufficiently flexible to allow use in other systems.

The transmit path comprises all the circuitry required to convert digital data into suitably filtered analog I and Q signals for subsequent up-conversion and transmission. This includes digital control of the output amplitudes, digital control of the output offsets and fully programmable digital filters: default coefficients provide the RRC response required for TETRA.

The receive section accepts differential analog I and Q signals at baseband and converts these into a suitably filtered digital form for further processing and data extraction. A facility is provided for digital offset correction and the digital filters are fully programmable with default coefficients providing the RRC response required for TETRA.

Auxiliary DAC and ADC functions are included for the control and measurement of the RF section of the radio system. This may include AFC, AGC, RSSI, or may be used as part of the control system for a Cartesian Loop.

The CMX980A requires a 3.0V to 5.0V supply and is available in the following packages: 44-pin PLCC (CMX980AL6) and 44-pin QFP (CMX980AL7).

 **www.mxcom.com tel: 800 638 5577 336 744 5050 fax: 336 744 5054** **Doc. 20480201.001**
4800 Bethania Station Road, Winston-Salem, NC 27105-1201, USA 

# CONTENTS

**Section** **Page**


**1 Block Diagram ..... 4**

**2 Signal List ..... 5**

**3 External Components ..... 7**

3.1 Rx Inputs ..... 7

3.2 Tx Outputs ..... 7

**4 General Description ..... 8**

4.1 Connection and Decoupling of Power Supplies ..... 8

4.2 Programmable FIR Filter Architecture ..... 9

4.3 Tx Data Path ..... 9

4.3.1 Modulator ..... 10

4.3.2 Filters ..... 10

4.3.3 Gain Multiplier ..... 10

4.3.4 Offset Adjust ..... 10

4.3.5 Sigma-Delta D-A Converters and Reconstruction Filters ..... 10

4.3.6 Phase Pre-distortion ..... 10

4.3.7 Ramping Output Amplitude ..... 10

4.3.8 Symbol Clock Phase Adjustment ..... 11

4.3.9 1.5.3.9 Direct Write to Tx 79-tap Filter Input ..... 11

4.3.10 Test Access to DAC Input ..... 11

4.4 Rx Data Path ..... 11

4.4.1 Anti-Alias Filtering and Sigma-Delta A-D Converters ..... 11

4.4.2 Rx FIR Filters ..... 11

4.4.3 Offset Registers ..... 12

4.4.4 I and Q Channel Gain ..... 12

4.5 Auxiliary Circuits ..... 12

4.5.1 10-Bit DACs ..... 12

4.5.2 10-Bit ADC ..... 12

4.5.3 Power Ramping and Control ..... 12

4.6 IRQ Function ..... 12

4.7 Serial Interface ..... 13

4.7.1 Command Interface ..... 14

4.7.2 Command Read Interface ..... 14

4.7.3 Rx Data Interface ..... 14

4.7.4 Transmission of Data ..... 15

4.7.5 Command Control Serial Word ..... 16

4.7.6 Coefficient Memory ..... 17

4.7.7 Auto Power Save Mode ..... 18

4.8 Register Description ..... 18
4.8.1 Key to Register Map ..... 18
4.8.2 Register Reset State ..... 18
4.8.3 Register and Access Point Summary ..... 19

**5 Application Notes ..... 64**
5.1 Interrupt Handling ..... 64
5.1.1 Tx FIFO status interrupts ..... 64
5.1.2 Tx/Rx FIR filter tap overflow and Gain, Phase and Offset overflow interrupts ..... 64
5.1.3 Rx ADC I and Q channel overflow - due to excessive input amplitude interrupts ..... 64
5.2 Configuration ..... 64
5.3 Reset ..... 64
5.4 Developing and Optimizing FIR Filter Coefficients ..... 64
5.4.1 Tx Path Details ..... 65
5.4.2 Rx Path Details ..... 66
5.4.3 General Procedure for Reconfiguring the CMX980A FIR Filters ..... 67
5.5 Receiver Re-Synchronization ..... 68
5.6 Guidelines for use of Power Save Modes ..... 69
5.6.1 Auxiliary Section ..... 69
5.6.2 Tx Section ..... 69
5.6.3 Rx Section ..... 69
5.6.4 Tx and Rx Bias Section ..... 69
5.6.5 Serial Interface Section ..... 69

**6 Performance Specification ..... 70**
6.1 Electrical Performance ..... 70
6.1.1 Absolute Maximum Ratings ..... 70
6.1.2 Operating Limits ..... 71
6.1.3 Operating Characteristics ..... 72
6.2 Packaging ..... 82

MX-COM, Inc. reserves the right to change specifications at any time and without notice.

# 1 Block Diagram

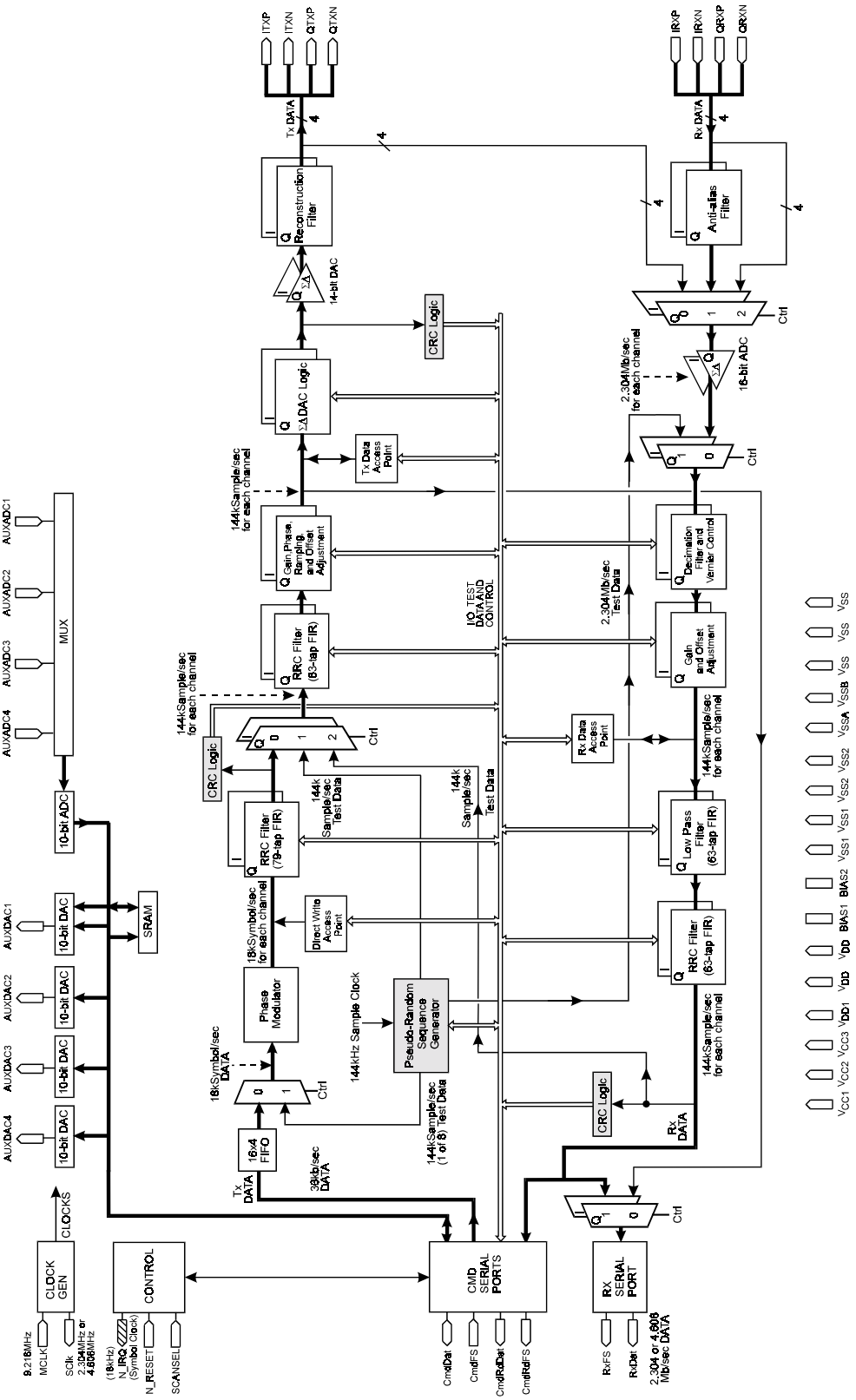



**Figure 1: Block Diagram**

# 2 Signal List

| Packages | | Signal | | Description |
|---|---|---|---|---|
| 44-pin PLCC (L6) | 44-pin QFP (L7) | | | |
| Pin No. | Pin No. | Name | Type | |
| 15 | 9 | MCLK | input | Master clock input (typically 9.216MHz) |
| 16 | 10 | SClk | output | Serial interface clock |
| 17 | 11 | CmdDat | bi-directional | Command serial interface Data |
| 18 | 12 | CmdFS | input | Command serial interface Frame |
| 19 | 13 | CmdRdDat | output | Command serial interface Read Data |
| 20 | 14 | CmdRdFS | output | Command serial interface Read Frame |
| 11 | 5 | RxDat | output | Receive serial interface Data |
| 12 | 6 | RxFS | output | Receive serial interface Strobe |
| 23 | 17 | N_IRQ | output | Interrupt request |
| 14 | 8 | N_RESET | input | Chip Reset |
| 24 | 18 | ~ | input | For manufacturers use only. Connect this pin to $V_{SS.}$ |
| 25 | 19 | ITXP | output | Transmit "I" channel, positive output |
| 26 | 20 | ITXN | output | Transmit "I" channel, negative output |
| 30 | 24 | QTXP | output | Transmit "Q" channel, positive output |
| 29 | 23 | QTXN | output | Transmit "Q" channel, negative output |
| 42 | 36 | IRXP | input | Receive "I" channel, positive input |
| 41 | 35 | IRXN | input | Receive "I" channel, negative input |
| 38 | 32 | QRXP | input | Receive "Q" channel, positive input |
| 37 | 31 | QRXN | input | Receive "Q" channel, negative input |
| 43 | 37 | AUXADC1 | input | Auxiliary ADC channel 1 |
| 44 | 38 | AUXADC2 | input | Auxiliary ADC channel 2 |
| 1 | 39 | AUXADC3 | input | Auxiliary ADC channel 3 |
| 2 | 40 | AUXADC4 | input | Auxiliary ADC channel 4 |
| 10 | 4 | AUXDAC1 | output | Auxiliary DAC channel 1 |
| 9 | 3 | AUXDAC2 | output | Auxiliary DAC channel 2 |
| 8 | 2 | AUXDAC3 | output | Auxiliary DAC channel 3 |
| 7 | 1 | AUXDAC4 | output | Auxiliary DAC channel 4 |
| 36 | 30 | BIAS1 | bi-directional | Analog bias level. This pin should be decoupled to $V_{SSB.}$ |
| 35 | 29 | BIAS2 | bi-directional | DAC reference level. This pin should normally be connected to $V_{SSB}$. |
| 32 | 26 | $V_{CC1}$ | Power | I Channel analog positive supply rail. This pin should be decoupled to $V_{SS1.}$ |
| 33 | 27 | $V_{CC2}$ | Power | Q Channel analog positive supply rail. This pin should be decoupled to $V_{SS2.}$ |
| 34 | 28 | $V_{CC3}$ | Power | Analog Bias positive supply rail. Levels and voltages are dependent upon this supply. This pin should be decoupled to $V_{SSB.}$ |
| 6 | 44 | $V_{DD1}$ | Power | Auxiliary analog positive supply rail. This pin should be decoupled to $V_{SSA.}$ |

| Packages | | Signal | | Description |
|---|---|---|---|---|
| 44-pin PLCC (L6) | 44-pin QFP (L7) | | | |
| Pin No. | Pin No. | Name | Type | |
| 3, 21 | 41, 15 | $V_{DD}$ | Power | Digital positive supply rail. This pin should be decoupled to $V_{SS}$. |
| 27, 40 | 21, 34 | $V_{SS1}$ | Ground | I Channel analog negative supply rail. |
| 28, 39 | 22, 33 | $V_{SS2}$ | Ground | Q Channel analog negative supply rail. |
| 31 | 25 | $V_{SSB}$ | Ground | Analog Bias negative supply rail. |
| 5 | 43 | $V_{SSA}$ | Ground | Auxiliary analog negative supply rail. |
| 4, 13, 22 | 42, 7, 16 | $V_{SS}$ | Ground | Primary digital negative supply rail. |
| **Notes:** Carefully observe the position of pin 1 on each package type. | | | | |

**Table 1: Signal List**

# 3 External Components

## 3.1 Rx Inputs

When using the internal anti-alias filter, the following is recommended:

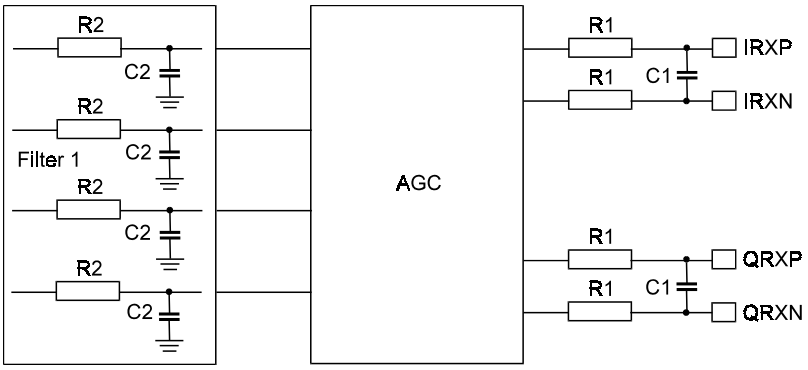



**Figure 2: Recommended External Components - Rx Input**

***Example Values: MCLK = 9.216MHz***

| | | |
|---|---|---|
| R1 | = | 220Ω |
| R2 | = | 1.2kΩ |
| C1 | = | 1.5nF (R1, C1 precise values are not critical, -3dB at 240kHz) |
| C2 | = | 3.9nF (R2 x C2 product, giving –3dB at 32kHz, should be preserved within 10%) |

**Table 2: Recommended External Components - Rx Inputs**

The RC stage formed by R2 and C2 combined with the internal anti-alias filter and Rx FIRs gives a good approximation to the desired filter characteristics and near-constant group delay over the passband. When not using the internal anti-alias filter, it is suggested that the user should follow the guidelines in Section 4.4.1. In both cases, there should be at least one filter pole close to the chip inputs.

## 3.2 Tx Outputs

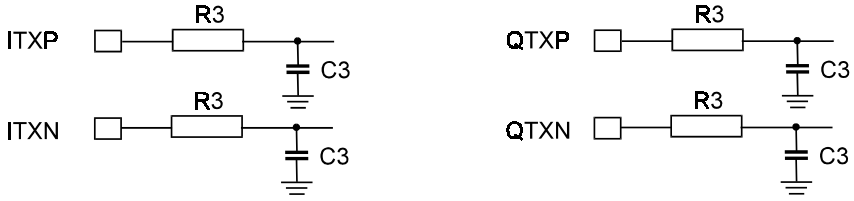



**Figure 3: Recommended External Components - Tx Outputs**

***Example Values: MCLK = 9.216MHz***

| | | |
|---|---|---|
| R3 | = | 6.2kΩ |
| C3 | = | 1nF (R3 x C3 product, giving –3dB at 25kHz, should be preserved within 10%) |

**Table 3: Recommended External Components - Tx Outputs**

The RC stage formed by R3 and C3 combined with the internal reconstruction filter and the Tx FIRs gives a good approximation to the desired filter characteristics and near-constant group delay over the passband. Decoupling capacitors should be employed as detailed in Section 4.4.1.

When the default master clock frequency is not used, the R2 x C2 and R3 x C3 products may be scaled with MCLK, but care should be taken to ensure that FIR filter coefficients are designed to compensate for any amplitude and phase distortion due to both on and off-chip filter components. This compensation is included in the default filter coefficients. See Section 5.4 for further details.

# 4 General Description

The device is designed to operate at a master clock frequency of 9.216MHz, but may be used over the full specified frequency range provided that guidelines in this document are followed. Many internal functions scale with the master clock frequency, which is referred to as “MCLK” where this is the case.

## 4.1 Connection and Decoupling of Power Supplies

Optimum performance from the CMX980A can only be obtained by the use of adequate decoupling and the separation of analog and digital signals, including the use of separate ground planes. Printed circuit board layout should follow the recommendations shown in Figure 4.

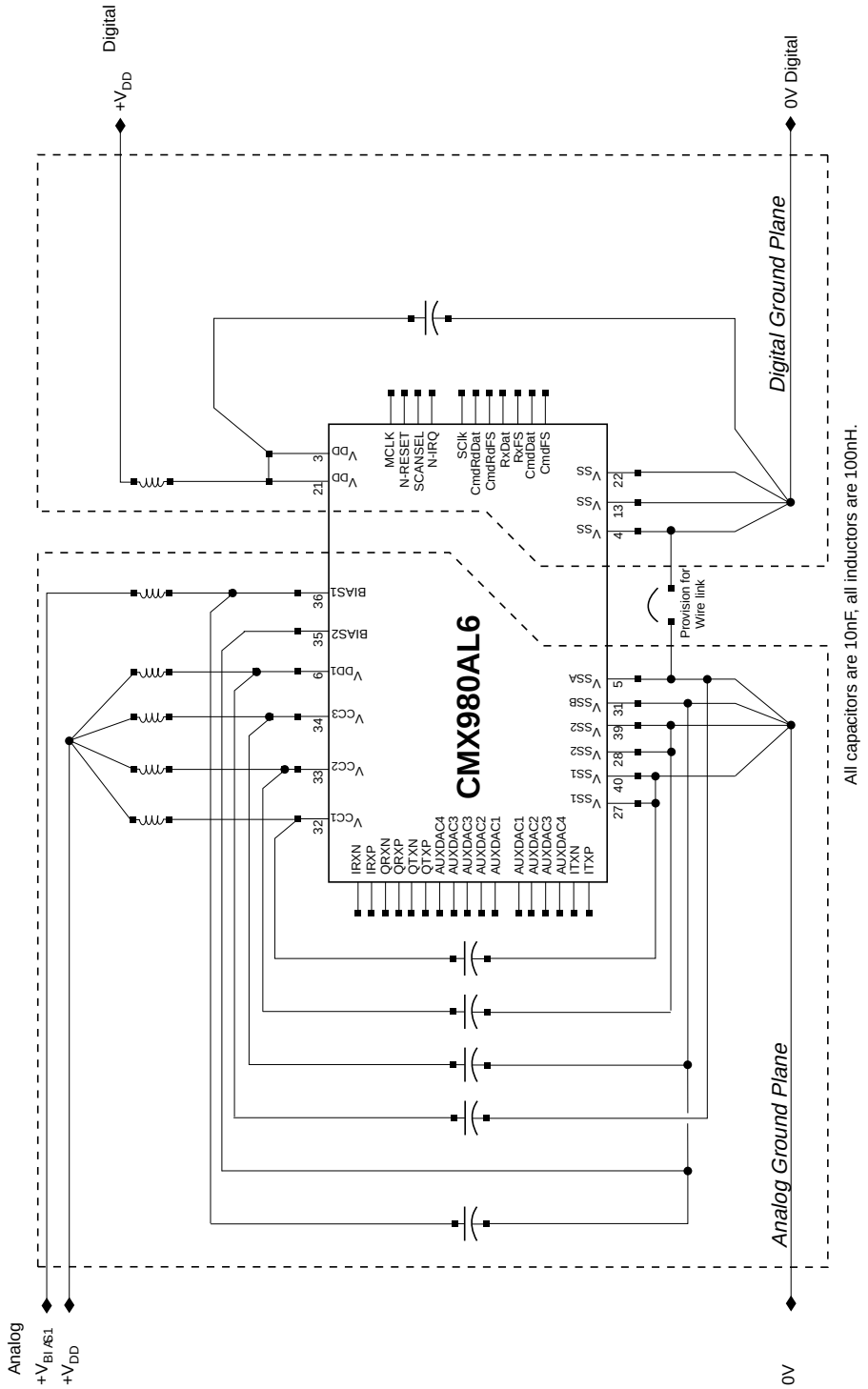



**Figure 4: Recommended Decoupling Components**

### 4.2 Programmable FIR Filter Architecture

Within both, the transmit and receive data paths a common FIR filter architecture for the implementation of the filtering requirements is employed. The filters use a small local static RAM for efficient data and coefficient storage during filter operations, together with a dedicated hardware multiplier and accumulator for each filter.

On reset, the coefficient RAMs are loaded with default values that provide the required response to meet the needs of a TETRA baseband system. In the default modes the dynamic range of arithmetic units are sufficient for all normal input data levels without causing overflows. Each filter has an odd number of default coefficients, which are symmetrical, giving a linear phase response. These coefficients may be overwritten to adapt to other systems or compensate for deficiencies outside the device. However the user is then responsible for ensuring that user supplied values do not cause arithmetic overflows to occur within an accumulation cycle. Overflow logic within each filter can detect such events and cause interrupts to be generated under user control.

The data RAMs store the filter input data samples and operate upon these values to provide the general FIR transfer function:

$$y(k) = \sum_{n=1}^{n=FL} A_n . D_{(n-k)}$$

where: $FL$ is the filter tap length

$A_n$ is the nth filter coefficient

$D_{(n-k)}$ is the data sample supplied to the filter n-k samples previously

When a filter is de-activated, coefficient RAMs retain their state, while the data RAMs are reset to zero. This ensures that the filters start from a quiescent state and prevents filter “memory” from a previous data frame. Asserting the N_RESET pin will cause all programmable filter coefficients to return to default values. Alternatively, the Tx and Rx path filter coefficients may be reset independently from each other by use of a control bit. The data RAMs, unlike the coefficient RAMs, are not directly accessible to the user.

Read or write operations to the coefficient RAMs can be performed by accessing the base address, which points to the MSB register of the first coefficient A1. This should be followed by a LSB register access that will auto-index the internal RAM address pointer to A2. Successive operations will continue to auto-index the RAM address pointer until A(FL) is reached. A further access after this point leads to a reserved location A0 that should not be altered. Continuing operations beyond this point returns the pointer to A1 again.

All filters, except the 79-tap Tx, allow access to the complete coefficient set, although the default values are symmetrical about (FL+1)/2. This will enable users to realize non-symmetrical filter functions should this be required.

All filters can be effectively by-passed by setting any single coefficient to normalized unity ( $2^{11}$-1 in the Tx and $2^{15}$-1 in Rx) and all others to zero. The chosen position of the “unity” coefficient will vary the internal group delay, thus this feature should be used with care. For example, the Tx ramping feature has a built in delay, which defaults to the expected group delay for the Tx filter path. Ramp delay may be bypassed, if required, by setting the appropriate bit in the **BISTControl** Register. The default group delay can be retained by choosing the central coefficient as “unity”.

The 79-tap filter has only one half of the coefficient RAM available, so can only implement symmetrical (linear phase) filter responses. Thus, when accessing this filter only locations A1-A40 are valid. In addition, to bypass this filter, the central coefficient (A40) should be chosen as “unity”, since this is the only unique coefficient.

### 4.3 Tx Data Path

The features described below give a high degree of flexibility for the user to compensate in the baseband processing for non-ideal performance in the IF, RF and RF linear amplifier sections.

### 4.3.1 Modulator

This takes the 2-bit symbols, performs a Gray Code conversion and uses a recursive adder to generate a 3-bit code representing the 8 possible phase states. A look up table provides the digitally encoded I and Q values for each phase state. The modulator function can be by-passed if required; in this case, the 3-bit code representing the 8 possible phase states which are passed to the look up table is provided directly via the serial interface.

### 4.3.2 Filters

Digital filtering is applied to the data from the modulator by two programmable FIR filters. The first has 79-taps and provides stop band rejection and sampling correction. The second has 63-taps and provides the primary Root Raised Cosine (RRC) shaping with Roll-off factor ($\alpha$) of 0.35, together with correction for droop in the switched capacitor reconstruction filter. These FIR filters operate at eight times the incoming symbol rate and are configured as two filters in cascade for each I and Q channel.

### 4.3.3 Gain Multiplier

This feature allows user control of the signal amplitudes in the I and Q channels independently. The multiplier provides a resolution of 11 bits; i.e. the gain is adjustable in steps of 1/2048 of maximum level. Additional logic allows a mode of operation that will enable ramping up to the set signal level, stay at this value while instructed by the user, then ramp back down to zero. The maximum value for each channel, the ramping up rate and the ramping down rate are all programmable via the serial interface.

### 4.3.4 Offset Adjust

Offset registers allow any offsets introduced in the analog sections of the transmit path to be corrected digitally via the serial interface. The offset adjust is independently applied to each of the I and Q channels. The adjustment range is plus and minus full scale in each section with a resolution of 1 LSB. Thus, care must be exercised by the user to avoid excessive offsets being applied to the Sigma-Delta DAC.

### 4.3.5 Sigma-Delta D-A Converters and Reconstruction Filters

The converters are designed to have low distortion and >80dB dynamic range. These $2^{nd}$ order converters operate at a frequency of 128 x symbol rate so as to over-sample the data at their inputs a further 16 times. The reconstruction filters are $3^{rd}$ order, switched capacitor, low pass filters designed to work in conjunction with an external RC.

### 4.3.6 Phase Pre-distortion

A further feature allows the user to compensate for a non-orthogonal carrier phase in the external quadrature modulator by adding a programmable fraction of up to 1/8 of the filtered I and Q channel signals to each other immediately prior to the DAC input.

### 4.3.7 Ramping Output Amplitude

A facility is provided to allow ramping of the outputs in two modes. When enabled by the user, the signal from the gain multiplier stage is multiplied by an envelope value. The value in this register, increments or decrements at a rate programmed by the user, which is held in the TxRampUpInc and TxRampDnDec Registers respectively.

The ramping envelope can be selected by the user to be linear or non-linear. In non-linear mode, the envelope function is sigmoidal, minimizing spectrum spread whilst fast ramping is in operation. The RCR is a 11-bit register (not user accessible), representing a value from 0 to 1.0, which can be incremented by the value TxRampUpInc until the count of 2047 (1.0) is reached, or decremented by the value in TxRampDnDec until zero is reached.

In linear mode, this value (RCR) is used directly to provide the envelope amplitude, whilst in non-linear mode it is input to a look-up table of the sigmoidal function, which in turn provides the envelope amplitude. Ramping begins from zero when the user applies valid transmission data with the TxRampUp bit in the TxData Register set and continues in increments of TxRampUpInc until the set gain level (see Section 4.3.3) is reached. To begin the ramp down phase of a transmit burst the user writes post-amble data with the TxRampUp bit cleared then the RCR decrements by an amount TxRampDnDec until the result is less than or equal to zero, whereupon the gain is set to zero. Internal flag registers are available to indicate to the user that ramp down is complete.

The TxRampUpInc and TxRampDnDec Registers are both 9-bit words input via the serial interface prior to the start of a transmission; this gives programmable ramping rates from 0.125 to 64 symbol-times.

### 4.3.8 Symbol Clock Phase Adjustment

In order to comply with the requirement to maintain the phase error between the Mobile Station (MS) and Base Station (BS) symbol clock to less than ±1/4 symbol time, a mechanism to allow phase adjustment of the CMX980A symbol clock is provided.

This phase adjustment is achieved by writing a command to the **SymClkPhase** Register**,** which allows adjustment in steps of ±1/4 or ±1/8 symbol times. It is intended that the user determine the symbol clock phase of the BS after clock recovery has been performed on the received data. Then, allowing for the fixed Tx path delay, the CMX980A phase can be advanced or retarded until it is within the specified error limit. The internal symbol clock phase can be accessed by allowing the symbol clock reference signal to appear on the N_IRQ pin, or alternatively using the I/Q identification mode (see Section 4.7.3) which places the symbol clock in the Rx I channel LSB. Thus via hardware or software means the internal Tx symbol clock reference time can be determined and the phase with respect to the BS adjusted.

### 4.3.9 Direct Write to Tx 79-tap Filter Input

A mechanism to allow direct write to the I and Q Tx 79-tap filter inputs at the symbol rate is provided for use in systems where a different modulation scheme is to be employed. See Section 4.7.4 for further details.

### 4.3.10 Test Access to DAC Input

A mechanism to allow read and write access to DAC input data is provided for use in testing or in other systems where the modulator and filter blocks are not required. By operating the serial port at the high serial clock rate and without a frame gap, it is possible to provide only half of the normal bit rate for two channels, thus data can be provided at MCLK/64 for a single channel or MCLK/128 for both channels. The user should provide the appropriate data at the required sample rate (MCLK/64 or MCLK/128) via the serial interface, which will be transferred to the DAC logic at the next internal sample clock after the data is written to the register. Write operations to the upper and lower byte register and I and Q channels must be synchronized in phase by the user to the sample clock strobe. This is to avoid splitting the I and Q channel or upper and lower bytes into different samples. The phase of the sample clock can be determined by allowing the Symbol Clock (which is in phase with the internal sample clock but 1/8 of the rate) to appear on the N_IRQ pin.

Note that data input at this point will have to be pre-filtered to compensate for the reconstruction filter droop (approximately 2dBs at MCLK/1024), which is normally compensated by the internal FIR default coefficients. In addition, data input at a MCLK/128 sample rate will have a sinx/x alias around MCLK/128, which will be reduced to about 65dBs below the wanted signal by the reconstruction filter. There is some scope to improve this by enhancing the recommended single pole filter stage on the Tx output, but any adverse change in the in-band gain and group delay performance will have to be compensated prior to loading the data into the IC.

## 4.4 Rx Data Path

### 4.4.1 Anti-Alias Filtering and Sigma-Delta A-D Converters

The sampling frequency of the Sigma-Delta A-D is 128x symbol rate. The high over-sampling rate relaxes the design requirements on the anti-alias filter. However, to achieve optimum performance the anti-alias filter must reject the sampling frequency to about -110dB, of which at least 30dB must be provided externally. Additionally, in order to ease the complexity of the subsequent digital filters, there is a further requirement that the anti-alias filter suppress 8x symbol rate to about -15dB. The on-chip anti-alias filter is designed to achieve this when used in conjunction with some external filtering. If required, the on-chip anti-alias filter can be by-passed and powered down, although external anti-aliasing must then be provided. The fourth-order Sigma-Delta A-D converters are designed to have low distortion and >96dB dynamic range. The baseband I and Q channels must be provided as differential signals; this minimizes in-band pick up both on and off the chip.

Both I and Q Sigma-Delta converters produce a single bit output sampled at MCLK/4. This data is passed to a non-programmable decimation FIR filter, which is sampled at MCLK/4 and gives sufficient rejection at 8x symbol rate (MCLK/64) to permit decimation to that frequency (note that around -30dB is provided by the primary anti-alias filters).

### 4.4.2 Rx FIR Filters

Digital filtering is applied to the data from the Sigma-Delta A-D converter decimation filters by two 63-tap FIR filters in cascade. The default coefficients are set to give a Root Raised Cosine response with roll-off factor ($\alpha$) of 0.35. The first filter is used to enhance stop-band rejection, while the second filter provides the primary shaping requirements for root raised cosine response.

### 4.4.3 Offset Registers

System generated offsets may be removed by control of the offset register via the serial interface.

### 4.4.4 I and Q Channel Gain

Programmable gain modules are provided in both I and Q channels. These blocks allow the user to adjust the dynamic range of the received data within the digital filters, thus optimizing the filter signal to noise performance for a range of levels at the Rx input pins. In the receive section the gain-multiplier sign bit is user accessible, therefore phase inversion in each channel is possible by programming negative numbers into the gain registers.

The two channels are independently programmable. This enables differential gain corrections to be made within the digital domain.

## 4.5 Auxiliary Circuits

### 4.5.1 10-Bit DACs

Four 10-bit DACs are provided to assist in a variety of control functions. The DACs are designed to provide an output as a proportion of the supply voltage, depending on the digital input. They are monotonic with an absolute accuracy of better than 1%. Control and Data for these come via the serial interface.

### 4.5.2 10-Bit ADC

A 10-bit ADC is provided to assist in a variety of measurement and control functions. The ADC is designed to produce a digital output proportional to the input voltage; full scale being the positive supply. It is monotonic with an absolute accuracy of about 1%. An input multiplexer allows the input to be selected from one of four sources. Control and digital data output is via the serial interface.

### 4.5.3 Power Ramping and Control

One of the DACs has an additional feature that enables a set of values to be sequenced out at a pre-selected frequency. This is aimed at enabling power ramping of a RF output with a suitable profile. The sequence may be reversed for power down. The sequence of values is stored in a dedicated RAM, which can be loaded via the serial interface.

## 4.6 IRQ Function

An interrupt request (IRQ) pin (labeled N_IRQ) is provided for asynchronous communication with an external processor. The N_IRQ pin will be asserted (taken low) when any of the error or user information flags are activated by an internal operation. Some examples of operations that may generate an interrupt are:

1. An attempt by the user to write to a full Tx data-input FIFO
2. An attempt is made by the Tx to read from the Tx data-input FIFO when it is empty.
3. An internal arithmetic overflow has occurred in an FIR filter.

The IRQ feature may also be used to establish the phasing of the received I and Q channel data from the RxDat serial port should synchronization be lost for any reason.

The cause of the IRQ can be obtained by reading the error flags register. All possible causes of an IRQ are masked on reset. Mask status can be altered by writing to the IRQ mask register.

Note that default coefficients and settings have been optimized to maximize performance and should not cause arithmetic overflows. However, use of non-default coefficients, large offset corrections or large Tx phase adjustments may cause problems, which can be corrected by scaling down coefficients or via the gain multiplier feature.

Additionally, the internal symbol-clock signal may be brought out to this pin. This is intended for a number of uses, primarily in the following areas:

1. In multi-chip systems where symbol phase synchronization between devices is necessary.
2. To assist in timing the write operations to the 79-tap filter input in direct write mode.
3. To provide a reference signal during phase synchronization to the BS symbol clock.

## 4.7 Serial Interface

All digital data I/O and control functions for the CMX980A are via the serial interface. It is expected that the CMX980A will be used in conjunction with a DSP and/or other processor. The device has three serial interface ports, each port is based on the industrial standard three-wire serial interface. This interface allows communication with standard DSP ICs using a minimum of external components. The three serial interface ports are:

***Cmd*** Command port, generally this is an input port receiving commands and data from the host, but may also be configured as a bi-directional I/O interface.

***CmdRd*** Command read port, an output port to send command read data back to the host. Read data is only sent on this port in response to a read command.

***RxData*** Receive data port, an output port to send receive data back to the host. Data is only present on this interface when the Rx Data path is active. This port may also be configured as the CmdRd port.

***Functions performed by the serial interface include:***

- Power up or down and optional bypassing of selected blocks
- Setting digital filter coefficients
- Loading ramp up and ramp down increments for Tx data operations
- Loading and transmitting data
- Loading offset correction, gain multiplier and phase adjustment registers
- Enabling/disabling of output via the Rx serial interface
- Vary sampling time for Rx data relative to the sample (MCLK/64) clock.
- Loading data into auxiliary DACs
- Initiating conversions using auxiliary ADCs and reading results
- Writing data to, and reading data from, the Waveform Generation SRAM
- Power Ramping step control

***The three interfaces consist of the following signal pins:***

| | | |
|---|---|---|
| SClk | Output | *Serial Clock* pin. This pin is common for all three interfaces. |
| CmdDat | In/Out | Command port *Data* pin. This pin is by default an input, but may be configured as an open drain bi-directional pin. |
| CmdFS | Input | Command port *Frame Sync* pin. This pin is used to mark the first bit in a serial frame. |
| CmdRdDat | Output | Command read port *Data* pin. This pin only has active data on it in response to a read command. |
| CmdRdFS | Output | Command read port *Frame Sync* pin. This pin is used to mark the first bit in a serial frame. |
| RxDat | Output | Receive data port *Data* pin. This pin is only active when the Rx Data path is active. |
| RxFS | Output | Receive data port *Frame Sync* pin. This pin is used to mark the first bit in a serial frame. |
| **Note:** All *Frame Sync* strobe signals are actually coincident with the last bit of a dataframe. See Figure 5 and Figure 6 for further details. | | |

### 4.7.1 Command Interface

A serial command word consists of a 16-bit frame. Each frame is marked by an active *Frame Sync* event that precedes the MSB bit. A command word can be either a control word or a transmit data word.

| MSB | | | | LSB |
|---|---|---|---|---|
| W / $\overline{R}$ | Address | | Data | |
| 15 | 14 | 8 | 7 | 0 |

**Command Control Serial Word**

| MSB | | | | | | LSB |
|---|---|---|---|---|---|---|
| 1 | Tx Data Address | | U/D | 4/1 | Tx Data | |
| 15 | 14 | 10 | 9 | 8 | 7 | 0 |

**Command Transmit Data Serial Word**

### 4.7.2 Command Read Interface

Command read data is either output on one of the serial read ports, or driven out in the last 8 bits (data field) on the *Cmd* port. When command read data is output on a serial read port, the read address is put in the most significant half of the word and the read data in the least significant half.

| MSB | | | LSB |
|---|---|---|---|
| 0 | Read Address | Data | |
| 15 | 14 8 | 7 | 0 |

**Command Read Serial Word**

### 4.7.3 Rx Data Interface

The Rx Data interface is used only for output of the I and Q received data, unless it is operating in the mode where *CmdRd* data is directed to it. When data reception is enabled, I and Q received data will be output at either 8 x or 4 x the symbol rate, under control of command register **RxSetup1**. (see Section 4.8). This is achieved by reducing the serial interface clock rate from MCLK/2 to MCLK/4 and discarding alternate data samples under control of command registers **ConfigCtrl1** and **RxSetup1**. 16-bit I and Q data words are output at the Rx Data interface, I data and MSB first (by default), on the rising edge of SClk. To facilitate channel identification of the serial data, should initial synchronization be lost, and synchronization of the internal Tx symbol clock with received data, the CMX980A has an I/Q identification mode, which is controlled by setting the *RxIdentMode* bit in the **RxSetup1** Register.

### 4.7.4 Transmission of Data

The address of the Tx FIFO is given consecutive locations ($0x04-$0x07), which allows the address bits A1 and A0 (bits 11 and 10) of the Command Transmit Data Serial Word to be utilized as transmit control functions. Data to be transmitted can be in either one or four 2-bit symbol blocks, which are subsequently modulated into the DQPSK constellation, or in 3-bit words, which map directly into constellation points according to the table shown below:

| 3 bit code | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| **I** | 1 | 0.7071 | 0 | -0.7071 | -1 | -0.7071 | 0 | 0.7071 |
| **Q** | 0 | 0.7071 | 1 | 0.7071 | 0 | -0.7071 | -1 | -0.7071 |

**Table 4: Constellation Map**

The eight points in the DQPSK constellation each have a magnitude of 1 and are spaced at 45° intervals around the unit circle. The default operating mode modulates two bit symbols into the TETRA constellation by representing each symbol as a phase change, according to the following mapping, where the left hand bit is considered as the first bit of the symbol and corresponds to bit 0, 2, 4 or 6 of the TxData word (see the description in PAGE 0 ADDRESSED REGISTERS).

| Symbol | | Phase Change |
|---|---|---|
| 1 | 1 | -135 |
| 0 | 1 | +135 |
| 0 | 0 | +45 |
| 1 | 0 | -45 |

The user initiates a transmit frame by asserting the TxEn bit in the TxSetup Register. However, internal transmission of the data will wait until specific conditions have been met. Firstly, a valid data word must be written into the FIFO with the TxRampEn bit of the TxSetup Register asserted. Secondly, the internal symbol clock must be active. Therefore, there is a variable delay between asserting the TxEn bit and transmission starting. The user may poll the TxPathEn bit of the TxFIFOStatus Register to establish when transmission has started, and in this case, the active state of TxPathEn is High. In general, the user will wish to know when the transmit frame has completed. This is indicated by TxPathEn returning Low.

To relieve the user of polling overheads when waiting for Tx frame completion, an interrupt can be set up to occur on the transition of the TxPathEn bit from High to Low. In such circumstances, the interrupt activation state of the TxPathEn can be considered Low.

Two control bits are associated with each data transmission word. One controls the format of the word and the other initiates and terminates a transmission cycle. This close association enables precise control of the transmission frame. To relieve the user of the need to synchronize each TxData write with the internal transmit cycle, transmit data words are written into an internal 4-word-deep FIFO. Symbols or constellation points are then read as needed from this FIFO. It is necessary to make sure that there is always a word to be read, three data interlock mechanisms.

#### 4.7.4.1 Data Interlock Mechanisms

There are three possible transmission data interlock mechanisms. It is recommended that the user should always use one of these methods.

Software polling

Serial Clock when ready

Interrupt data demand

Software polling requires the user to first check that the FIFO is not full before writing each TxData word. This may be accomplished by inspecting the relevant FIFO status bits before writing one or more TxData words.

The Serial Clock when ready method is a hardware interlock mechanism (enabled by setting the TxHandshakeEn bit of ConfigCtrl1 Register active). The mechanism allows the user to write TxData words without doing any FIFO checks: the hardware handshake is implemented by stopping the serial port clock when the FIFO is full. To prevent a serial port lockout-condition, the handshake is only enabled once the transmission frame has been initiated and is automatically disabled at the end of a frame. This mechanism should be used with care, because stopping the clock will freeze all other serial port transfers (the serial port clock SClk is common to all three serial ports), including access to auxiliary data converters and receive data.

Interrupt data demand is used to request data when the FIFO has reached a defined level. An interrupt is generated when the data in the FIFO reaches the pre-defined level of “nearly” empty (1 word remaining) or when the FIFO is “nearly” full (1 location available). In each case, the user is responsible for managing the response latency in detecting and servicing the interrupt and for writing new data into the FIFO so that symbol ‘run-out’ does not occur.

#### 4.7.4.2 Direct write to 79-tap filter mode

The FIFO and DQPSK modulator may be bypassed thus allowing the user direct access to the Tx filter chain input. The 12-bit data words must be supplied to input holding registers at MCLK/512 samples/sec for both I and Q channels. To allow a single serial-operation write, the TxDirectWrite79tapI and TxDirectWrite79tapQ Registers are in the page 1 address map. By utilising the four least significant address bits to map to the most significant bits of the data, a 12-bit data word can be transferred in a single serial-write frame.

#### 4.7.4.3 Power Ramping and Frame Interlock

The RampUp bit in the TxData word is used to control both the power ramping function and the frame activation. To start a transmission frame, a transmission word is written with the RampUp bit active. All subsequent TxData words prior to frame termination must also have this bit active. The frame is terminated by writing transmit data words with the RampUp bit inactive. Subsequent TxData words must also have this bit inactive, until initiation of a new frame is required. While the power ramping is active, (up or down) the user must supply transmission symbols or valid constellation points. Once the ramp down operation has completed, all subsequent TxData writes with the RampUp bit inactive will be ignored.

### 4.7.5 Command Control Serial Word

A command word either directly accesses an internal register for a read or write operation, or addresses a memory access point to indirectly access a block of internal memory. For test purposes all registers that can be written may also be read. Not all registers may be written, as some are just status registers. Each register or memory access point is assigned a unique address: the whole (8-bit) address range is reserved for the CMX980A.

A page address technique is used to extend the available address space beyond the 128 locations allowed by 7-bit address fields. This gives four pages of 128 locations, of which the first two (page 0 and page 1) are used. The device configuration and control registers ConfigCtrl1 and ConfigCtrl2 are accessible across all pages, ConfigCtrl2 bits 6 and 7 forming the 2-bit page address.

#### 4.7.5.1 Indirect Memory Addressing

All internal memory access is via an access point. First, a command word access is used to reset the internal address pointer, then data port access operations post-increment this address pointer.

***Example:*** To program the fifth and sixth locations of the Auxiliary SRAM with $0x01AA, the commands would be:

| | | | | |
|---|---|---|---|---|
| $0x8000⇒Cmd | ; | set **ConfigCtrl1** all bits Low | ; | use default conditions |
| $0x8118⇒Cmd | ; | set **ConfigCtrl2** bits 7 and 6 Low | ; | required for Page 0 addressing. |
| | ; | set **ConfigCtrl2** bit 4 High | ; | post-increment addresses on a read operation |
| | ; | set **ConfigCtrl2** bit 3 High | ; | enable read/write access to the Auxiliary SRAM |
| $0x7300⇒Cmd | ; | read **SramData** LSB Register | ; | read fourth memory location (LSB). Post-increment pointer. |
| CmdRd⇒$0x73xx | ; | **SramData** LSB Register data returned | ; | discard this byte |
| $0xF002⇒Cmd | ; | write **SramData** LSB Register | ; | write $0x02 to fifth memory location (LSB) |
| $0xF16A⇒Cmd | ; | write **SramData** MSB Register | ; | write $0x6A to sixth memory location (MSB) |
| $0x7000⇒Cmd | ; | read **SramData** LSB Register | ; | read fifth memory location (LSB) |
| CmdRd⇒$0x7002 | ; | **SramData** LSB Register data returned | ; | check this byte is $0x02 |
| $0x7100⇒Cmd | ; | read **SramData** MSB Register | ; | read sixth memory location (MSB) |
| CmdRd⇒$0x716A | ; | **SramData** MSB Register data returned | ; | check this byte is $0x6A |
| $0x8110⇒Cmd | ; | set **ConfigCtrl2** bit 3 Low | ; | disable read/write access to the Auxiliary SRAM |

### 4.7.6 Coefficient Memory

The convention for naming filter coefficients is A1 to An, where n is given by the filter tap length, i.e. for a 63-tap filter, n = 63. Within the filter architecture, location A0 has a special purpose and must contain zero for correct operation of the computational algorithm. The internal architecture of the 63-tap filters allows access to all coefficients, but the default values are symmetrical about the central coefficient to provide linear phase response. The user is free to write non-symmetrical values, giving the possibility of non-linear phase correction for off-chip components in these filters. The Tx 79-tap filter differs by having coefficients A1 to A40 only, taking advantage of the filter symmetry to reduce its RAM size. Thus write or read operations beyond the A40 coefficient number will be reflected about the central coefficient e.g. the 47th read operation from the 79-tap filter would access coefficient location A33 (80-47).

To access the coefficient RAMs, the user asserts the CoeffRamIoEn bit in the ConfigCtrl2 Register, then performs the operation (read or write) to the MSB of the required FIR filter. The first access after the CoeffRamIoEn bit goes high is directed to location A1. Completing the coefficient access, by addressing the LSB, automatically moves the Coefficient Ram Pointer to A2. The process is repeated until the required number of locations has been accessed.

There is no practical reason to write or read beyond location A40 in the 79-tap filter, but in any case the user must avoid write operations at the (Filter Length + 1) location in any filter. As previously stated this location must be zero for the filters to operate correctly.

Note that filter coefficient read/write operations should be performed with the appropriate path (Tx or Rx) disabled, but the clock stop bits must NOT be set.

The global reset (N_RESET pin) forces the default coefficients in all filters when asserted (Low).

### 4.7.7 Auto Power Save Mode

By setting the AutoClkStopMode bit in the ClkStopCtrl Register, the serial interface will enter an automatic power down mode. In this mode, if no serial port activity on the CmdFS is detected after a time out (TMO) period the serial interface will enter a standby state. In this state all master clock activity within the interface is stopped (to reduce power to a minimum) and the SClk pin stops in the high state. It will remain in this state until the user asserts the CmdFS pin for at least one MCLK cycle time, when normal serial port activity will recommence and serial port operation can continue as normal. Subsequent periods of TMO without CmdFS activity will cause the serial interface to enter power down mode again.

The time out period TMO is fixed internally to 4096 master clock periods (444µs when using a 9.216MHz master clock).

When in the power down state and the SClk pin is high, the CmdFS pin may be asserted asynchronously but, when the SClk re-starts, subsequent CmdFS strobes must respect the timing constraints given in the timing section of this document. The serial interface is stopped in the state where it tests the CmdFS pin for a high state, so re-starting from this point by asserting CmdFS will begin a serial operation cycle in the interface logic.

Applying global reset while in the power down state will return the device to normal serial mode.

The use of Auto Power Save mode, by setting the AutoClkStopMode bit, is available only in low data rate mode (set DataRateHi bit of ConfigCtrl1 Register inactive), as this mode is envisaged for use in low speed/low power applications. However, systems that use high data rate mode can make use of this facility by setting a low data rate (set DataRateHi bit of ConfigCtrl1 Register inactive) before asserting the AutoClkStopMode bit, then returning to the high data rate mode by setting the DataRateHi bit active.

## 4.8 Register Description

This section describes in detail each of the registers and access points addressed by the Command Control Serial Word.

### 4.8.1 Key to Register Map

Each section that follows describes in detail the operation and use of each of the registers in the device. The registers are split into their functional groups, grouping associated registers together. Each section consists of a Title, an Address, a Function Reference Field, a Description, and a Bit Specification.

The Function Reference Field describes the overall access available to this section (RW/W/R, where R = Read and W = Write).

The Bit Specification describes the function of each individual bit, or a range of bits within a register. There is a separate line for each distinct field of bits. The State column indicates the action available to each group of bits (RW/W/R). Address and data format illustrations show the bit positions in multiple-byte transfers. “R” indicates a reserved bit, which should be set to logic zero when writing. Its value is undefined when read. “X” indicates a don’t know/don’t care state.

### 4.8.2 Register Reset State

All I/O access points (both read and write) are reset to logic zero on taking N_RESET Low, except where explicitly shown in this document. The reset state of status bits will depend on the level of the status signal being monitored. Other registers (both read and write) are not affected by taking N_RESET Low.

### 4.8.3 Register and Access Point Summary

| ***Control and Status Registers (Universal Access)*** | | | |
|---|---|---|---|
| **Page** | | | |
| 21 | $0x00 | ConfigCtrl1 | Configuration control register 1 |
| 22 | $0x01 | ConfigCtrl2 | Configuration control register 2 |
| ***Control and Status Registers (Page 0)*** | | | |
| 23 | $0x02 | PowerDownCtrl | Power control register |
| 24 | $0x03 | TxSetup | Transmit setup register |
| 25 | $0x04-$0x07 | TxData | Transmit data registers |
| 28 | $0x08 | RxSetup1 | Receive setup control register 1 |
| 29 | $0x09 | RxSetup2 | Receive setup control register 2 |
| 29 | $0x0A | AnaCtrl | Analog configuration control register |
| 30 | $0x0B | AuxAdcCtrl | Auxiliary ADC data converter control register |
| 33 | $0x0C | RamDacCtrl | Ram Dac control register |
| 36 | $0x0D | LoopBackCtrl | Loopback control register |
| 37 | $0x0E | TxErrorStatus | Transmit error status register |
| 38 | $0x0F | TxErrStatMask | Transmit error status interrupt mask register |
| 46 | $0x3C | ClkStopCtrl | Clock-Stop Control register |
| ***Auxiliary Function Registers*** | | | |
| 31 | $0x10-$0x17 | AuxAdcData | Auxiliary ADC data registers |
| 34 | $0x18-$0x1F | AuxDacData | Auxiliary DAC data registers |
| ***Status and Interrupt Registers*** | | | |
| 39 | $0x20 | RxErrorStatus | Receive error status register |
| 40 | $0x21 | RxErrorStatMask | Receive error status interrupt mask register |
| 41 | $0x22 | TxFIFOStatus | Transmission data FIFO status register |
| 42 | $0x23 | TxFIFOStatMask | Tx data FIFO status interrupt mask register |
| 43 | $0x24 | SymClkPhase | Tx Symbol clock phase adjustment register |
| | $0x25 | | Not Used |
| ***Memory I/O Access Points*** | | | |
| 44 | $0x26-$0x2D | CoeffRamData | Coefficient memory I/O access addresses |
| | $0x2E-$0x2F | | Not Used |
| ***Rx Data Path Registers*** | | | |
| 55 | $0x30-$0x31 | RxIQGainMult | Receive I channel gain attenuation registers |
| 56 | $0x32-$0x33 | RxIQOffset | Receive I channel offset correction registers |
| 55 | $0x34-$0x35 | RxIQGainMult | Receive Q channel gain attenuation registers |
| 56 | $0x36-$0x37 | RxIQOffset | Receive Q channel offset correction registers |
| ***Rx Data Path Access Points*** | | | |
| 57 | $0x38-$0x39 | RxDataAccess | Receive path data access point (I) |
| 57 | $0x3A-$0x3B | RxDataAccess | Receive path data access point (Q) |
| | $0x3D-$0x3F | | Not Used |

| | | | |
|---|---|---|---|
| ***TX Data Path Registers*** | | | |
| 52 | $0x40-$0x41 | TxPhase | Transmit I channel phase correction registers |
| 50 | $0x42-$0x43 | TxIQGainMult | Transmit I channel gain attenuation registers |
| 51 | $0x44-$0x45 | TxIQOffset | Transmit I channel offset correction registers |
| 52 | $0x46-$0x47 | TxPhase | Transmit Q channel phase correction registers |
| 50 | $0x48-$0x49 | TxIQGainMult | Transmit Q channel gain attenuation registers |
| 51 | $0x4A-$0x4B | TxIQOffset | Transmit Q channel offset correction registers |
| 48 | $0x4C-$0x4D | TxRampUpInc | Transmit ramp-up increment registers |
| 49 | $0x4E-$0x4F | TxRampDnDec | Transmit ramp-down decrement registers |
| ***Tx Data Path Access Points*** | | | |
| 54 | $0x50-$0x51 | TxDataAccess | Transmit path data access point (I) |
| 54 | $0x52-$0x53 | TxDataAccess | Transmit path data access point (Q) |
| | $0x54-$0x5F | | Not Used |
| ***Self Test Registers*** | | | |
| 59 | $0x60-$0x61 | BISTPRSG | Built-in self test pseudo-random sequence generator |
| 58 | $0x62 | BISTControl | Built-in self test control register |
| | $0x63 | | Not Used |
| 60 | $0x64-$0x6D | BISTCRCRegisters | Built-in self test cyclic redundancy code checkers |
| | $0x6E-$0x6F | | Not Used |
| ***SRAM Memory Access Points*** | | | |
| 47 | $0x70-$0x73 | SramData | Auxiliary DAC1 memory I/O access addresses |
| | $0x74-$0x7F | | Not Used |
| ***Direct Write Registers (Page 1)*** | | | |
| 62 | $0x10-$0x1F | DirectWrite79tapI | Direct write access to 79-tap I channel filter |
| 63 | $0x20-$0x2F | DirectWrite79tapQ | Direct write access to 79-tap Q channel filter |

**Notes:**

1. Addresses $0x80 to $0xFF cannot be used as the MSB controls the direction of data flow:
2. “1” = High = Read and “0” = Low = Write.

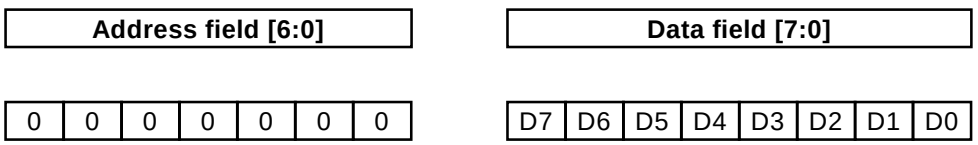
#### 4.8.3.1 UNIVERSAL REGISTERS ACCESSIBLE IN ALL PAGES

**ConfigCtrl1**

Title: Configuration Control register
Address: $0x00
Function: RW
Description: General configuration bits, together with operational control signal bits.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | *DataRateHi* | High | RW | When set active all serial port data transfers will be at half of the master clock rate. When inactive, all serial port data rates will be at a quarter of the master clock rate. This has the effect of altering the Rx sample output rate from 8 times the symbol rate when active to 4 times when inactive. |
| 6 | *TxHandshakeEn* | High | RW | When set active enables the transmit hardware interlock protocol, thereby stopping the *Serial Clock* (SClk) if the transmit path is enabled and the transmit FIFO is full. |
| 5 | *BiDirCmdPortEn* | High | RW | When this bit is set active the *Cmd* port will drive its data line out of the chip for the last 8 bits of read operations. When set inactive command read data will be returned on either the *Rx* or the *CmdRd* port (default). |
| 4 | *RxDataForCmdRdEn* | High | RW | This bit only takes effect if the *BiDirCmdPortEn* bit is inactive. When set active this bit causes all command read operations to respond with data on the *Rx* serial port. When set inactive the command read data will be output via the *CmdRd* port (default). |
| (5,4) | *CommandReadDataMode* | 00<br>01<br>10, 11 | RW | The *BiDirCmdPortEn* bit and *RxDataForCmdRdEn* bit together control the method by which command read data is returned to the user.<br>(Default) Read data returned on *CmdRd* port.<br>Read data returned on *Rx* port and *CmdRd port*<br>Read data returned on *Cmd* port. |
| 3 | *LowRxRdFS* | High | RW | When set active both the CmdRdFS and the RxFS output pins will be driven active low, when set inactive the two frame sync's will be driven active high (default). |
| 2 | *RxDataPortDisable* | High | RW | When set active tristates the RxDat and RxFS pins. |
| 1 | *RdCmdPortDisable* | High | RW | When set active tristates the CmdRdDat and CmdRdFS pins. |
| 0 | *SymboModuBypass* | High | RW | Setting this bit bypasses the modulator, thereby taking the least significant 3 bits of each Command Transmit Data Serial Word received via the serial interface to represent an absolute constellation mapping. |

**Address and Data format for ConfigCtrl1 access**



### ConfigCtrl2

Title: Configuration Control register
Address: $0x01
Function: RW
Description: General configuration bits, together with operational control signal bits.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7:6 | *PageAddress* | Data | RW | Page address field. All registers except ConfigCtrl1 and ConfigCtrl2 use these bits to decode their actual address. Reset defaults to page 0. The page address field applies to all further commands, until a different page address is selected. |
| 5 | *n_SlowDown* | Low | RW | When active, this bit reduces the slew rate of digital outputs. This reduces power consumption; ground bounce and reflection problems associated with fast edges on poorly terminated lines. De-activation speeds up the digital outputs, but increases power consumption, ground bounce and reflection problems. It is anticipated that the latter mode will be used only in 3.3V systems. |
| 4 | *SRamIoRdInc* | High | RW | This bit determines whether a read or write operation to the Auxiliary SRAM will increment the address pointers. When set active causes read operations to move the address pointer on, this would therefore allow an efficient write then read verify scheme to be used. When set inactive write operations increment the address pointer. |
| 3 | *SRamIoEn* | High | RW | When set active allows read/write access to the Auxiliary SRAM. This bit should not be activated when the SRAM is being accessed by the RamDac. When this bit is set active, the first access to the SramData Register will access the first SRAM address location. Subsequent read or write accesses will increment the address pointer to the next memory location. |
| 2 | *CoeffRamIoRdInc* | High | RW | This bit determines whether a read or write operation to a coefficient memory will increment the address pointers. When set active the address pointer is incremented by any coefficient ram read operation, thereby allowing a write then read verification. When set inactive, write operations increment the address pointer. |
| 1 | *CoeffRamIoEn* | High | RW | When set active allows read/write access to all the coefficient memories. This bit is valid only when the Tx and Rx Data paths are inactive. When this bit is set active, the first access to any of the coefficient memories will access the first coefficient location (A1). Subsequent read or write accesses to any coefficient memory will increment the address pointers for all the coefficient memories. |
| 0 | *n_BigEndData* | Low | RW | When set active causes serial port read data, from the Rx port to be generated with the MSB data bit as the first serial word bit. If inactive, the LSB is first. On taking N_RESETLow this bit is active (i.e. the default is MSB first). |

**Address and Data format for ConfigCtrl2 access**

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |

| Data field [7:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

#### 4.8.3.2 PAGE 0 ADDRESSED REGISTERS

**PowerDownCtrl**

Title: Power Control register
Address: $0x02
Function: RW
Description: This register, together with the following bits, controls the power saving features:

| | | |
|---|---|---|
| *TxCtrlEn* | bit of register | **TxSetup** |
| *TxClkStop* | bit of register | **TxSetup** |
| *TxEn* | bit of register | **TxSetup** |
| *RxIFClkStopMode* | bit of register | **ClkStopCtrl** |
| *Aux_ClkStopMode* | bit of register | **ClkStopCtrl** |
| *AutoClkStopMode* | bit of register | **ClkStopCtrl** |
| *RxClkStop* | bit of register | **ClkStopCtrl** |
| | bits [5:0] of register | **AuxAdcCtrl** |
| *DataRateHi* | bit of register | **ConfigCtrl1** |
| *TxHandshakeEn* | bit of register | **ConfigCtrl1** |
| *RxEn* | bit of register | **RxSetup1** |

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | *BiasChainPowDn* | Low | RW | When set active powers down the analog bias chain. |
| 6 | *BiaslCtrl* | High | RW | When set active, increases Tx and Rx analog bias currents. |
| 5 | *BiasPowDn* | Low | RW | When set active powers down the analog bias section. |
| 4 | *AuxDac4PowDn* | Low | RW | When set active powers down Auxiliary Dac4. |
| 3 | *AuxDac3PowDn* | Low | RW | When set active powers down Auxiliary Dac3. |
| 2 | *AuxDac2PowDn* | Low | RW | When set active powers down Auxiliary Dac2. |
| 1 | *AuxDac1PowDn* | Low | RW | When set active powers down Auxiliary Dac1. |
| 0 | *RxAafPowDn* | Low | RW | When set active powers down the receive analog anti-alias filter (AAF). |

**Address and Data format for PowerDownCtrl access**

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Data field [7:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### TxSetup

Title: Transmit Setup register
Address: $0x03
Function: RW
Description: Sets up the transmit functions.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | | | RW | Reserved. Set this bit Low. Undefined on read. |
| 6 | *TxCtrlEn* | Low | RW | When set active enables the Tx control logic clock (default state). When inactive removes Tx control logic clock and reduces power consumption. Ensure bit is active (Low) before commencing Tx FIFO operations. |
| 5 | *TxDirectWriteEn* | High | RW | When asserted, enables direct write to the I and Q 79-tap filter inputs. This bypasses the FIFO and DQPSK modulator. See the Page 1 addressed registers. |
| 4 | *LinearRamp* | High | RW | When asserted (high) selects Linear Ramping, default (Low) selects Sigmoidal Ramping. |
| 3 | *TxClkStop* | High | RW | When set active causes the *TxEn* bit to also be used to gate the Tx Data path master clock. When inactive (default state) the Tx Data path master clock is always supplied. |
| 2 | *TxEn* | High | RW | When set active, enables the Tx Data path, allowing transmission to start when the correct enable sequence has been seen. This bit may only be cleared when the *TxPathEn* status bit in the TxFIFOStatus Register is inactive, setting inactive during a transmission cycle will cause erroneous behaviour. This bit also acts as a transmit section power enable bit. |
| 1 | *TxRampEn* | High | RW | When set active, this bit enables the transmit amplitude ramping function. Ramping is then controlled by the *TxRampUp* bit of the TxData Register When this bit is inactive, the *TxRampUp* bit will directly control the transmit amplitude (High meaning full amplitude, Low meaning zero amplitude). |
| 0 | *TxFirCoeffReset* | Low | RW | When set active this bit forces all the Tx Data path filters to load their default coefficient values. This bit will be set active on taking N_RESET Low, and therefore needs to be deactivated before default filter coefficients can be overwritten. |

**Address and Data format for TxSetup access**

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |

| Data field [6:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| R | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

**TxData**

Title: Transmit Data register
Address: $0x04 - $0x07 (Mapped over four locations, two address bits being used as data bits)
Function: W FIFO input
R FIFO output
Description: This transmit data register is 10 bits wide. The two least significant bits of the address bus are used to drive bits 8 and 9, hence, it can be considered to be mapped over four consecutive locations. This data word is written into a four-word deep FIFO. The FIFO will be read when the Tx Data path demands data. This will only occur when the TxEn bit of the TxSetup Register is set active. For test purposes, the FIFO data output may be accessed by reading these registers.

**Data write with symbol modulator not bypassed**

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 9 | *TxRampUp* | High | W | This bit is written to the FIFO. While the *TxEn* bit of the TxSetup Register is active, it controls the Tx Data path ramping. Setting it active will cause the amplitude to ramp up to its full value, conversely setting the bit inactive will cause the amplitude to ramp down to its minimum value. If the bit is changed while the amplitude is being ramped, the ramp direction will change to the direction set by this bit.<br>While the *TxRampEn* bit is inactive, the *TxRampUp* bit will directly control the transmit amplitude (High meaning full amplitude and Low meaning zero amplitude). |
| 8 | *MultiSymbol* | High | W | This bit is written to the FIFO and when this bit is set active, the FIFO symbol data will be marked as a four symbol word. When set inactive, the FIFO symbol data will be marked as a single symbol word. This bit is inactive if the *SymbModuBypass* bit of the ConfigCtrl1 Register is active. |
| 7:6 | *TxRelSymbol4* | Data | W | Fourth symbol in word to be written to FIFO. |
| 5:4 | *TxRelSymbol3* | Data | W | Third symbol in word to be written to FIFO. |
| 3:2 | *TxRelSymbol2* | Data | W | Second symbol in word to be written to FIFO. |
| 1:0 | *TxRelSymbol1* | Data | W | First symbol in word to be written to FIFO. |

**Data write with symbol modulator bypassed**

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 9 | *TxRampUp* | High | W | (See above) |
| 8:3 | (not used) | Data | W | Redundant data that is still written into the FIFO. Set these bits Low. |
| 2:0 | *TxAbsSymbol* | Data | W | IQ constellation point that is written into the FIFO. |

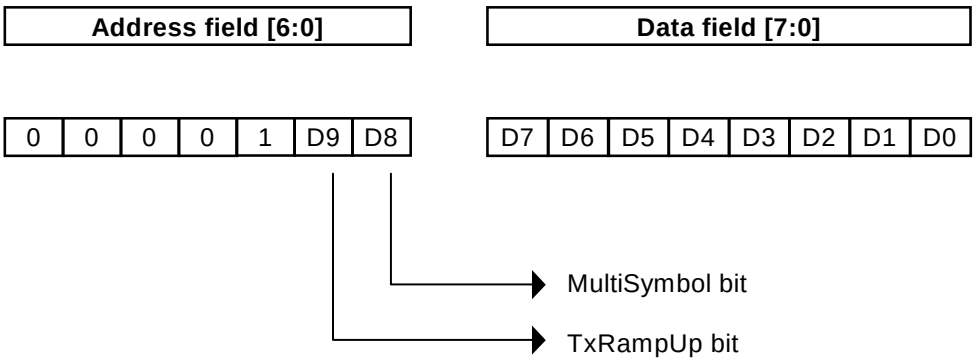
**Read operation**

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x04*** | | | | |
| 7:2 | | | | Reserved. Bit values are not defined. |
| 1:0 | *UpperFIFORdData* | Data | R | Reads address access bits 9 and 8 of the FIFO data output register, these are placed in bits 1 and 0. |
| ***Address $0x05*** | | | | |
| 7:0 | *LowerFIFORdData* | Data | R | Reads address access bits 7 to 0 of the FIFO data output register. Reading this location also performs a FIFO read operation, thereby moving the next (if any) FIFO data location into the FIFO data output register. |
| ***Address $0x06 and $0x07*** | | | | |
| 7:0 | | | R | Reserved. Bit values are not defined. |

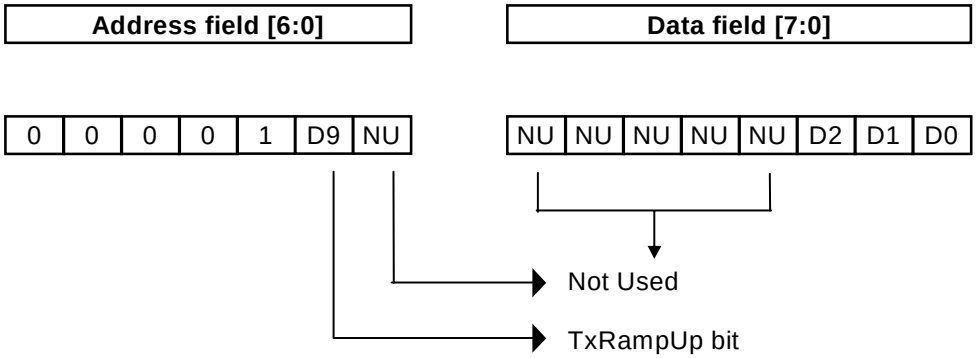
For these read operations to be valid, the Tx Data path must be active *(TxEn* bit of **TxSetup** Register set active) and the *SymbModuBypass* bit of the **ConfigCtrl1** Register must also be set active.

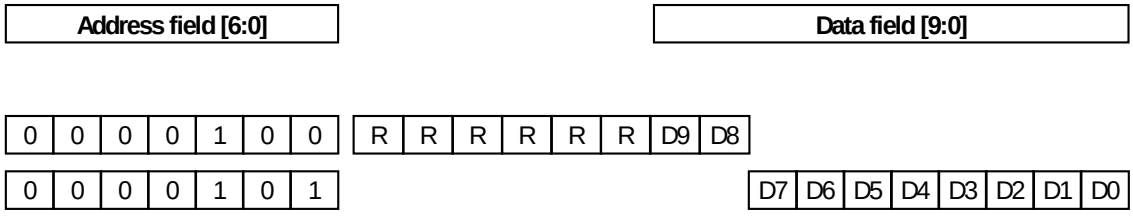
## Address and Data format for TxData Write access




## Address and Data format for TxData (Modulator Bypass Mode) Write access




## Address and Data format for TxData Read access



### RxSetup1

Title: First Receive Setup control register
Address: $0x08
Function: RW
Description: Receive path setup and initialization control bits.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | *Rx32BitMode* | High | RW | When set active, the *Rx* port operates on 32-bit frames - I data in the MSB word, Q data in the LSB word. |
| 6 | *RxSampleSel* | Data | RW | This bit is used to select which pair of I,Q samples is supplied from the possible two pairs when the *DataRateHi* bit in ConfigCtrl1 Register is in the low mode (inactive). It has no effect when *DataRateHi* is active. When set High, this bit selects odd-numbered samples ($I_1$, $Q_1$), ($I_3$, $Q_3$) etc. and when set Low this bit selects the even-numbered samples, where the sampling starts with ($I_0$, $Q_0$) after *RxEn* is set active. |
| 5 | *RxIdentMode* | High | RW | When asserted this bit causes the received data to carry an identification bit in the LSB of the data word. Q channel data is identified by a logic '1' and I channel data by the internal symbol clock phase, which is logic '0' for seven out of eight samples and logic '1' for the other sample. In low data rate mode, the symbol clock may coincide with the discarded symbol; this can be rectified by toggling the *RxSampleSel* bit. The received data dynamic range is therefore reduced from 16 to 15 bits in this mode. The user can swap modes during a receive data burst without affecting Rx operations. |
| 4 | *RxEn* | High | RW | When set active, enables the Rx Data path, which then processes the signals on the IRXP, IRXN and QRXP, QRXN pins, outputting results via the *RxData* serial port. This bit also acts as a receive section power enable bit. |
| 3 | *RxBistActive* | High | RW | When set active, enables Rx Built-In Self Test operation. |
| 2 | *AnaAdcReset* | Pulse | W | When this bit is set High, a 4-clock-cycle ADC auto reset event is generated. It is not necessary to clear this bit before another ADC auto reset event is initiated. |
| | | | R | The read state of this bit indicates the logic level last written to this bit. It does not have a functional significance and is only available for test purposes. |
| 1 | *AnaEnAutoReset* | Low | RW | When active this bit enables the ADC auto reset function. On taking N_RESET Low, this bit is set active, which is the default operating condition. |
| 0 | *RxFirCoeffReset* | Low | RW | When set active forces all the Rx Data path filters to load their default coefficient values. This bit will be set active on taking N_RESET Low, and therefore needs to be deactivated before default filter coefficients can be overwritten. Normal filter operation is unaffected by leaving this bit set. |

**Address and Data format for RxSetup1 access**

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Data field [7:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### RxSetup2

Title: Second Receive Setup control register
Address: $0x09
Function: RW
Description: Receive I and Q vernier control bits.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7:4 | *QvernierDelay* | High | RW | Q channel vernier sampling delay, allowing the sampling point to be adjusted to a resolution of 4 master clock periods. |
| 3:0 | *IvernierDelay* | High | RW | I channel vernier sampling delay, allowing the sampling point to be adjusted to a resolution of 4 master clock periods. |

Note: The values are in the format of 4-bit signed 2s-complement integers - the MSB being the sign. Therefore the function can be interpreted as adjusting the reference phase by ± 28 master clock periods.

**Address and Data format for RxSetup2 access**

| Address field [6:0] | | | | | | | Data field [7:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### AnaCtrl

Title: Analog configuration Control register
Address: $0x0A
Function: RW
Description: Reserved. All bits should be set Low.

### AuxAdcCtrl

Title: Auxiliary ADC data converter Control register
Address: $0x0B
Function: RW
Description: This register controls the operation of the four ADC channels. These are implemented using a single ADC converter, which is multiplexed on to each of the ADC channels. A conversion cycle consists of performing a conversion for each of the active channels in turn, in the order channel 1, channel 2, channel 3, channel 4. Note that when no channel is active, or when in one-shot mode and conversions on all active channels have completed, the analog ADC circuitry is automatically powered down.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | | | RW | Reserved. This bit should be set Low. Undefined on read. |
| 6 | *AdcConvertRate* | High | RW | This bit changes the ADC conversion rate. If this bit is set Low, the ADC is clocked by MCLK/8, yielding a conversion time of 80x MCLK periods per ADC channel. The maximum sample rate is lower than this. With a single channel selected, the maximum rate is MCLK/84 samples/second. Setting this bit high will halve the ADC clock rate, and hence double the conversion time. |
| 5 | *AdcContConv* | High | RW | Continuous conversion mode control bit; when inactive, sets the ADCs into one-shot conversion mode; when active, the ADCs will continuously convert. One-shot conversion mode is initiated by the *StartConvert* bit. In continuous convert mode, the ADC will start a new conversion cycle on all active channels after the previous cycle has completed. |
| 4 | *EnableAdc4* | High | RW | Setting this bit high will enable ADC channel 4 for conversion. This bit may be updated at any time, but will only change the active state of the ADC channel for the next time it is converted. |
| 3 | *EnableAdc3* | High | RW | Setting this bit high will enable ADC channel 3 for conversion. This bit may be updated at any time, but will only change the active state of the ADC channel for the next time it is converted. |
| 2 | *EnableAdc2* | High | RW | Setting this bit high will enable ADC channel 2 for conversion. This bit may be updated at any time, but will only change the active state of the ADC channel for the next time it is converted. |
| 1 | *EnableAdc1* | High | RW | Setting this bit high will enable ADC channel 1 for conversion. This bit may be updated at any time, but will only change the active state of the ADC channel for the next time it is converted. |
| 0 | *StartConvert* | High | | One-shot conversion control bit. Only valid when the ADCs are set to one-shot conversion mode. |
| | | | W | Setting this bit High starts the ADC data conversion. Setting this bit Low will stop the conversion. This should only be used for test purposes, because the ADC conversion logic will automatically set this bit Low when the conversion operation has completed. |
| | | | R | This bit can be set High or Low by the serial interface, but the ADC conversion logic will automatically set it Low when the current conversion cycle has completed. |

**Address and Data format for Auxillary ADC Control access**

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 1 |

| Data field [6:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| R | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### AuxAdcData

Title: Auxiliary ADC Data registers
Address: (Eight registers) $0x10 to $0x17
Function: R
Description: These registers enable the user to inspect the conversion value for each of the four auxiliary ADCs. There are two read registers per ADC, one to obtain the least significant two bits of the data, the other for the most significant eight bits. Reading these registers does not affect the ADC conversion cycle. Reading the MSB read register directly reads the ADC output and simultaneously causes the two bits in the LSB read register to be written into a holding register. This holding register is read when the LSB read register is read. This mechanism is necessary to allow the user to read MSB and LSB data from the same ADC conversion cycle. If only the MSB read register is read, the converter can be considered as an 8-bit ADC. If a 10-bit conversion is required, the MSB read register must be read first.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x10*** | | | | |
| 7:2 | | | R | Reserved. Bit Values are not defined |
| 1:0 | *Adc1LsbData* | Data | R | Least significant two bits of the data from the last conversion of *AuxAdc1*. |
| ***Address $0x11*** | | | | |
| 7:0 | *Adc1MsbData* | Data | R | Most significant eight bits of the data from the last conversion of the *AuxAdc1.* |
| ***Address $0x12*** | | | | |
| 7:2 | | | R | Reserved. Bit values are not defined. |
| 1:0 | *Adc2LsbData* | Data | R | Least significant two bits of the data from the last conversion of the *AuxAdc2*. |
| ***Address $0x13*** | | | | |
| 7:0 | *Adc2MsbData* | Data | R | Most significant eight bits of the data from the last conversion of the *AuxAdc2.* |
| ***Address $0x14*** | | | | |
| 7:2 | | | R | Reserved. Bit values are not defined. |
| 1:0 | *Adc3LsbData* | Data | R | Least significant two bits of the data from the last conversion of *AuxAdc3.* |
| ***Address $0x15*** | | | | |
| 7:0 | *Adc3MsbData* | Data | R | Most significant eight bits of the data from the last conversion of the *AuxAdc3.* |
| ***Address $0x16*** | | | | |
| 7:2 | | | R | Reserved. Bit values are not defined. |
| 1:0 | *Adc4LsbData* | Data | R | Least significant two bits of the data from the last conversion of the *AuxAdc4*. |
| ***Address $0x17*** | | | | |
| 7:0 | *Adc4MsbData* | Data | R | Most significant eight bits of the data from the last conversion of the *AuxAdc4*. |

## Address and Data format for Auxillary ADC Data access

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | N1 | N0 | 0 |
| 0 | 0 | 1 | 0 | N1 | N0 | 1 |

| Data field [9:0] | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | D1 | D0 |
| D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | |

| R | R | R | R | R | R |
|---|---|---|---|---|---|

| N1 | N0 | ADC Channel |
|---|---|---|
| 0 | 0 | Channel 1 |
| 0 | 1 | Channel 2 |
| 1 | 0 | Channel 3 |
| 1 | 1 | Channel 4 |

**RamDacCtrl**

Title: RamDac Control register
Address: $0x0C
Function: RW
Description: This register controls the operation of DAC 1, together with the operation of the memory (DacSram) which can be used to drive the digital input of DAC 1.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7:6 | | | RW | Reserved. These bits should be set Low. Undefined on read. |
| 5:3 | *RamDacRate* | High | RW | These three bits set the rate at which the RamDac memory's DAC access address pointer changes. The three bit value (RamDacRate) causes a change rate of ($36 \times 2^{RamDacRate}$) kHz. See table below. |
| 2 | *RamDacInc* | High | RW | This bit activates the RamDac memory scan operation. Setting it active will cause the memory address to increment up to the top (highest) location, conversely setting the bit inactive will cause the memory address to decrement down to the bottom location. If the bit is changed while the memory is being scanned, the current scan will complete before the new state of the *RamDacInc* bit takes effect. |
| 1 | *AutoCycle* | High | RW | This bit is only valid if the *RamDacActive* bit is active. When set active, the Auxiliary SRAM memory will be continually scanned at the rate set by the *RamDacRate* bits. This enables a symmetrical periodic waveform to be driven out on the AUXDAC1 pin. The Auxiliary SRAM address cycles from the bottom location up to the top location, and back down to the bottom again. |
| 0 | *RamDacActive* | High | RW | DAC 1 input mode bit. When inactive, the AuxDacData Registers (offsets 0 and 1) are used as the source for conversion. If this bit is active, the DAC is driven from the output of the RamDac memory. |

| RamDacCtrl[5:3] | Dac Update Frequency (Relative to MCLK) |
|---|---|
| 0 0 0 | MCLK/512 |
| 0 0 1 | MCLK/256 |
| 0 1 0 | MCLK/128 |
| 0 1 1 | MCLK/64 |
| 1 0 0 | MCLK/32 |
| 1 0 1 | MCLK/16 |
| 1 1 0 | MCLK/8 |
| 1 1 1 | MCLK/4 |

**Table 5: Ram Dac Rate Select Table**

**Address and Data format for RamDacCtrl access**

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |

| Data field [5:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| R | R | D5 | D4 | D3 | D2 | D1 | D0 |

### AuxDacData

Title: Auxiliary DAC Data registers
Address: (Eight registers) $0x18 to $0x1F
Function: RW
Description: There are two input registers for each of the four auxiliary DACs. Writing to the *AuxDac#LsbData* register writes the least significant two bits of DAC data. Writing to the *AuxDac#MsbData* register writes the most significant eight bits of DAC data and then passes all ten bits to the appropriate DAC input (only if the *RamDacActive* bit is set Low for DAC 1). If the *AuxDac#MsbData* register is written while the *AuxDac#LsbData* register is left constant, the converter may be treated as an 8-bit DAC.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x18*** | | | | |
| 7:2 | | | RW | Reserved. These bits should be set Low. Undefined on read. |
| 1:0 | *AuxDac1LsbData* | Data | RW | Writing to this address writes the least significant two bits of the *DacData1* register. These two bits may be read for test purposes. |
| ***Address $0x19*** | | | | |
| 7:0 | *AuxDac1MsbData* | Data | RW | Writing to this address writes the most significant eight bits of the *DacData1* register and updates DAC 1. This register may also be read for test purposes. |
| ***Address $0x1A*** | | | | |
| 7:2 | | | RW | Reserved. These bits should be set Low. Undefined on read. |
| 1:0 | *AuxDac2LsbData* | Data | RW | Writing to this address writes the least significant two bits of the *DacData2* register. These two bits may be read for test purposes. |
| ***Address $0x1B*** | | | | |
| 7:0 | *AuxDac2MsbData* | Data | RW | Writing to this address writes the most significant eight bits of the *DacData2* register and updates DAC 2. This register may also be read for test purposes. |
| ***Address $0x1C*** | | | | |
| 7:2 | | | RW | Reserved. These bits should be set Low. Undefined on read. |
| 1:0 | *AuxDac3LsbData* | Data | RW | Writing to this address writes the least significant two bits of the *DacData3* register. These two bits may be read for test purposes. |
| ***Address $0x1D*** | | | | |
| 7:0 | *AuxDac3MsbData* | Data | RW | Writing to this address writes the most significant eight bits of the *DacData3* register and updates DAC 3. This register may also be read for test purposes. |
| ***Address $0x1E*** | | | | |
| 7:2 | | | RW | Reserved. These bits should be set Low. Undefined on read. |
| 1:0 | *AuxDac4LsbData* | Data | RW | Writing to this address writes the least significant two bits of the *DacData4* register. These two bits may be read for test purposes. |
| ***Address $0x1F*** | | | | |
| 7:0 | *AuxDac4MsbData* | Data | RW | Writing to this address writes the most significant eight bits of the *DacData4* register and updates DAC 4. This register may also be read for test purposes. |

## Address and Data format for Auxillary DAC Data access

| Address field [6:0] | | | | | | | Data field [9:0] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | N1 | N0 | 0 | | | | | | | | R | R | R | R | R | R | D1 | D0 |
| 0 | 0 | 1 | 1 | N1 | N0 | 1 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | | | | | | | |

| N1 | N0 | Channel Selected |
|---|---|---|
| 0 | 0 | Channel 1 |
| 0 | 1 | Channel 2 |
| 1 | 0 | Channel 3 |
| 1 | 1 | Channel 4 |

### LoopBackCtrl

Title: LoopBack Control register
Address: $0x0D
Function: RW
Description: This register is only used for test purposes. For normal operation all these bits should be inactive.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7:6 | | | RW | Reserved. These bits should be set Low. Undefined on read. |
| 5 | *FirReset* | High | RW | When active, this bit holds all FIR filters in reset, by resetting the FIR address pointers. This by itself does not reset the data register RAMs. A separate access is provided to disable the complete Tx or Rx Data path. Taking N_RESET Low will also reset the FIR filter coefficients to their default values. |
| 4 | *DigLoopBack* | High | RW | When set active this bit enables the digital loopback feature. This connects the output of the final Rx Data path 63-tap FIR filter to the input of the Tx Data path 63-tap FIR filter, thereby allowing an analog signal presented at the Rx inputs to be filtered by a raised cosine filter and monitored at the Tx outputs as an analog signal. |
| 3 | *AnaLoopBack* | High | RW | When set active this bit enables the analog loopback feature. This connects the output of the Tx Data path DAC to the input of the Rx Data path ADC, thus passing transmit constellation data through a raised cosine filter and allowing the resultant data samples to be monitored digitally at the Rx output. |
| 2 | *RxDPAccessSel* | High | RW | When set active this bit disables the Rx Data path sample clock, thereby enabling the Data path access register to directly update the output of the Rx Data path operator. |
| 1 | *TxDPAccessSel* | High | RW | When set active this bit disables the Tx Data path sample clock, thereby enabling the Data path access register to directly update the input to the Sigma-Delta DAC without the data being overridden by subsequent sample clocks |
| 0 | *TxtoRxDataPath* | High | RW | When set active this bit connects the output of the Tx Gain, Phase, Ramping and Offset Adjustment block to the Rx Data interface, enabling the digital Tx output to be observed on the *RxData* port in real time. Data is taken from the I and Q channels on alternate MCLK/64 sample clocks, with the phase controlled by the *EvenSamplePhase* bit of the RxErrorStatus Register. Note that if the corresponding mask bit for the *EvenSamplePhase* bit is set inactive, an interrupt will be generated on the next I-phase Tx data. The 14 bits of Tx data are output in the least significant 14 bits of the serial word on the *RxData* port, so that the sign of the Tx data is bit 13. The *TxtoRxDataPath* bit is mainly intended for manufacturer's test purposes and should be set Low. |

**Address and Data format for LoopBackCtrl access**

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 0 | 1 |

| Data field [5:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| R | R | D5 | D4 | D3 | D2 | D1 | D0 |

### TxErrorStatus

Title: Transmit Error Status register.
Address: $0x0E
Function: R
Description: This register is the Tx Data path error status register. The *TxIrqActive* bit is set active when one of the other bits in this register is the source of an interrupt event. All these error conditions are caused by transitory events, therefore the error condition is latched (marked with an 'L'). Reading this status register causes all latched bits to be set inactive, unless an error event is currently pending.
Setting any bit of this register High will cause an interrupt to be generated (N_IRQ will be set Low) if the source of the interrupt has not been masked in the corresponding Mask register.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | | | R | Reserved. Bit value is not defined. |
| 6 | *TxDataPathQOF* | High | RL | Data path gain, phase and offset (GPO) adjustment-unit: Q channel overflow error status bit. |
| 5 | *TxDataPathIOF* | High | RL | Data path gain, phase and offset (GPO) adjustment-unit: I channel overflow error status bit. |
| 4 | *DecimationOF* | High | RL | The Rx path decimation filter accumulator overflow error status bit. (Note: For optimisation of the chip design, this Rx control bit is located in a Tx control register) |
| 3 | *SymbolClkEn* | Data | R | Direct access to internal symbol clock enable signal. Allows this timing reference to appear on the IRQ pin when unmasked in the TxErrStatMask Register |
| 2 | *Tx63tapOF* | High | RL | 63-tap I and Q Tx filter data accumulator overflow error status bit. |
| 1 | *Tx79tapOF* | High | RL | 79-tap I and Q Tx filter data accumulator overflow error status bit. |
| 0 | *TxIrqActive* | High | RL | This bit is set High if there is an active interrupt caused by one of the status bits in this register. |

**Address and Data format for TxErrorStatus access**

| Address field [6:0] | | | | | | | Data field [6:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | R | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### TxErrStatMask

Title: Transmit Error Status interrupt Mask register
Address: $0x0F
Function: RW
Description: Masks interrupts in the TxErrorStatus Register. On taking N_RESET Low, these bits are set active, so masking out all possible interrupt sources. Each bit which is taken inactive will allow its associated status bit, when active, to generate an interrupt.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | | Data | RW | Reserved for manufacturer's test purposes. This bit should be set Low. |
| 6 | *n_TxDataPathQOF_Mask* | Low | RW | GPO Q channel error interrupt mask bit. |
| 5 | *n_TxDataPathIOF_Mask* | Low | RW | GPO I channel error interrupt mask bit. |
| 4 | *n_DecimationOF_Mask* | Low | RW | Decimation filter error interrupt mask bit. |
| 3 | *n_SymbolClkEn_Mask* | Low | RW | Symbol clock reference signal mask bit. |
| 2 | *n_Tx63tapOF_Mask* | Low | RW | 63-tap I and Q filter error interrupt mask bit. |
| 1 | *n_Tx79tapOF_Mask* | Low | RW | 79-tap I and Q filter error interrupt mask bit. |
| 0 | | Data | | Reserved for manufacturer's test purposes. This bit should be set Low. |

**Address and Data format for TxErrStatMask access**

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |

| Data field [6:1] | | | | | | | |
|---|---|---|---|---|---|---|---|
| R | D6 | D5 | D4 | D3 | D2 | D1 | R |

### RxErrorStatus

Title: Receive Error Status register.
Address: $0x20
Function: R
Description: This register is the Rx Data path error status register. The *RxIrqActive* bit is set active when one of the other bits in this register is the source of an interrupt event. All these error conditions are caused by transitory events, therefore the error condition is latched (marked with an 'L'). Reading this status register causes all latched bits to be set inactive unless an error event is currently pending.
Setting any bit of this register High will cause an interrupt to be generated (N_IRQ will be set Low) if the source of the interrupt has not been masked in the corresponding Mask register.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | *RxDataPathQOF* | High | RL | Data path gain, phase and offset (GPO) adjustment unit: Q channel overflow error status bit. |
| 6 | *RxDataPathIOF* | High | RL | Data path gain, phase and offset (GPO) adjustment unit: I channel overflow error status bit. |
| 5 | *AdcQOF* | High | RL | ADC Q channel overflow error due to excessive input amplitude. |
| 4 | *AdcIOF* | High | RL | ADC I channel overflow error due to excessive input amplitude. |
| 3 | *RxLowPassOF* | High | RL | 63-tap Low-pass I and Q filter data accumulator overflow error status bit. |
| 2 | *RxRRCOF* | High | RL | 63-tap RRC I and Q filter data accumulator overflow error status bit. |
| 1 | *EvenSamplePhase* | High | RL | When this status bit is active, the associated interrupt may be used to re-synchronize the Rx data if for any reason data synchronization is lost. If the corresponding mask bit is set inactive, an interrupt will be generated on the next Q-phase data in the Rx output register. The next falling edge of SClk with RxFS High indicates the LSB of the Q channel data. The mask bit should be disabled after this to prevent continuous Q-phase interrupts. |
| 0 | *RxIrqActive* | High | RL | This bit is set High if there is an active interrupt caused by one of the status bits in this register. |

**Address and Data format for RxErrorStatus access**

| Address field [6:0] | | | | | | | Data field [7:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### RxErrorStatMask

Title: Receive Error Status interrupt Mask register.
Address: $0x21
Function: RW
Description: Masks interrupts in the **RxErrorStatus** Register. On taking N_RESET Low, these bits are set active, so masking out all possible interrupt sources. Each bit which is taken inactive will allow its associated status bit, when active, to generate an interrupt.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | *n_RxDataPathQOF_Mask* | Low | RW | GPO Q channel error interrupt mask bit. |
| 6 | *n_RxDataPathIOF_Mask* | Low | RW | GPO I channel error interrupt mask bit. |
| 5 | *n_AdcQOF_Mask* | Low | RW | ADC Q channel error interrupt mask bit. |
| 4 | *n_AdcIOF_Mask* | Low | RW | ADC I channel error interrupt mask bit. |
| 3 | *n_RxLowPassOF_Mask* | Low | RW | 63-tap Low-pass I and Q filter error interrupt mask bit. |
| 2 | *n_RxRRCOF_Mask* | Low | RW | 63-tap RRC I and Q filter error interrupt mask bit. |
| 1 | *EvenSamplePhase_Mask* | Low | RW | Rx data Q-phase interrupt mask bit. |
| 0 | | Data | RW | Reserved for manufacturer's test purposes. This bit should be set Low. |

**Address and Data format for RxStatMask access**

| Address field [6:0] | | | | | | | Data field [7:1] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | R |

### TxFIFOStatus

Title: Transmit data FIFO Status register
Address: $0x22
Function: R
Description: This register is the Tx Data FIFO status register. The *TxIrqActive* bit is set active when one of the other bits in this register is the source of an interrupt event. Some of these status conditions are caused by transitory events, therefore their state is latched (marked with an 'L'). The bits marked with a parenthesized 'L' are only latched in their interrupt generation state if their associated mask bit is inactive. Reading this status register causes all latched bits to be set inactive, unless an error event is currently pending.
Setting any bit of this register High will cause an interrupt to be generated (N_IRQ will be set Low) if the source of the interrupt has not been masked in the corresponding Mask register.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | *TxPathEn* | Low | R(L) | When High this bit shows that the Tx Data path is currently active, enabling the user to confirm whether ramp down has completed. For interrupt generation purposes, a logic Low on this bit will be considered as active. However, un-masking the interrupt on this bit will generally appear to invert the read status. This is because the read operation is switched from the active low source signal to an active high output latch that stores the last (uncleared) interrupt state. |
| 6 | *FIFOUnderRead* | High | RL | Error status bit. When active indicates a read from the FIFO occurred while the FIFO was empty. |
| 5 | *FIFOOverWrite* | High | RL | Error status bit. When active indicates a write to the FIFO occurred while the FIFO was full. |
| 4 | *FIFONotFull* | High | R(L) | Most significant FIFO length status bit. When High, this bit indicates the FIFO is not full. For interrupt generation purposes, a logic High on this bit will be considered as active. |
| 3 | *FIFONearlyFull* | High | R(L) | This interrupt is generated when the FIFO contains 3 used locations and one remains available |
| 2 | *FIFONearlyEmpty* | High | R(L) | This interrupt is generated when the FIFO contains one remaining data entry and 3 locations are free. |
| 1 | *FIFOEmpty* | High | R(L) | When active indicates the FIFO is empty. |
| 0 | *FifoIrqActive* | High | RL | This bit is set High if there is an active interrupt caused by one of the status bits in this register. |

**Address and Data format for TxFIFOStatus access**

| Address field [6:0] | | | | | | | Data field [7:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### TxFIFOStatMask

Title: Transmit data FIFO Status interrupt Mask register
Address: $0x23
Function: RW
Description: Masks interrupts in the TxFIFOStatus Register. On taking N_RESET Low, these bits are set active, so masking out all possible interrupt sources. Each inactive bit will allow its associated status bit to generate an interrupt. In the case of the status bits marked in the TxFIFOStatus Register with a parenthesized 'L', taking the mask bit inactive will enable the latching mechanism.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | *n_TxPathEn_Mask* | Low | RW | Tx Data path active interrupt mask bit. |
| 6 | *n_FIFOUnderRead_Mask* | Low | RW | FIFO underflow interrupt mask bit. |
| 5 | *n_FIFOOverWrite_Mask* | Low | RW | FIFO overflow interrupt mask bit. |
| 4 | *n_FIFOFull_Mask* | Low | RW | FIFO full interrupt mask bit. |
| 3 | *n_FIFONearlyFull_Mask* | Low | RW | FIFO nearly full interrupt mask bit. |
| 2 | *n_FIFONearlyEmpty_Mask* | Low | RW | FIFO nearly empty interrupt mask bit. |
| 1 | *n_FIFOEmpty_Mask* | Low | RW | FIFO empty interrupt mask bit. |
| 0 | | Data | RW | Reserved for manufacturer's test purposes. This bit should be set Low. |

**Address and Data format for TxFIFOStatMask access**

| Address field [6:0] | | | | | | | Data field [7:1] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | R |

### SymClkPhase

Title: Transmit Symbol Clock Phase adjustment register
Address: $0x24
Function: RW
Description: Allows phase adjustment of the internal symbol clock reference phase with respect to system time. The CMX980A symbol clock is a division of the Master clock frequency and is synchronized initially to the N_RESET signal. To allow correction or adjustment of this phase, without recourse to chip reset, a mechanism is provided via this register to advance or retard the phase of the symbol clock. The adjustment can be done in steps of ±1/4 or ±1/8 of a symbol time. The ±1/4 step complies with the TETRA requirement on phase adjustment with respect to the BS.

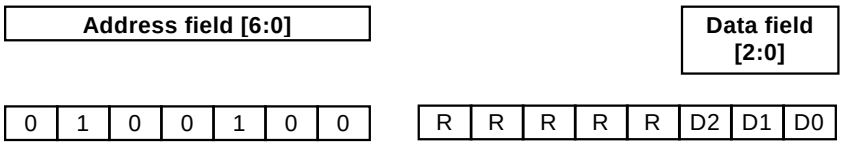
To ensure that the symbol clock phase is adjusted at the correct time and only in the step sizes allowed an interlock mechanism is used. The operation to alter the phase involves two write commands to the SymClkPhase Register in succession. The first can be considered as an “arm” write operation and the second as the execute. The execute write must occur before the next symbol clock as this signal cancels any “arm” state that is pending.

The simplest way is to time the “arm” write after the symbol clock strobe is detected on the N_IRQ pin, then followed immediately by the execute write.

If detection of the N_IRQ is not desired then three successive writes guarantee that the change will take place.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7:3 | | | RW | Reserved. These bits should be set Low. Undefined on read. |
| 2 | *SyncStep* | | RW | This determines the phase advance/retard step size.<br>1: ±1/4 symbol change<br>0: ±1/8symbol change |
| 1 | *SyncInc* | | RW | This determines whether the operation is an advance or retard step<br>1: Advance the phase by step size<br>0: Retard the phase by step size |
| 0 | *Enable_Change* | High | RW | The handshake and phase change enable bit. It is set by the user and cleared by the next internal symbol clock. |

**Address and Data format for SymClkPhase access**



### CoeffRamData

Title: I/O access addresses for the four user-accessible coefficient memories.
Address: $0x26 to $0x2D (mapped over 8 locations)
Function: RW
Description: Each coefficient RAM has both MSB and LSB address ports assigned for read/write access. There are two transmit (Tx) FIR filters with read/write coefficients and two receive (Rx) filters, with coefficient sizes of 12 and 16 bits respectively. Access to the coefficient memory is valid only when the CoeffRamIoEn bit is active.
Asserting the CoeffRamIoEn will reset the Coefficient Address Pointer to the first location (A1). The MSB port should be accessed first, as accessing the LSB port will move the Coefficient Address Pointer to the next coefficient location (A[n+1]) (refer to description of CoeffRamIoRdInc bit for details). Subsequent accesses to the LSB port of the coefficient address will increment the Coefficient Address Pointer.
In the 79-tap Tx filter the coefficients are symmetrical and "odd" and only 40 locations can be programmed. Performing an I/O access after the last Coefficient Address Pointer (A[41-79] ) is not valid, and may corrupt existing coefficients.
All other filters have access to coefficients A1 to A[FirLength], thus the user can chose to program symmetrical or non-symmetrical filter responses. In either type of filter access to the coefficient location A(FirLength+1) should be avoided as this location must contain zero for correct filter operation.
Only one FIR filter coefficient RAM may be accessed at a time. If further filter coefficient RAMs are to be accessed then the CoeffRamIoEn must first be deactivated, and then activated again, allowing the next FIR filter coefficient RAM to be incrementally accessed.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x26*** | | | | |
| 7:0 | *TxRRCCoeffLSB* | Data | RW | Transmit 63-tap RRC filter LSB coefficient data port<br>Post-increment the coefficient address pointer. |
| ***Address $0x27*** | | | | |
| 7:4 | | | RW | Reserved. Set these bits High. Undefined on read. |
| 3:0 | *TxRRCCoeffMSB* | Data | RW | Transmit 63-tap RRC filter MSB coefficient data port. |
| ***Address $0x28*** | | | | |
| 7:0 | *Tx79tapCoeffLSB* | Data | RW | Transmit 79-tap filter LSB coefficient data port.<br>Post-increment the coefficient address pointer. |
| ***Address $0x29*** | | | | |
| 7:4 | | | RW | Reserved. Set these bits High. Undefined on read. |
| 3:0 | *Tx79tapCoeffMSB* | Data | RW | Transmit 79-tap filter MSB coefficient data port. |
| ***Address $0x2A*** | | | | |
| 7:0 | *RxRRCCoeffLSB* | Data | RW | Receive 63-tap RRC filter LSB coefficient data port.<br>Post-increment the coefficient address pointer. |
| ***Address $0x2B*** | | | | |
| 7:0 | RxRRCCoeffMSB | Data | RW | Receive 63-tap RRC filter LSB coefficient data port. |
| | | | | |
| ***Address $0x2C*** | | | | |
| 7:0 | *RxLowPsCoeffLSB* | Data | RW | Receive 63-tap Low Pass filter LSB coefficient data port.<br>Post-increment the coefficient address pointer. |
| ***Address $0x2D*** | | | | |
| 7:0 | *RxLowPsCoeffMSB* | Data | RW | Receive 63-tap Low Pass filter MSB coefficient data port. |

**Address and Data format for 63-tap Tx RRC FIR Coefficient Ram IO access**

| Address field [6:1] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 1 | A0 |
| (Coefficient Pointer)++ | | | | | | 0 |
| Coefficient Pointer | | | | | | 1 |

| Coefficient Data field [11:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| R | R | R | R | D11 | D10 | D9 | D8 |

**Address and Data format for 79-tap Tx FIR Coefficient Ram IO access**

| Address field [6:1] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 0 | A0 |
| (Coefficient Pointer)++ | | | | | | 0 |
| Coefficient Pointer | | | | | | 1 |

| Coefficient Data field [11:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| R | R | R | R | D11 | D10 | D9 | D8 |

**Address and Data format for 63-tap Rx RRC FIR Coefficient Ram IO access**

| Address field [6:1] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | A0 |
| (Coefficient Pointer)++ | | | | | | 0 |
| Coefficient Pointer | | | | | | 1 |

| Coefficient Data field [15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |

**Address and Data format for 63-tap Rx Low Pass FIR Coefficient Ram IO access**

| Address field [6:1] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | 0 | A0 |
| (Coefficient Pointer)++ | | | | | | 0 |
| Coefficient Pointer | | | | | | 1 |

| Coefficient Data field [15:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 |

### ClkStopCtrl

Title: Clock-Stop Control register
Address: $0x3C
Function: RW
Description: Control of power down and clock operation.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7:4 | | | RW | Reserved. These bits should be set Low. Undefined on read. |
| 3 | *RxIFClkStopMode* | High | RW | When set active, this bit puts the Rx data serial-interface logic into clock-stop mode. The interface will cease activity and enter a power down state. It will remain in this state until the user disables this bit. |
| 2 | *Aux_ClkStopMode* | High | RW | When set active, this bit puts the auxiliary ADC and RamDac logic into clock-stop mode. This reduces power within this section when auxiliary functions are not in use. |
| 1 | *AutoClkStopMode* | High | RW | When set active with the serial port configured at low data rate (*DataRateHi* bit of ConfigCtrl1 Register set inactive (Low)), this bit puts the serial interface logic into auto stop mode. The interface will cease activity and enter a power down state if no CmdFS activity is detected for 4096 master clock cycles (444µs with a 9.216MHz MCLK). To re-start serial interface operations, the user asserts the CmdFS strobe for at least one MCLK cycle. |
| 0 | *RxClkStop* | High | RW | When set active causes the *RxEn* bit of RxSetup1 Register to gate the Rx Data path master clock. When inactive (default state) the Rx Data path master clock is always supplied. |

**Address and Data format for ClkStopCtrl access**

| Address field [6:0] | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |

| Coefficient Data field [3:0] | | | | | | | |
|---|---|---|---|---|---|---|---|
| R | R | R | R | D3 | D2 | D1 | D0 |

### SramData

Title: I/O access address for the auxiliary DAC1 memories.
Address: $0x70 to $0x73 (mapped over 4 locations)
Function: RW
Description: These four address locations allow access to the 64 x 10 bit SRAM. The contents of this RAM can be pre-loaded with a table of values which can be automatically sent to auxiliary DAC1 in either a single cycle or continuous mode, see RamDacCtrl for details. Therefore, the RAM can be used in conjunction with DAC1 to enable user defined profile power ramping of an external RF power transmitter stage.
The RAM contents are addressed incrementally by first taking the SRamIoEn bit active. While this bit is inactive the SRam Address Pointer is held reset. The physical address applied to the RAM is formed from the 4-bit SRam Address Pointer and the two LSB bits from the I/O Access address (A1,A0). Therefore, four locations in the RAM can be accessed by directly addressing $0x70 to $0x73. However, accessing location $0x73, post-increments (by a block of four addresses) the SRam Address Pointer, thus moving the pointer to the next RAM location block.
The 10-bit data word is split between “odd” and “even” locations with the MSB byte in “odd” addresses (A0 = 1) and 2 LSB’s in “even” addresses.
The SRamIoRdInc bit determines whether a read or a write operation will increment the SRam Address Pointer. All 16 locations are accessed incrementally; further accesses to this port while the SRamIoEn bit is active are not valid and may cause data loss.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x70*** | | | | |
| 7:2 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 1:0 | *SRamLSBPort0* | Data | RW | Access port for the LSB register. |
| ***Address $0x71*** | | | | |
| 7:0 | *SRamMSBPort0* | Data | RW | Access port for the MSB register |
| ***Address $0x72*** | | | | |
| 7:2 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 1:0 | *SRamLSBPort1* | Data | RW | Access port for the LSB register. |
| ***Address $0x73*** | | | | |
| 7:0 | *SRamMSBPort3* | Data | RW | Access port for the MSB register.<br>Post-increment *Sram* address pointer. |

**Address and Data format for Sram Data I/O access**

| Address field [6:0] | | | | | | | Data field [9:0] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | A1 | A0 | | | | | | | | | | |
| Sram Address Pointer | | | | | 0 | 0 | | | R | R | R | R | R | R | D1 | D0 |
| Sram Address Pointer | | | | | 0 | 1 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | | |
| Sram Address Pointer | | | | | 1 | 0 | | | R | R | R | R | R | R | D1 | D0 |
| (Sram Address Pointer)++ | | | | | 1 | 1 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | | |

### TxRampUpInc

Title: Transmit Ramp Up Increment registers.
Address: $0x4C to $0x4D (mapped over 2 locations)
Function: RW
Description: The value in this register sets the scale of the Tx amplitude gain increments which occur over each sample clock period, thus determining the Tx amplitude ramp up time period. The value is always positive. The ramp up time, in terms of the number of symbols, is given by the formula:

$$N_{symbols} = \frac{64}{N_{inc}}$$

Where: $N_{symbols}$ is the ramp time in terms of number of symbols.

$N_{inc}$ is the value in the register.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x4C*** | | | | |
| 7:0 | *RampUpIncLSB* | Data | RW | Least significant 8 bits of the ramp up increment register. |
| ***Address $0x4D*** | | | | |
| 7:1 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 0 | *RampUpIncMSB* | Data | RW | Most significant bit of the ramp up increment register. |

**Address and Data format for TxRampUpInc access**

| Address field [6:0] | | | | | | | Data field [8:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | | | |
| | | | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | R | R | R | R | R | R | R | D8 |

### TxRampDnDec

Title: Transmit Ramp Down Decrement registers.
Address: $0x4E to $0x4F (mapped over 2 locations)
Function: RW
Description: The value in this register sets the scale of the Tx amplitude gain decrements that occur over each sample clock period, thus determining the Tx amplitude ramp down time period. The value is always positive. The ramp down time, in terms of the number of symbols, is given by the formula:

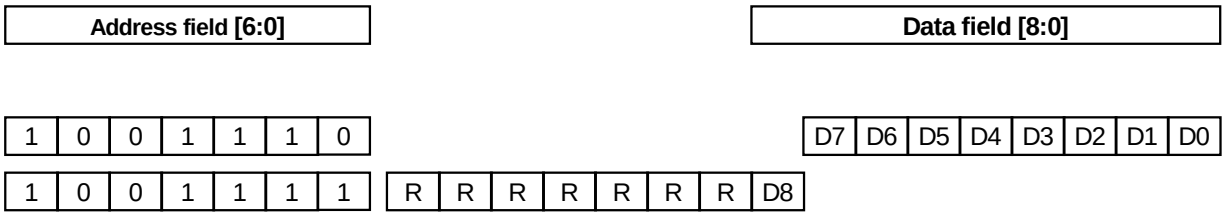
$$N_{symbols} = \frac{64}{N_{inc}}$$

Where: $N_{symbols}$ is the ramp time in terms of number of symbols.

$N_{inc}$ is the value in the register.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x4E*** | | | | |
| 7:0 | *RampDnIncLSB* | Data | RW | Least significant 8 bits of the ramp down decrement register. |
| ***Address $0x4F*** | | | | |
| 7:1 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 0 | *RampDnIncMSB* | Data | RW | Most significant bit of the ramp down decrement register. |

**Address and Data format for TxRampDnDec access**



**TxIQGainMult**

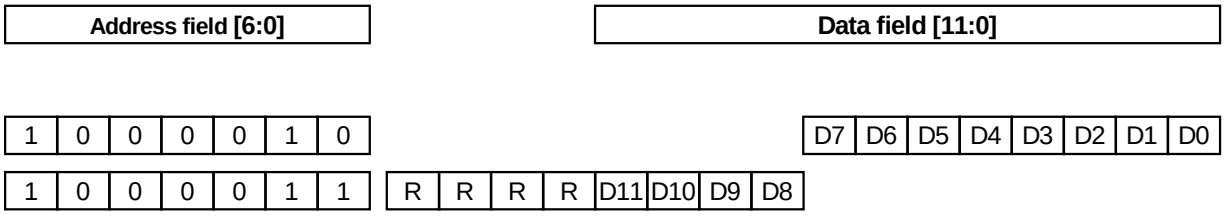
Title: Transmit I and Q channel Gain Multiplier registers
Address: $0x42, $0x43 , $0x48 and $0x49 (4 locations)
Function: RW
Description: A 2s-complement multiplication is performed on the Tx I and Q data signals, using these 12-bit 2s-complement words as the respective multiplicands. This function may be considered as a digital attenuator. This register sets the multiplier, the result being given by the formula:

$$D_{out} = D_{in}\left[\frac{G_{val}}{2^{11}}\right]$$

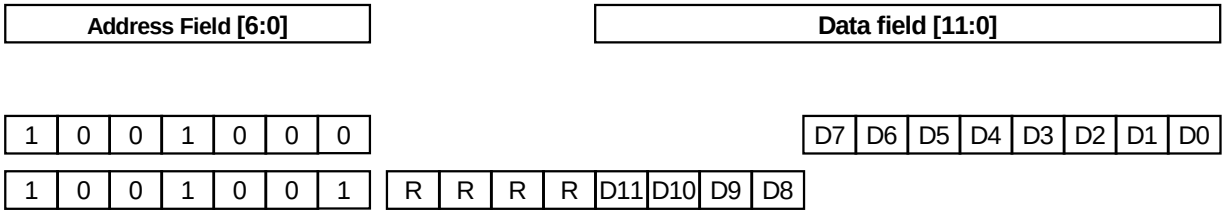
Where: $D_{in}$ is the signal input,
$D_{out}$ is the signal output,
$G_{val}$ is the value in the register.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x42*** | | | | |
| 7:0 | *TxIGainLSB* | Data | RW | Least significant 8 bits of the TxIGain register ($G_{val}$). |
| ***Address $0x43*** | | | | |
| 7:4 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 3:0 | *TxIGainMSB* | Data | RW | Most significant 4 bits of the TxIGain register ($G_{val}$). |
| ***Address $0x48*** | | | | |
| 7:0 | *TxQGainLSB* | Data | RW | Least significant 8 bits of the TxQGain register ($G_{val}$). |
| ***Address $0x49*** | | | | |
| 7:4 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 3:0 | *TxQGainMSB* | Data | RW | Most significant 4 bits of the TxQGain register ($G_{val}$). |

**Address and Data format for TxIGain access**




**Address and Data format for TxQGain access**



### TxIQOffset

Title: Transmit I and Q channel Offset correction register
Address: $0x44, $0x45, $0x4A, and $0x4B (4 locations)
Function: RW
Description: This register controls the Tx Data path signal offset. This offset is a 2s-complement value (Noffset), which is applied to the Tx signal after the Gain Multiplier (Gval), but before the DAC. The offset applied is at the discretion of the user. Inappropriate values may cause arithmetic overflow in the subsequent operator sections. The result is given by the formula:

$$D_{out} = D_{in} + \left[ \frac{N_{offset}}{2^{11}} \right]$$

Where: $D_{in}$ is the signal input,
$D_{out}$ is the signal output,
$N_{offset}$ is the signed 2s-complement value in the register.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x44*** | | | | |
| 7:0 | *TxIOffsetLSB* | Data | RW | Least significant 8 bits of the TxIOffset register ($N_{OFFSET}$). |
| ***Address $0x45*** | | | | |
| 7:4 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 3:0 | *TxIOffsetMSB* | Data | RW | Most significant 4 bits of the TxIOffset register ($N_{OFFSET}$). |
| ***Address $0x4A*** | | | | |
| 7:0 | *TxQOffsetLSB* | Data | RW | Least significant 8 bits of the TxQOffset register ($N_{OFFSET}$). |
| ***Address $0x4B*** | | | | |
| 7:4 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 3:0 | *TxQOffsetMSB* | Data | RW | Most significant 4 bits of the TxQOffset register ($N_{OFFSET}$). |

**Address and Data format for TxIOffset access**

| Address field [6:0] | | | | | | | Data field [11:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | R | R | R | R | D11 | D10 | D9 | D8 | | | | |

**Address and Data format for TxQOffset access**

| Address field [6:0] | | | | | | | Data field [11:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | R | R | R | R | D11 | D10 | D9 | D8 | | | | |

### TxPhase

Title: Transmit I and Q channel Phase correction register
Address: $0x40, $0x41, $0x46, $0x47 (4 locations)
Function: RW
Description: This register controls the Tx Data path I and Q channel phase compensation. The phase may be adjusted by ±7.1° with respect to the input data signal phase. As each channel has separate phase adjustments, the maximum differential phase compensation that can be achieved is ±14.2°. The phase adjustment value written to this register is a 2s-complement value ($N_{phase}$).
The amount of phase adjustment applied is given by the formula:

$$\phi = tan^{-1}\left[\frac{N_{phase}}{2^{11}}\right]$$

Where: $\phi$ is the phase adjustment,
$N_{phase}$ is the value in the register and has a range of -256 to +255.

**Note:** Although each channel is separately adjustable with its own compensation value, the effect of phase adjustment is only detectable by measuring the phase angle between I and Q channels. It should be noted that the $N_{phase}$ value has the effect of lagging the I channel for positive values of $N_{phase}$ (conversely, leading the phase for negative values) and leading the Q channel for positive values of $N_{phase}$ (conversely, lagging the phase for negative values). For example, putting the value 10 (decimal) into both TxIPhase and TxQPhase would produce a differential phase on I and Q of:

$$90^o - 2(\tan^{-1}(4.88 \times 10^{-3})) = 89.44^o$$

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x40*** | | | | |
| 7:0 | *TxIPhaseLSB* | Data | RW | Least significant 8 bits of the TxIPhase register ($N_{phase}$). |
| ***Address $0x41*** | | | | |
| 7:1 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 0 | *TxIPhaseMSB* | Data | RW | Most significant bit of the TxIPhase register (sign bit). |
| ***Address $0x46*** | | | | |
| 7:0 | *TxQPhaseLSB* | Data | RW | Least significant 8 bits of the TxQPhase register ($N_{phase}$). |
| ***Address $0x47*** | | | | |
| 7:1 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 0 | *TxQPhaseMSB* | Data | RW | Most significant bit of the TxQPhase register (sign bit). |

## Address and Data format for TxIPhase access

| Address field [6:0] | | | | | | | Data field [8:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | R | R | R | R | R | R | R | D8 |

## Address and Data format for TxQPhase access

| Address field [6:0] | | | | | | | Data field [8:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | R | R | R | R | R | R | R | D8 |

### TxDataAccess

Title: Tx Data path Access point.
Address: $0x50 to $0x53 (mapped over 4 locations)
Function: RW
Description: This register block allows direct access to the Tx Data path values just after the gain, phase and offset adjustment block. Both read and write operations are permitted. A read operation reads the signal values on the I and Q channels. A write operation will write data directly to the Sigma-Delta DAC input. To prevent normal Tx data overwriting this value the TxDPAccessSel bit in the LoopBackCtrl Register should be set active. The MSB read data register is buffered to enable access to a discrete sample value (if this register was not buffered, data from different sample periods could be in the MSB and LSB registers). Therefore the LSB register must be read first for correct operation.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x50*** | | | | |
| 7:0 | *TxDPIDataLSB* | Data | RW | Least significant 8 bits of the TxDPIData register. This register must be read before its associated MSB register. |
| ***Address $0x51*** | | | | |
| 7:6 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 5:0 | *TxDPIDataMSB* | Data | RW | Most significant 6 bits of the TxDPIData register. |
| ***Address $0x52*** | | | | |
| 7:0 | *TxDPQDataLSB* | Data | RW | Least significant 8 bits of the TxDPQData register. This register must be read before its associated MSB register. |
| ***Address $0x53*** | | | | |
| 7:6 | | | RW | Reserved. Set these bits Low. Undefined on read. |
| 5:0 | *TxDPQDataMSB* | Data | RW | Most significant 6 bits of the TxDPQData register. |

**Address and Data format for TxDPIData access**

| Address field [6:0] | | | | | | | Data field [13:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | | | |
| | | | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | R | R | D13 | D12 | D11 | D10 | D9 | D8 |

**Address and Data format for TxDPQData access**

| Address field [6:0] | | | | | | | Data field [13:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | | | | | | | | |
| | | | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | R | R | D13 | D12 | D11 | D10 | D9 | D8 |

### RxIQGainMult

Title: Receive I and Q channel Gain Multiplier register
Address: $0x30, $0x31, $0x34 and $0x35 (4 locations)
Function: RW
Description: A 2s-complement multiplication is performed on the magnitude of the Rx Data path signal and the result is then re-normalised to the system's dynamic range: thus the function may be considered as a digital attenuator. The value is signed 2s-complement therefore phase inversion of the channel data can be achieved by using negative numbers. This multiplication is applied to the Rx signal after the ADC decimation filter, but before offset adjustment and the two 63-tap FIR filters. This register sets the multiplier, the result being given by the formula:

$$D_{out} = D_{in}\left[\frac{G_{val}}{2^{15}}\right]$$

Where: $D_{in}$ is the signal input,
$D_{out}$ is the signal output,
$G_{val}$ is the value in the register.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x30*** | | | | |
| 7:0 | *RxIGainLSB* | Data | RW | Least significant 8 bits of the RxIGain register ($G_{VAL}$). |
| ***Address $0x31*** | | | | |
| 7:0 | *RxIGainMSB* | Data | RW | Most significant 8 bits of the RxIGain register ($G_{VAL}$). |
| ***Address $0x34*** | | | | |
| 7:0 | *RxQGainLSB* | Data | RW | Least significant 8 bits of the RxQGain register ($G_{VAL}$). |
| ***Address $0x35*** | | | | |
| 7:0 | *RxQGainMSB* | Data | RW | Most significant 8 bits of the RxQGain register ($G_{VAL}$). |

**Address and Data format for RxIGain access**

| Address field [6:0] | Data field [15:0] |
|---|---|
| 0 1 1 0 0 0 0 | D7 D6 D5 D4 D3 D2 D1 D0 |
| 0 1 1 0 0 0 1 | D15 D14 D13 D12 D11 D10 D9 D8 |

**Address and Data format for RxQGain access**

| Address field [6:0] | Data field [15:0] |
|---|---|
| 0 1 1 0 1 0 0 | D7 D6 D5 D4 D3 D2 D1 D0 |
| 0 1 1 0 1 0 1 | D15 D14 D13 D12 D11 D10 D9 D8 |

### RxIQOffset

Title: Receive I and Q Channel Offset correction register
Address: $0x32, $0x33, $0x36, and $0x37 (4 locations)
Function: RW
Description: This register controls the Rx Data path signal offset. This offset is a 2s-complement value ($N_{OFFSET}$), which is applied to the Rx signal after the Gain Multiplier ($G_{VAL}$), but before the two 63-tap FIR filters. The offset applied is at the discretion of the user. Inappropriate values may cause arithmetic overflow in the subsequent operator sections. The result is given by the formula:

$$D_{out} = D_{in} + \left[ \frac{N_{offset}}{2^{15}} \right]$$

Where: $D_{in}$ is the signal input,
$D_{out}$ is the signal output,
$N_{offset}$ is the signed 2s-complement value in the register.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x32*** | | | | |
| 7:0 | *RxIOffsetLSB* | Data | RW | Least significant 8 bits of the RxIOffset register ($N_{OFFSET}$). |
| ***Address $0x33*** | | | | |
| 7:0 | RxIOffsetMSB | Data | RW | Most significant 8 bits of the RxIOffset register ($N_{OFFSET}$). |
| ***Address $0x36*** | | | | |
| 7:0 | RxQOffsetLSB | Data | RW | Least significant 8 bits of the RxQOffset register ($N_{OFFSET}$). |
| ***Address $0x37*** | | | | |
| 7:0 | RxQOffsetMSB | Data | RW | Most significant 8 bits of the RxQOffset register ($N_{OFFSET}$). |

**Address and Data format for RxIOffset access**

| Address field [6:0] | Data field [15:0] |
|---|---|
| 0 1 1 0 0 1 0 | D7 D6 D5 D4 D3 D2 D1 D0 |
| 0 1 1 0 0 1 1 | D15 D14 D13 D12 D11 D10 D9 D8 |

**Address and Data format for RxQOffset access**

| Address field [6:0] | Data field [15:0] |
|---|---|
| 0 1 1 0 1 1 0 | D7 D6 D5 D4 D3 D2 D1 D0 |
| 0 1 1 0 1 1 1 | D15 D14 D13 D12 D11 D10 D9 D8 |

### RxDataAccess

Title: Rx Data path Access point.
Address: $0x38 to $0x3B (mapped over 4 locations)
Function: RW
Description: This register block allows direct access to the Rx Data path values just after the Rx gain and offset adjustment block. Both read and write operations are permitted. A read operation reads the signal values on the I and Q channels. A write operation will write data to the Rx Data path operator output. To prevent normal Rx data overwriting this value the RxDPAccessSel bit in the LoopBackCtrl Register should be set active. The MSB read data register is buffered to enable access of a discrete sample value (if this register was not buffered, data from different sample periods could be in the MSB and LSB registers). Therefore the LSB register must be read first for correct operation.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x38*** | | | | |
| 7:0 | *RxDPIDataLSB* | Data | RW | Least significant 8 bits of the RxDPIData register. This register must be read before its associated MSB register. |
| ***Address $0x39*** | | | | |
| 7:0 | *RxDPIDataMSB* | Data | RW | Most significant 8 bits of the RxDPIData register. |
| ***Address $0x3A*** | | | | |
| 7:0 | *RxDPQDataLSB* | Data | RW | Least significant 8 bits of the RxDPQData register. This register must be read before its associated MSB register. |
| ***Address $0x3B*** | | | | |
| 7:0 | *RxDPQDataMSB* | Data | RW | Most significant 8 bits of the RxDPQData register. |

**Address and Data format for RxDPIData access**

| Address field [6:0] | Data field [15:0] |
|---|---|
| 0 1 1 1 0 0 0 | D7 D6 D5 D4 D3 D2 D1 D0 |
| 0 1 1 1 0 0 1 | D15 D14 D13 D12 D11 D10 D9 D8 |

**Address and Data format for RxDPQData access**

| Address field [6:0] | Data field [15:0] |
|---|---|
| 0 1 1 1 0 1 0 | D7 D6 D5 D4 D3 D2 D1 D0 |
| 0 1 1 1 0 1 1 | D15 D14 D13 D12 D11 D10 D9 D8 |

### BISTControl

Title: Built In Self Test Control register
Address: $0x62
Function: RW
Description: This register block allows control of BIST operations.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| 7 | *TestCompleteAck* | High/Low | RW | This bit is set by the user and cleared by the BIST controller when a BIST cycle has been completed. |
| 6 | *n_RampDelayEn* | Low | RW | Allow Ramp control signal delay. This delay is required for normal operations, by matching the FIR filter delays. For BIST operations it can be disabled thus reducing BIST test time. |
| 5 | *BISTDataRateHi* | High | RW | Selects BIST data rate = MCLK/64<br>Default rate (Low) = MCLK/4 |
| 4 | *BISTEn* | High | RW | Enables BIST operations and starts BIST master clock. |
| 3 | *ContinuousBIST* | High | RW | Selects continuous BIST mode.<br>Default (Low) selects single cycle mode. |
| 2 | *EnRxDigitalFeedBack* | High | RW | Selects Rx digital loop feedback for 63-tap Tx RRC FIR input data. Default (Low) selects normal Tx data. |
| 1 | *En63tlQData* | High | RW | Selects BIST data for 63-tap Tx FIR filter input.<br>Default (Low) selects normal data. |
| 0 | *EnSymTestData* | High | RW | Selects BIST data for 79-tap FIR filter input.<br>Default (Low) selects normal data. |

**Address and Data format for BistControl access**

| Address field [6:0] | | | | | | | Data field [7:0] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

## BISTPRSG

Title: Built In Self Test Pseudo Random Sequence Generator
Address: $0x60 to $0x61 (2 locations)
Function: RW
Description: This register block allows control of BIST operations. This 16-bit number controls the length of the BIST data sequence. It is the initial value (or seed) written to the pseudo-random sequence generation logic. The length of the BIST data sequence is a function of the feedback logic equation and this initial value. The feedback function is fixed so run lengths are therefore controlled by this value.
Which values to apply to give specific run lengths can be determined from a look-up table. This table may be provided on request.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x60*** | | | | |
| 7:0 | *BISTPRSGLSB* | Data | RW | Least significant 8 bits of the BISTPRSG register. This register must be read before its associated MSB register. |
| ***Address $0x61*** | | | | |
| 7:0 | *BISTPRSGMSB* | Data | RW | Most significant 8 bits of the BISTPRSG register. This register must be read after its associated LSB register. |

**Address and Data format for BISTPRSG access**

| Address field [6:0] | Data field [15:0] |
|---|---|
| 1 1 0 0 0 0 0 | D7 D6 D5 D4 D3 D2 D1 D0 |
| 1 1 0 0 0 0 1 | D15 D14 D13 D12 D11 D10 D9 D8 |

### BISTCRCRegisters

Title: Built In Self Test Cyclic Redundancy Code checking Registers
Address: $0x64 to $0x6D (10 locations)
Function: R
Description: This register block allows BIST CRC checksums to be read.

| Bit | Name | Active | State | Function |
|---|---|---|---|---|
| ***Address $0x64*** | | | | |
| 7:0 | *79tapI_CRCLSB* | Data | R | Transmit I channel 79-tap filter LSB register. |
| ***Address $0x65*** | | | | |
| 7:0 | *79tapI_CRCMSB* | Data | R | Transmit I channel 79-tap filter MSB register. |
| ***Address $0x66*** | | | | |
| 7:0 | *79tapQ_CRCLSB* | Data | R | Transmit Q channel 79-tap filter LSB register. |
| ***Address $0x67*** | | | | |
| 7:0 | *79tapQ_CRCMSB* | Data | R | Transmit Q channel 79-tap filter MSB register |
| ***Address $0x68*** | | | | |
| 7:0 | *SDM_CRCLSB* | Data | R | Transmit SDM DAC LSB register. |
| ***Address $0x69*** | | | | |
| 7:0 | *SDM_CRCMSB* | Data | R | Transmit SDM DAC MSB register. |
| ***Address $0x6A*** | | | | |
| 7:0 | *RXI_CRCLSB* | Data | R | Receive I channel LSB register. |
| ***Address $0x6B*** | | | | |
| 7:0 | *RXQ_CRCLSB* | Data | R | Receive I channel MSB register. |
| ***Address $0x6C*** | | | | |
| 7:0 | *RXQ_CRCLSB* | Data | R | Receive Q channel LSB register. |
| ***Address $0x6D*** | | | | |
| 7:0 | *RXQ_CRCMSB* | Data | R | Receive Q channel MSB register. |

**Address and Data format for 79-tap I channel CRC register access**

| Address field [6:0] | Data field [15:0] |
|---|---|

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | | | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | | | | | | |

**Address and Data format for 79-tap Q channel CRC register access**

| Address field [6:0] | Data field [15:0] |
|---|---|

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | | | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | | | | | | |

**Address and Data format for SDM CRC register access**

| Address field [6:0] | Data field [15:0] |
|---|---|

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | | | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | | | | | | |

**Address and Data format for Rx I Channel CRC register access**

| Address field [6:0] | Data field [15:0] |
|---|---|

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | | | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | | | | | | |

**Address and Data format for Rx Q Channel CRC register access**

| Address field [6:0] | Data field [15:0] |
|---|---|

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | | | | | | | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | | | | | | |

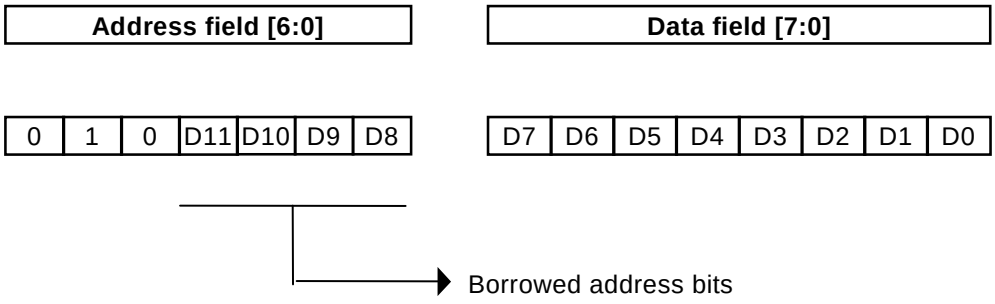
 

#### 4.8.3.3 PAGE 1 ADDRESSED REGISTERS

**DirectWrite79tapI**

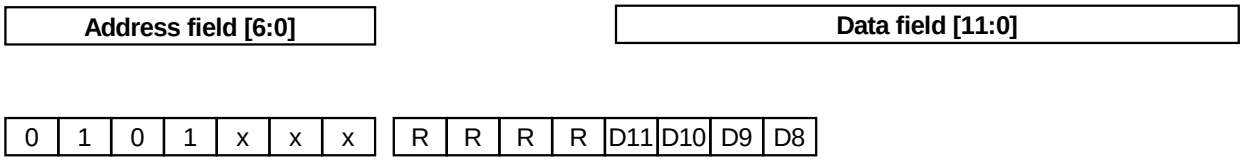
Title: Direct write access to 79-tap I channel filter
Address: $0x20 - $0x2F (16 locations)
Function: RW
Description: These registers are the direct access points to the 79-tap filter I channel. Writing accesses the holding latch whose value is transferred to the filter input at the next internal sample clock enable time. Read the address returns the value of the holding latch.
To allow the data latch value into the filter the TxDirectWriteEn bit must be set in the TxSetup Register otherwise the input data for the filter is taken from the DQPSK modulator output.
These registers can be read via two operations, giving the most significant nibble and least significant byte respectively. In the read case, the D11 bit becomes an address bit, which is used to select between the most significant nibble (D11 = '1') and the least significant byte (D11 = '0'). The D10, D9 and D8 bits are ignored.
The format for write data is to apply the most significant 4 bits of the data value to the address field [3:0] and the least significant 8 bits to the data field [7:0]. In this way a 12-bit value can be written in one serial operation.

**Address and Data format for 79-tap filter I channel Direct Write "Write" access**




**Address and Data format for 79-tap I channel filter Direct Write "Read" access LSB data access**


**Address and Data format for 79-tap I channel filter Direct Write "Read" access MSB data access**

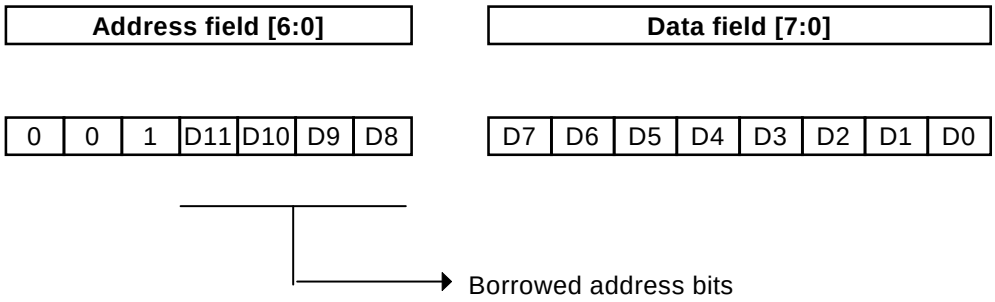


### DirectWrite79tapQ

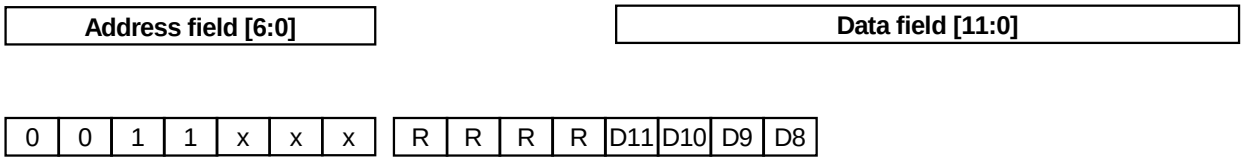
Title: Direct write access to 79 tap Q channel filter
Address: $0x10 - $0x1F (16 locations)
Function: RW
Description: These registers are the direct access points to the 79-tap filter Q channel. Writing accesses the holding latch whose value is transferred to the filter input at the next internal sample clock enable time. Read the address returns the value of the holding latch.
To allow the data latch value into the filter the TxDirectWriteEn bit must be set in the TxSetup Register otherwise the input data for the filter is taken from the DQPSK modulator output.
These registers can be read via two operations, giving the most significant nibble and least significant byte respectively. In the read case, the D11 bit becomes an address bit, which is used to select between the most significant nibble (D11 = '1') and the least significant byte (D11 = '0'). The D10, D9 and D8 bits are ignored.
The format for write data is to apply the most significant 4 bits of the data value to the address field [3:0] and the least significant 8 bits to the data field [7:0]. In this way a 12-bit value can be written in one serial operation.

**Address and Data format for 79-tap filter Q channel Direct Write "Write" access**




**Address and Data format for 79-tap Q channel filter Direct Write "Read" access LSB data access**


**Address and Data format for 79-tap Q channel filter Direct Write "Read" access MSB data access**



# 5 Application Notes

## 5.1 Interrupt Handling

Interrupt handling requires an extra read to clear the source of the interrupt. Handling interrupts is sometimes a source of confusion. The notes below are intended to clarify the operation of interrupts:

### 5.1.1 Tx FIFO status interrupts

These interrupts can only be cleared by first carrying out the appropriate action to stop the source of the Tx FIFO interrupt (this would usually require writing some data to the Tx FIFO) and then carrying out a further read on the **TxFIFOStatus** Register ($0x22) to reset the N_IRQ pin.

### 5.1.2 Tx/Rx FIR filter tap overflow and Gain, Phase and Offset overflow interrupts

A typical interrupt handling procedure for Tx (the same can be applied to Rx) would be:

- Read **TxErrorStatus** Register ($0x0E) and confirm that a Tx FIR filter error has occurred.
- This will reset the N_IRQ pin.

### 5.1.3 Rx ADC I and Q channel overflow - due to excessive input amplitude interrupts

These interrupts will remain set until the source of the excessive amplitude has been reduced to below the acceptable level. Once this has been achieved, the RxErrorStatus Register can be read in order to reset the N_IRQ pin.

Note: Never enable these interrupts with the Rx path disabled, as this will continuously generate an interrupt.

## 5.2 Configuration

Configuration registers **ConfigCtrl1** and **ConfigCtrl2** are not double buffered and so should not be altered during Tx.

## 5.3 Reset

The N_RESET pin should be held active (Low) during power-up. The N_RESET pin requires two complete MCLK clock cycles whilst active in order to take effect.

## 5.4 Developing and Optimizing FIR Filter Coefficients

If it is required to re-optimize FIR filter coefficients for a different application, or to compensate for the behavior of components external to the CMX980A, the default coefficients can be overwritten. There are many ways to develop FIR filter coefficients for a non-TETRA application.

The basic algorithm is to take the required frequency domain response, apply an inverse Fourier transform and use a windowing function to reduce the impulse response to the desired length. The impulse response is then identical to the required FIR coefficients. In the case of the CMX980A, both transmit and receive filters are configured as two cascaded filters. When developing customized coefficients, the user has a choice of whether to design the two filters separately or to develop a single filter and then factorize the resultant polynomial in Z (representing the impulse response of the overall FIR filter) into two shorter polynomials of appropriate length. Various commercial and public domain software is available which may help with this process.

In order to develop optimal FIR filter coefficients for the CMX980A, knowledge of the non-programmable filters in the design is required, together with a more detailed understanding of the function of certain external components. Please refer to the block diagram in Figure 1 and the external component diagrams in Figure 2 and Figure 3.

The combined effect of all of the filters in the Tx or Rx, when using default FIR coefficients, is to give a linear phase root raised cosine filter shape, with a symbol rate of MCLK/512 and
$\alpha = 0.35$. This tracks fairly well with MCLK frequency, provided that the dominant external RC poles (R3/C3 for Tx, R2/C2 for Rx, as shown in Figure 2 and Figure 3) are also scaled with MCLK. For the case of MCLK = 8.192MHz, this means increasing the RC products by approximately 10%.

There is a small attenuation caused by two pole on-chip continuous time filters in both the Tx and Rx, which do not scale with MCLK. This will cause attenuation at 10kHz of between 0.05dB and 0.15dB in the Rx (this can be bypassed), and between 0.03dB and 0.08dB in the Tx. This effect can be ignored in many applications, but is described here for completeness.

### 5.4.1 Tx Path Details

Data can either be input via the DQPSK modulator or the direct write port at a symbol rate of MCLK/512. Due to the zero padding of the data from symbol rate to 8x sample rate, the ratio of symbol rate to MCLK rate is fixed.

I and Q Data is passed through the following elements:

A. a pair of programmable FIR filters (79-tap and 63-tap)

B. a gain/phase/offset adjustment block

C. matched pair of sigma delta DACs and a switched capacitor data reconstruction filter (which requires one external RC pole).

D. a two pole continuous time active filter which suppresses clock noise. This has a fixed pole frequency of 140kHz (subject to ± 40% tolerance for process variations). It has no significant effect on the passband for all allowable MCLK frequencies, but the level of clock noise suppression supplied by this filter (48dB for MCLK = 9.216MHz) is reduced at lower MCLK frequencies.

The reconstruction filter has significant attenuation in the passband, with the following characteristic (including the external RC):

| MCLK/Freq | 0 | 4608 | 2304 | 1536 | 1152 | 1024 | 922 |
|---|---|---|---|---|---|---|---|
| Attenuation (dB) | 0 | 0.1 | 0.4 | 0.9 | 1.6 | 2.1 | 2.6 |

This attenuation is compensated in the default filter coefficients by convolving the required FIR response with a 15-tap pre-emphasis FIR filter. The reconstruction filter and the FIR filters will track with the MCLK frequency, provided that the external RC is scaled in proportion. The FIR filter approximately cancels the reconstruction filter attenuation up to a frequency of MCLK/900.

Coefficients of the pre-emphasis FIR filter used in the default filter coefficients are shown below.

-0.00737876
-0.00987614
-0.0150585
-0.0206503
-0.0260154
-0.0304823
-0.0334478
1.22444
-0.0334478
-0.0304823
-0.0260154
-0.0206503
-0.0150585
-0.00987614
-0.00737876

If there is no source of attenuation or phase distortion external to the IC for which the user wishes to compensate, this filter can be combined with the main shaping filter as described in Section 5.4. If there are additional sources of attenuation or phase distortion, these can be catered for either by designing a new pre-emphasis filter that incorporates gain to compensate for the attenuation in the above table, or by designing another compensating filter and cascading with the filter described here. Both approaches are essentially equivalent.

### 5.4.2 Rx Path Details

The I and Q Rx data paths are nominally identical and consist of the following elements:

A. a continuous time anti-alias filter, which can be bypassed (2 poles at 100kHz, Q = 0.5 on-chip and 1 pole at 32kHz off chip), with the following characteristic:

| MCLK/Freq | 0 | 4608 | 2304 | 1536 | 1152 | 920 |
|---|---|---|---|---|---|---|
| Attenuation (dB) | 0 | 0.03 | 0.13 | 0.29 | 0.52 | 0.77 |

**Note:** The 32kHz pole is responsible for around 60% of the passband attenuation. The two on-chip poles attenuate by approximately 0.1dB at 10kHz (equivalent to MCLK/922 for MCLK=9.216MHz), while the decimation filter supplies 0.2dB at this frequency. Only the decimation filter attenuation will track with MCLK, while the other poles will remain fixed (subject to component tolerances). There is an 800ns variation in nominal group delay of this decimation filter up to 10kHz. This could be compensated in the FIR coefficients if considered significant.

B. a sigma delta ADC and decimation filter

C. a gain/offset adjustment block

D. a pair of programmable FIR filters (both 63-tap)

The passband attenuation caused by the Rx AAF and decimation filters is compensated in the default filter coefficients by convolving the required FIR response with a 15-tap pre-emphasis FIR filter. It approximately cancels the filter attenuation up to a frequency of 10kHz. This pre-emphasis FIR filter will track with the MCLK frequency, unlike the Rx AAF, but performance should be adequate for MCLK frequencies between 7MHz and 12MHz. The user is free to alter the external RC filter and/or to bypass the two on-chip poles of the AAF. In either case, the compensation FIR filter may require adjustment. In the event of a user designing their own Rx AAF, it is suggested that close attention is paid to the effect of component tolerances.

Two example sets of coefficients for this filter, for MCLK frequencies of 9.216MHz and 8.192MHz, are given below. These assume that external components R2 and C2 are NOT changed from the default values.

| MCLK=9.216MHz | MCLK=8.192MHz |
|---|---|
| -0.00113692 | 0 |
| -0.00227383 | -0.00112409 |
| -0.00568459 | -0.00449636 |
| -0.00909534 | -0.00899271 |
| -0.0147799 | -0.0134891 |
| -0.0193276 | -0.0191095 |
| -0.0227383 | -0.0224818 |
| 1.16307 | 1.14994 |
| -0.0227383 | -0.0224818 |
| -0.0193276 | -0.0191095 |
| -0.0147799 | -0.0134891 |
| -0.00909534 | -0.00899271 |
| -0.00568459 | -0.00449636 |
| -0.00227383 | -0.00112409 |
| -0.00113692 | 0 |

### 5.4.3 General Procedure for Reconfiguring the CMX980A FIR Filters

1. Obtain or design the required filter characteristic(s), either in s- or z-transform format. Note that all programmable FIR filters are sampled at a frequency of MCLK/64. This is equivalent to 144kHz for MCLK = 9.216MHz and 128kHz for MCLK = 8.192MHz.
   The task may be made slightly easier if the desired filter specification can be split into two filters of roughly similar complexity, but if this is not possible, or if it is preferable to treat the desired filter as a single entity, then a single impulse response can be considered as a polynomial in Z and factorized into two polynomials of appropriate length.
   The Tx 79-tap filter MUST have symmetrical (hence linear phase) coefficients, but all of the 63-tap filters allow asymmetrical coefficients. Hence, any prototype filter function aimed at the Tx 79-tap filter must be linear phase.
2. Obtain the impulse response of the desired filter shape, either by simulation of the filter response to an impulse, or by using an inverse Fourier transform. Convolve the impulse response with the impulse response of the appropriate compensation filter described above. This may be accomplished either by concatenating the filters and simulating the impulse response of the combined system, or by multiplying the Z transfer function polynomials together.
3. Use a standard windowing function (e.g. Blackman, Hamming) to limit the impulse response to 141 samples for the Tx, 125 samples for the Rx, or 79 and 63 samples (Tx), 63 and 63 samples (Rx), if designing the filters separately. Alternatively, it may be possible to approximate small outer coefficients to zero.
4. Scale the FIR coefficients to appropriate values. To maximize the use of dynamic range, scale the dc gain to give values similar to those given by the default coefficients (approximately 4.5x for Tx and 1.6x for Rx). Note that 16-bit coefficients are used in Rx filters, but only 12-bit for Tx filters, while the use of appropriate dc gains within the system may result in one or two of the most significant bits being redundant.
5. Load in your new coefficients, run the chip with maximum sized signals and check for internal overflows (see TxErrorStatus and RxErrorStatus Registers). Optimize individual filter gains to a level at least 5% below those which cause overflows.

## 5.5 Receiver Re-Synchronization

```
/* Pseudo-code for the use of re-synchronizing feature in the CMX980A */
/*                                                                    */
/* When using low data rate on the rx data serial port          */
/* alternate I & Q samples are transmitted.                     */
/* If synchronisation is lost with the serial port by           */
/* missing a data word, then use the method below to            */
/* re-gain sync and identify the correct channel for            */
/* data received.                                                     */
/*                                                                    */
/* It is assumed that the RxSampleSel bit has been set          */
/* to give the requred sample phasing                           */
/* i.e. I before Q  or Q before I.                              */

module begin

  SerialCmdWrite( 7'h21, 8'hff );  // Disable all interrupt sources

  SerialCmdRead( 7'h20 );              // read the RxErrorStatus register
                                       // to clear all pending interrupts
                                       // disregard returned data

  SerialCmdWrite( 7'h21, 8'h02 );      // Enable the EvenSamplePhase
                                       // interrupt

  wait (until interrupt generated ) // This is system dependent


  SerialRxRead() ;                     // Next Rx Data will be from the
                                       // channel selected
                                       // by the RxSampleSel bit in the
                                       // RxSetup1 reg
                                       // RxSampleSel = 0 -> Q
                                       // RxSampleSel = 1 -> I

 end

endmodule

function prototypes
SerialCmdWrite( 7 bit Register Address, 8 bit Data )
SerialCmdRead( 7 bit Register Address ) -> returns serial data
```

## 5.6 Guidelines for use of Power Save Modes

The CMX980A contains a number of power save modes. In order to maximize flexibility for different architectures and modes of operation, several register bits are available which control different parts of the device. Operation of the various control bits are described in the appropriate sections. These guidelines provide an overview of the power saving features.

### 5.6.1 Auxiliary Section

When one or more Auxiliary DACs are not required, they can be individually powered down using the **PowerDownCtrl** Register, bits [4:1]

When the auxiliary ADC channels are not required, the ADC will automatically power down if no ADC channel is selected. If ADC conversions are only required occasionally, these can be performed in single shot mode - the ADC will automatically power down between conversions.

When neither auxiliary ADC channels nor the RamDac functions are required, the auxiliary section digital logic can be powered down using the *Aux_ClkStopMode* bit in the **ClkStopCtrl** Register.

Note that the auxiliary ADC will power up within four master clock cycles, while the DAC circuits will power up in less than 5µs.

### 5.6.2 Tx Section

The Tx can be powered down by setting *TxEn* and *TxCtrlEn* bits inactive and the *TxClkStop* bit active. All of these bits are found in the **TxSetup** register. Note that *TxEn* should not be set inactive until the *TxPathEn* bit in the **TxFIFOStatus** Register becomes inactive.

The power up time for the Tx is limited by the filter response time. Thus, the analog circuitry will be correctly biased to receive data by the time the data emerges from the digital filters.

### 5.6.3 Rx Section

The Rx can be powered down by setting the *RxEn* bit in the **RxSetup1** Register inactive and by setting the *RxClkStop* and *RxIFClkStopMode* bits in the **ClkStopCtrl** Register active.

In addition, if the Rx AAF is not required, it is powered down using the *RxAafPowDn* bit in the **PowerDownCtrl** Register. This bit also serves as a multiplex select for the function.

All of the analog circuitry within the Rx will power up within 10µs. Thus, the time from power up to valid data appearing at the RxDat pin will be dominated by the digital filter group delay (nominally 8 symbol periods).

### 5.6.4 Tx and Rx Bias Section

When neither Tx nor Rx is required, the bias section can be powered down by setting the *BiasPowDn* bit in the **PowerDownCtrl** Register inactive. A small amount of current can also be saved by setting the *BiasChainPowDn* bit in the **PowerDownCtrl** Register inactive. However, this causes the voltage on BIAS1 pin, which is used as the internal "analog ground", to move towards $V_{DD}$ with a 250µs time constant. Up to 2ms should be allowed for this node to recover before the Tx or Rx is enabled.

### 5.6.5 Serial Interface Section

A small power saving can be made if it is possible to run with a serial interface clock rate of MCLK/8. This is accomplished by setting the *DataRateHi* bit in the **ConfigCtrl1** Register inactive. Note that this reduces the Rx output rate to four samples per symbol, although symbol timing can still be adjusted using the *RxSampleSel* bit in the **RxSetup1** Register and the vernier control in the **RxSetup2** Register.

When running with a low serial interface clock rate, it is possible to invoke the serial interface clock stop mode by using the *AutoClkStopMode* bit in the **ClkStopCtrl** Register. When active, this mode will stop all serial interface activity if there is no activity on *CmdFS* for more than 4096 master clock cycles. Note that this mode of operation stops the serial clock.

When the main serial interface activity involves loading symbols for transmission, the user can operate in hardware handshake mode by using the *TxHandshakeEn* bit in the **ConfigCtrl1** Register. Note that this mode stops the serial clock if the transmit path is enabled and the transmit FIFO is full, thus the use of the serial interface is less flexible.

# 6 Performance Specification

## 6.1 Electrical Performance

### 6.1.1 Absolute Maximum Ratings

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

| | Min. | Max. | Units |
|---|---|---|---|
| Supply | | | |
| $V_{DD}$ - $V_{SS}$ | -0.3 | 7.0 | V |
| $V_{CC1}$ - $V_{SS1}$ | -0.3 | 7.0 | V |
| $V_{CC2}$ - $V_{SS2}$ | -0.3 | 7.0 | V |
| $V_{CC3}$ - $V_{SSB}$ | -0.3 | 7.0 | V |
| $V_{DD1}$ - $V_{SSA}$ | -0.3 | 7.0 | V |
| Voltage on any pin to $V_{SS}$ | -0.3 | $V_{DD}$ + 0.3 | V |
| Voltage on any pin to $V_{SS1}$ | -0.3 | $V_{CC1}$ + 0.3 | V |
| Voltage on any pin to $V_{SS2}$ | -0.3 | $V_{CC2}$ + 0.3 | V |
| Voltage on any pin to $V_{SSA}$ | -0.3 | $V_{DD1}$ + 0.3 | V |
| Voltage on any pin to $V_{SSB}$ | -0.3 | $V_{CC3}$ + 0.3 | V |
| Current into or out of $V_{DD}$, $V_{CC1}$, $V_{CC2}$, $V_{CC3}$, $V_{DD1}$, $V_{SS}$, $V_{SS1}$, $V_{SS2}$, $V_{SSB}$ and $V_{SSA}$ | -30 | +30 | mA |
| Current into or out of any other pin | -20 | +20 | mA |
| Voltage differential between power supplies: | | | |
| ($V_{DD}$, $V_{CC1}$, $V_{CC2}$, $V_{CC3}$ and $V_{DD1}$) | 0 | 0.3 | V |
| ($V_{SS}$, $V_{SS1}$, $V_{SS2}$, $V_{SSB}$ and $V_{SSA}$) | 0 | 50 | mV |
| **L6 Package** | | | |
| Total Allowable Power Dissipation at $T_{AMB}$ = 25°C | – | 1050 | mW |
| Derating above 25°C | – | 13 | mW/°C above 25°C |
| Storage Temperature | -55 | +125 | °C |
| Operating Temperature | -40 | +85 | °C |
| **L7 Package** | | | |
| Total Allowable Power Dissipation at $T_{AMB}$ = 25°C | – | 800 | mW |
| Derating above 25°C | – | 9 | mW/°C above 25°C |
| Storage Temperature | -55 | +125 | °C |
| Operating Temperature | -40 | +85 | °C |

### 6.1.2 Operating Limits

Correct operation of the device outside these limits is not implied.

| | Notes | Min. | Max. | Units |
|---|---|---|---|---|
| Supply | | | | |
| $V_{DD}$ - $V_{SS}$ | | 3.0 | 5.5 | V |
| $V_{CC1}$ - $V_{SS1}$ | | 3.0 | 5.5 | V |
| $V_{CC2}$ - $V_{SS2}$ | | 3.0 | 5.5 | V |
| $V_{CC3}$ - $V_{SSB}$ | | 3.0 | 5.5 | V |
| $V_{DD1}$ - $V_{SSA}$ | | 3.0 | 5.5 | V |
| Operating Temperature | | -40 | +85 | °C |
| MCLK Frequency - (nominally 9.216MHz) | $3.0V < V_{DD} - V_{SS} < 4.5V$ | 0.5 | 9.5 | MHz |
| MCLK Frequency - (nominally 9.216MHz) | $4.5V < V_{DD} - V_{SS} < 5.5V$ | 0.5 | 12.5 | MHz |

### 6.1.3 Operating Characteristics

***Details in this section represent design target values and are not currently guaranteed.***

For the following conditions unless otherwise specified:

MCLK Frequency = 9.216MHz, Symbol Rate = 18k bits/sec, PowerDownCtrl[6] = '0', Rx AAF selected when Rx enabled.

($V_{DD}$ - $V_{SS}$) = ($V_{CC1}$ - $V_{SS1}$) = ($V_{CC2}$ - $V_{SS2}$) = ($V_{CC3}$ - $V_{SSB}$) = ($V_{DD1}$ - $V_{SSA}$) = 3.0V to 3.6V for 3.3V parameters, 4.5V to 5.5V, for 5.0V parameters. $T_{AMB}$ = - 40°C to +85°C. All typical values are at 3.3V and 5.0V, unless specified otherwise.

It is assumed that all powersave and clock-stop modes are selected where appropriate and that BIST is always disabled.

| | Notes | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| **5VDC Parameters (MCLK not toggled)** | | | | | |
| $I_{DD}$ (Tx powersaved) | 1 | – | 16.0 | TBD | mA |
| $I_{DD}$ (Rx powersaved) | 1 | – | 13.0 | TBD | mA |
| $I_{DD}$ (Aux powersaved) | 1 | – | 22.5 | TBD | mA |
| $I_{DD}$ (All powersaved) | 1 | – | – | 50 | µA |
| $I_{DD}$ (Not powersaved) | 1 | – | 25.5 | TBD | mA |
| **5VAC Parameters (MCLK at 9.216MHz)** | | | | | |
| $I_{DD}$ (Tx powersaved) | 1 | – | 35.0 | TBD | mA |
| $I_{DD}$ (Rx powersaved) | 1 | – | 25.0 | TBD | mA |
| $I_{DD}$ (Aux powersaved) | 1 | – | 47.5 | TBD | mA |
| $I_{DD}$ (All powersaved except Autoclock) | 1 | – | 5.5 | TBD | mA |
| $I_{DD}$ (All powersaved) | 1 | – | 4.0 | TBD | mA |
| $I_{DD}$ (Not powersaved) | 1 | – | 52.0 | TBD | mA |
| **3.3VDC Parameters (MCLK not toggled)** | | | | | |
| $I_{DD}$ (Tx powersaved) | 1 | – | 9.5 | TBD | mA |
| $I_{DD}$ (Rx powersaved) | 1 | – | 8.0 | TBD | mA |
| $I_{DD}$ (Aux powersaved) | 1 | – | 13.5 | TBD | mA |
| $I_{DD}$ (All powersaved) | 1 | – | – | 50 | µA |
| $I_{DD}$ (Not powersaved) | 1 | – | 15.5 | TBD | mA |
| **3.3VAC Parameters (MCLK at 9.216MHz)** | | | | | |
| $I_{DD}$ (Tx powersaved) | 1 | – | 21.0 | TBD | mA |
| $I_{DD}$ (Rx powersaved) | 1 | – | 15.0 | TBD | mA |
| $I_{DD}$ (Aux powersaved) | 1 | – | 28.5 | TBD | mA |
| $I_{DD}$ (All powersaved except Autoclock) | 1 | – | 2.5 | TBD | mA |
| $I_{DD}$ (All powersaved) | 1 | – | 1.5 | TBD | mA |
| $I_{DD}$ (Not powersaved) | 1 | – | 31.0 | TBD | mA |
| **MCLK Input** | | | | | |
| 'High' pulse width | 2 | 30.0 | – | – | ns |
| 'Low' pulse width | 2 | 30.0 | – | – | ns |
| Input impedance (at 100Hz) | | 10.0 | – | – | MΩ |

**Notes:**

1. Not including any current drawn from the device pins by external circuitry.
2. Timing for an external input to the MCLK pin.

**General Points:**

3. The current quoted when MCLK is not toggled is essentially analog current (digital current is negligible in this case), while the current quoted when MCLK is toggled is a combination of analog and digital current.
4. Powering down the Rx AAF when Rx and Bias sections are enabled reduces 3.3V current by 0.9mA, 5.0V current by 1.5mA.
5. BiasChainPowDn and BiasPowDn must be inactive (i.e. bias section powered up) in order to power up the Tx and Rx sections, but NOT the auxiliary section. It is recommended that BiasChainPowDn is activated at least 0.5ms before powering up the Tx or Rx, owing to the long time constant associated with this function. When the Tx and Rx are powered down, setting BiasPowDn active reduces current by a further 200µA, while setting BiasChainPowDn active reduces current by a further 30-50µA.
6. Currents in other modes can be calculated from the above figures. For example, operation at 3.3V with Tx and Auxiliary sections power saved:
   From the table, auxiliary current is approximately 2.5mA (31-28.5mA).
   Subtracting this from Tx powersaved current (21mA) gives a predicted current of 18.5mA.
7. Supply currents for other MCLK frequencies can be calculated by assuming the analog current is constant and the digital current is proportional to MCLK frequency.

#### 6.1.3.1 Transmit Parameters

| Parameter | Notes | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| Input bit rate (2 bits per symbol) | | – | MCLK/256 | – | bps |
| Number of Channels | | – | 2 | – | |
| Modulation Type | | – | π/4 DQPSK | – | |
| FIR filter sampling rate | | – | MCLK/64 | – | Hz |
| DAC output update rate | | – | MCLK/4 | – | Hz |
| DAC resolution | | – | 14 | – | Bits |
| Integral accuracy | | – | – | ±2 | LSB |
| Differential accuracy | | – | – | ±1 | LSB |
| Signal to noise plus distortion | 1 | 65 | 70 | - | dB |
| Offset (without adjustment) | | – | ± 20.0 | ± 40.0 | mV |
| Gain matching, I to Q (without adjustment) | | – | – | ± 0.25 | dB |
| Phase matching, I to Q | | – | – | ± 0.5 | Degrees |
| Storage time | | – | – | 20 | Symbols |
| I,Q output level | | | | | |
| ($V_{CC}$ = 5.0V) | 2 | 3.8 | 4.0 | 4.2 | V |
| ($V_{CC}$ = 3.3V) | | 2.49 | 2.62 | 2.75 | V |
| Adjacent Channel Power | 3 | | | | |
| at MCLK/384 frequency offset | | – | -70 | -68 | dBc |
| at MCLK/192 frequency offset | | – | -78 | -76 | dBc |
| at MCLK/128 frequency offset | | – | -80 | -78 | dBc |
| at MCLK/96 frequency offset | | – | -88 | -86 | dBc |
| at MCLK/48 frequency offset | | – | -90 | -88 | dBc |
| at MCLK/20 frequency offset | | – | -92 | -90 | dBc |
| **TETRA Specific Parameters** | | | | | |
| Gain matching, (I or Q) to ideal Tx (normalized, 0 - 9kHz) | | – | – | ± 0.3 | dB |
| RRC Roll-off coefficient (α) | | – | 0.35 | – | |
| \|H(f)\| 0 - 5.85kHz | | -0.1 | 0.0 | +0.1 | dB |
| \|H(f)\| at 9kHz | | -2.9 | -3.0 | -3.1 | dB |
| \|H(f)\| at 10.05kHz | | -6.6 | -6.8 | -7.0 | dB |
| \|H(f)\| at 12.15kHz | | -30.0 | – | – | dB |
| Adjacent Channel Power | 4 | | | | |
| at 25kHz frequency offset | | – | -72 | -70 | dBc |
| at 50kHz frequency offset | | – | -80 | -78 | dBc |
| at 75kHz frequency offset | | – | -82 | -80 | dBc |
| at 100kHz frequency offset | | – | -90 | -88 | dBc |
| at 200kHz frequency offset | | – | -93 | -91 | dBc |
| at 500kHz frequency offset | | – | -95 | -93 | dBc |
| at 5MHz frequency offset | | – | -104 | -102 | dBc |

| Parameter | Notes | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| Adjacent Channel Power during ramping over 5 symbols at 25kHz frequency offset | | | | | |
| Linear Ramping | | – | -55 | 53 | dBc |
| Sigmoidal Ramping | | – | -60 | -57 | dBc |
| Vector Error (peak) | 5 | – | 0.045 | 0.07 | |

**Transmit Parameter Notes:**

1. Measured with an MCLK/4096 test signal in MCLK/1024 bandwidth
2. Peak to peak, differential at maximum gain
3. Power measured through an MCLK/460 filter centered at the stated frequency offset, relative to power measured through an MCLK/460 filter centered on the main channel, with Tx gain set to $0x5A7 (-3dB below maximum).
4. Power measured through an ideal RRC filter ($\alpha = 0.35$) centered at the stated frequency offset, relative to power measured through an ideal RRC filter ($\alpha = 0.35$) centered on the main channel, with Tx gain set to $0x5A7 (-3dB below maximum).
5. Vector errors are measured with ideal IF and RF sections, after gain and offset adjustment, and specified as a fraction of the nominal vector value.

**General Points:**

6. Unless otherwise indicated, all parameters refer to the entire Tx baseband I and Q channels, with recommended external components and default filter coefficients.
7. A gain multiplier function allows independent proportional control of each channel. The multiplier is a 12-bit word for each channel, input via the serial interface, representing a value from 0 to 1. This multiplication is applied to the signals from the FIR filters.
8. Offset adjustment for each channel is available by loading a 12-bit word into the transmit offset register via the serial interface.

### 6.1.3.2 Receive Parameters

| Parameter | Notes | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| Input impedance (Capacitive load to $V_{SS1}$ or $V_{SS2}$) | | – | – | 10.0 | pF |
| Input impedance (Source impedance should be < 1kΩ) | | 100 | – | – | kΩ |
| Differential Input voltage | | | | | |
| ($V_{CC}$ = 5.0V) | 1 | – | 3.4 | 3.8 | $V_{P-P}$ |
| ($V_{CC}$ = 3.3V) | 1 | – | 2.25 | 2.5 | $V_{P-P}$ |
| Signal to Noise | 2 | 83 | 86 | – | dB |
| Signal to Noise plus distortion | 2 | 80 | 82 | – | dB |
| 3rd order intercept (3.3V operation) | 3 | – | 200 | – | $V_{P-P}$ |
| ADC sampling rate | | – | MCLK/4 | – | Hz |
| ADC resolution | | – | 16 | – | Bits |
| Integral accuracy | | – | – | ±1 | LSB |
| Differential accuracy | | – | – | ±1 | LSB |
| FIR filter sampling rate | | | | | |
| (Decimation section) | | – | MCLK/4 | – | Hz |
| (RRC sections) | | – | MCLK/64 | – | Hz |
| Output rate (16 bit words per channel) - selectable | 4 | – | MCLK/64 or MCLK/128 | – | Hz |
| Offset (Without adjustment) | | – | ±15 | ±40 | mV |
| Gain matching, I to Q (Without adjustment, 0 - 10kHz) | | – | – | ± 0.1 | dB |
| Phase matching, I to Q (0 - 10kHz) | | – | – | ± 0.5 | Degrees |
| Storage time (with default coefficients) | | – | – | 17 | Symbols |
| With internal anti-alias filter disabled: | | | | | |
| External anti-alias requirements | 5 | | | | |
| at MCLK/70 | 6 | – | – | -15 | dB |
| at MCLK/4 | 6 | – | – | -110 | dB |
| With internal anti-alias filter enabled: | | | | | |
| External anti-alias requirements | 7, 8 | | | | |
| at MCLK/70 | 6 | – | – | -13 | dB |
| **TETRA Specific Parameters** | | | | | |
| RRC Roll-off coefficient (α) | 9 | – | 0.35 | – | |
| \|H(f)\| 0 - 5.85kHz | | -0.2 | 0 | +0.2 | dB |
| \|H(f)\| at 9kHz | | -2.9 | -3.0 | -3.1 | dB |
| \|H(f)\| at 10.05kHz | | -6.5 | -6.9 | -7.3 | dB |
| \|H(f)\| at 12.15kHz | | -30 | – | – | dB |
| \|H(f)\| at 16kHz | | -70 | – | – | dB |
| \|H(f)\| > 25kHz | | -85 | – | – | dB |

**Rx Notes:**

1. Note this means ±0.85V or ±0.56V on each input of the differential pair.
2. Both measured with MCLK/4096 Hz test signal, in MCLK/1024 Hz bandwidth.
3. Extrapolated from third harmonic distortion at maximum signal.
4. Output via the serial interface at MCLK/2 or MCLK/4.
5. These anti-alias filter requirements can be supplied by IF channel filtering, baseband filtering or a combination of both. It is recommended that, in order to maximize the performance obtained from the CMX980A for TETRA applications, at least 10dB and 25dB attenuation be provided at MCLK/70 and MCLK/4 respectively, prior to an external AGC function. Other applications may require less stringent external filtering.
6. With respect to maximum input level.
7. This should be supplied by a network equivalent to Figure 2.
8. These figures assume that 10-15dB attenuation at MCLK/4 is provided by IF channel filtering or by additional filtering at baseband. Note that the recommended configuration shown in Figure 2 includes an AGC after the 32kHz pole
9. With default coefficients and internal anti-alias filter selected

**General Points:**

10. Offset adjustment for each channel is available by loading a 16-bit word into the receive offset register via the serial interface.
11. Optimally, as much anti-alias filtering as possible should be carried out prior to any AGC function before the receive inputs. This allows the AGC to act on a reduced bandwidth signal and thereby improve the relative magnitude of the wanted part. The device has been designed to reduce the complexity of any external anti-alias filter as much as possible and a 4-pole Butterworth or 3-pole Chebyshev with a -3dB point at about MCLK/150 should be adequate. The internal anti-alias filter, if used, cannot provide the required 110dB attenuation at MCLK/4 and must be supplemented by external filtering. An example of a suitable external filter structure is shown in Figure 2.
12. Anti-alias filter requirements quoted are for mobile station applications. For base station applications, all attenuation figures should be increased by 3dB.

### 6.1.3.3 Auxiliary Circuit Parameters

| Parameter | Notes | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| **DACs** | | | | | |
| Resolution | | – | 10 | – | Bits |
| Settling time to 0.5 LSB | 1 | – | – | 10 | µs |
| Output resistance | | – | – | 250 | Ω |
| Integral non-linearity | | – | – | ±4 | Bits |
| Differential non-linearity | 2 | – | – | ±1 | Bit |
| Zero error (offset) | | – | – | ±20 | mV |
| Power (all DACs operating) | 6 | – | – | 10 | mW |
| Resistive Load | | 5.0 | – | – | kΩ |
| Output noise voltage in 30kHz bandwidth | | – | 10 | – | $\mu V_{RMS}$ |
| **ADC and Multiplexed inputs** | | | | | |
| Input source impedance | 3 | – | – | 25 | kΩ |
| Resolution | | – | 10 | – | Bits |
| Input signal "linear rate of change" for < 1 bit error | | – | – | 0.27 | mV/µs |
| Conversion time | 4 | 80/MCLK | - | 160/MCLK | Sec |
| Integral non-linearity | | – | – | ±2 | Bits |
| Differential non-linearity | 5 | – | – | ±1 | Bit |
| Zero error (offset) | | – | – | ±20 | mV |
| A-D clock frequency | | MCLK/16 | – | MCLK/8 | (Hz) |
| Input capacitance | | – | – | 5 | pF |
| Power | 6 | – | – | 3 | mW |

**Auxiliary Circuit Notes:**

1. Worst case large signal transition.
2. Guaranteed monotonic.
3. Gives <1 bit additional error under nominal conditions
4. Conversion time = $\frac{10}{A\text{-}D\ clock\ frequency}$, where *A-D clock frequency* is programmable to either MCLK/8 or MCLK/16, as shown in the table below:

| | **Minimum** | **Maximum** |
|---|---|---|
| A-D Clock Frequency: | MCLK/16 | MCLK/8 |
| Conversion Time: | 80/MCLK | 160/MCLK |

   Due to latency in the control logic, the maximum sample rate is lower than this. Refer to the Output Rate details in the Receive Parameters section.
5. No missing codes.
6. Measured at 3.3V supply voltage.

### 6.1.3.4 Operating Characteristics - Timing Diagrams

The following timings are provisional:

| Timing Parameters - Serial Ports | Marker | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| MCLK to SClk out - low to high | $t_{CSLH}$ | 15 | – | 50 | ns |
| MCLK to SClk out - high to low | $t_{CSHL}$ | 10 | – | 35 | ns |
| CmdDat setup to falling edge of SClk | $t_{SIS}$ | 35 | – | – | ns |
| CmdFS setup to falling edge of SClk | $t_{SIS}$ | 35 | – | – | ns |
| CmdDat hold from fall edge of SClk | $t_{SIH}$ | – | – | 0.0 | ns |
| CmdFS hold from fall edge of SClk | $t_{SIH}$ | – | – | 0.0 | ns |
| RxDat propagation from rising edge of SClk | $t_{SOP}$ | – | – | 5.0 | ns |
| RxFS propagation from rising edge of SClk | $t_{SOP}$ | – | – | 5.0 | ns |
| CmdRdDat propagation from rising edge of SClk | $t_{SOP}$ | – | – | 5.0 | ns |
| CmdRdFS propagation from rising edge of SClk | $t_{SOP}$ | – | – | 5.0 | ns |
| RxDat hold from rising edge of SClk | $t_{SOH}$ | -5.0 | – | – | ns |
| RxFS hold from rising edge of SClk | $t_{SOH}$ | -5.0 | – | – | ns |
| CmdRdDat hold from rising edge of SClk | $t_{SOH}$ | -5.0 | – | – | ns |
| CmdRdFS hold from rising edge of SClk | $t_{SOH}$ | -5.0 | – | – | ns |
| **Cmd port in Bi-dir mode ** | | – | – | – | |
| CmdDat propagation from rising edge of SClk | $t_{SOP}$ | – | – | 7.0 | ns |
| CmdDat hold from rising edge of SClk | $t_{SOH}$ | -7.0 | – | – | ns |

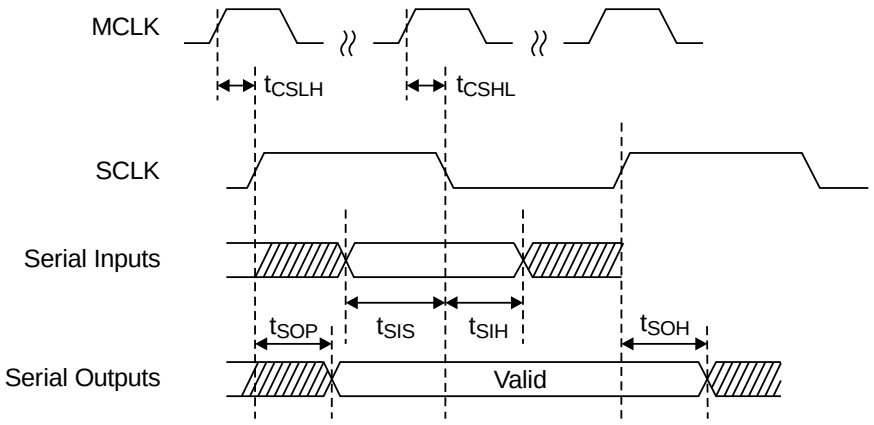



**Figure 5: Serial Port Interfaces - Timing Parameters**

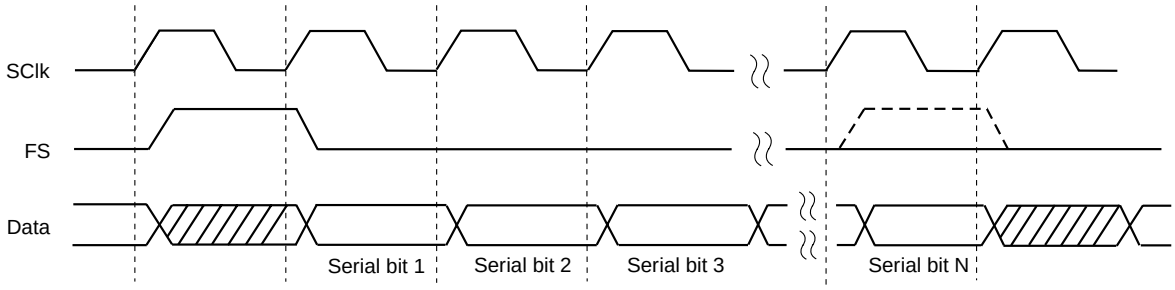



**Figure 6: Basic Serial Port Signals**

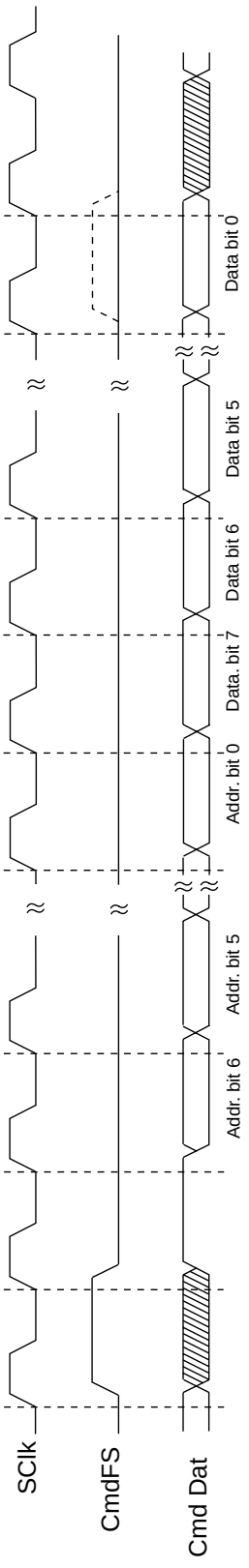



**Figure 7: Command Write Operation**

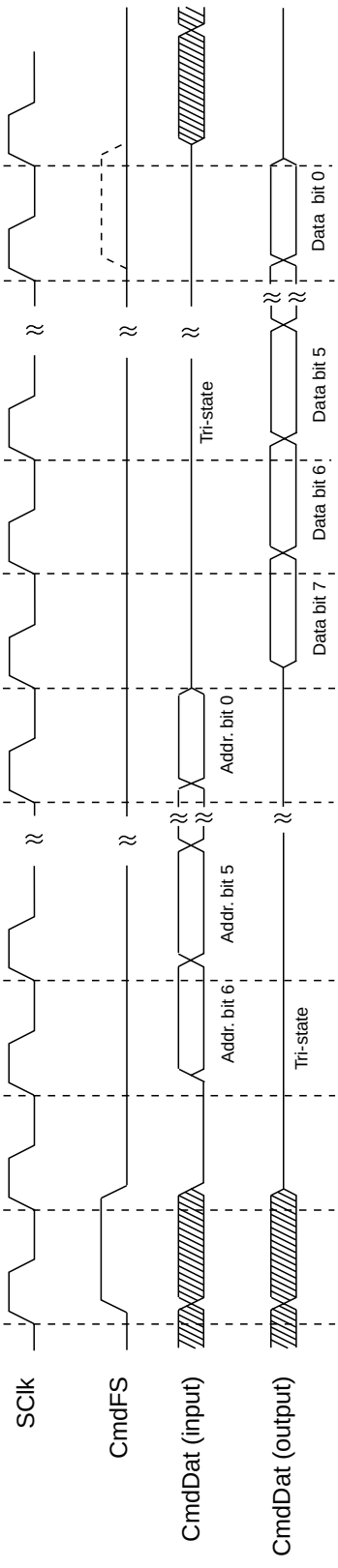



**Figure 8: Bi-directional Command Read Operation**

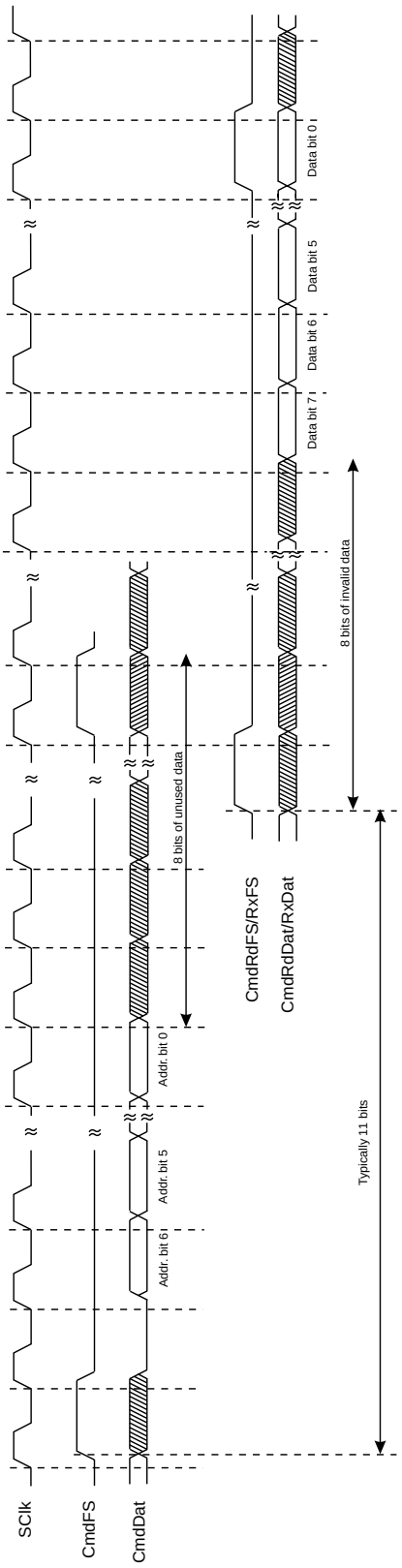



**Figure 9: Non bi-directional Command Read Operation**

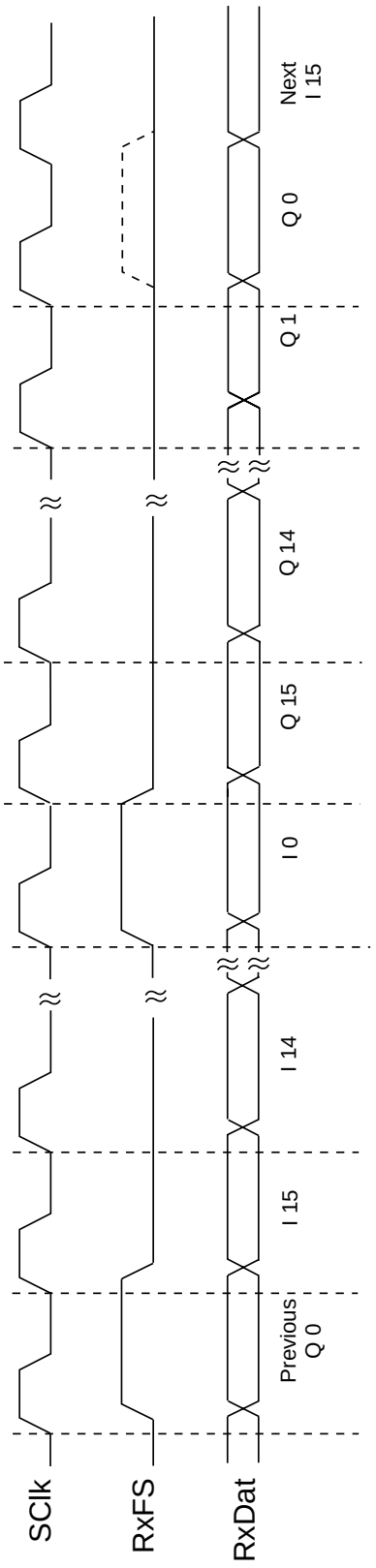
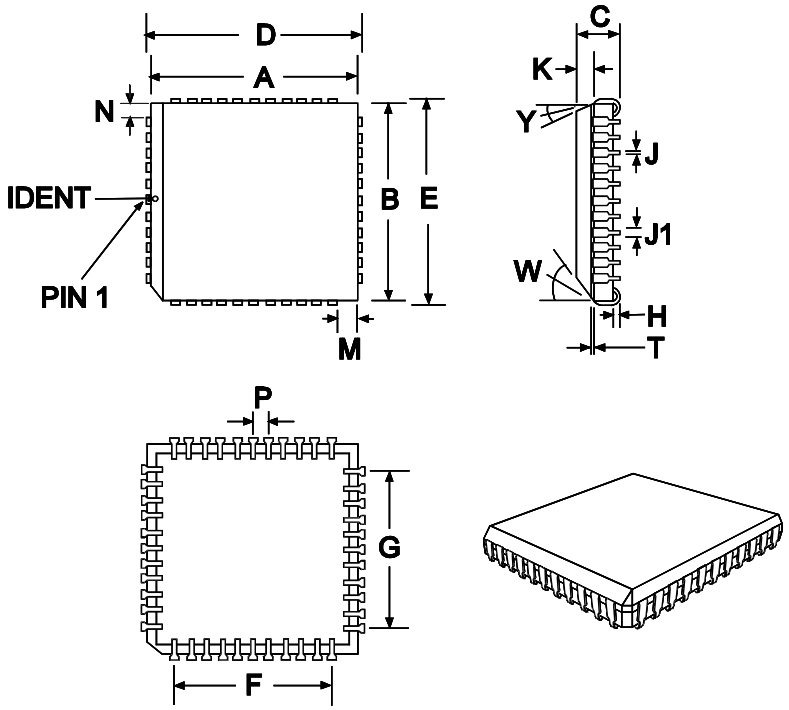



**Figure 10: Rx Data Serial Port Read Operation**

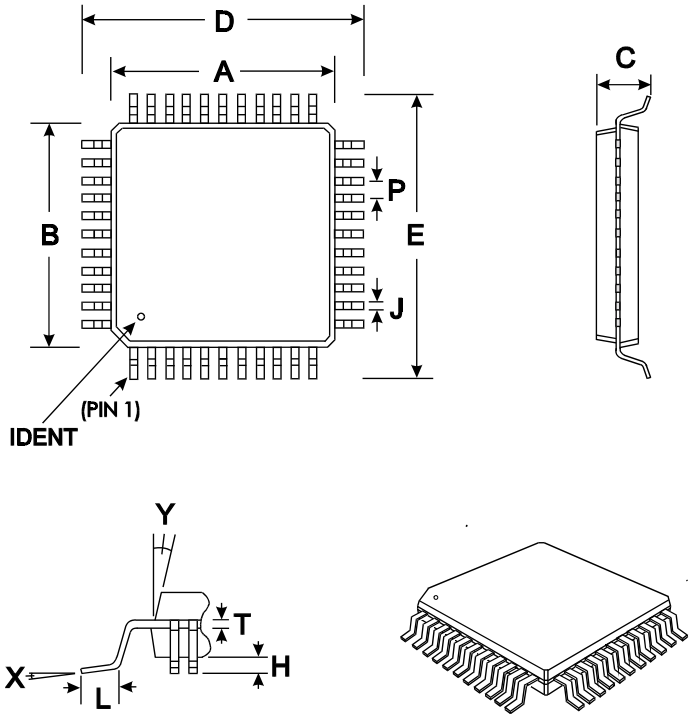
## 6.2 Packaging




| DIM. | MIN. | TYP. | MAX. |
|---|---|---|---|
| A | 0.650 (16.51) | | 0.656 (16.66) |
| B | 0.650 (16.51) | | 0.656 (16.66) |
| C | | | 0.180 (4.57) |
| D | 0.685 (17.40) | | 0.695 (17.65) |
| E | 0.685 (17.40) | | 0.695 (17.65) |
| F | | 0.500 (12.7) | |
| G | | 0.500 (12.7) | |
| H | 0.020 (0.51) | | |
| J | 0.013 (0.33) | | 0.021 (0.53) |
| J1 | | 0.029 (0.74) | |
| K | | 0.070 (1.78) | |
| M | | 0.062 (1.57) | |
| N | | 0.062 (1.57) | |
| P | | 0.050 (1.27) | |
| T | | 0.010 (0.25) | |
| W | | 0.30 (7.62) x 45° | |
| Y | | 7° | |

NOTE : All dimensions in inches (mm.)
Angles in degrees

**Figure 11: 44-pin PLCC (L6) Mechanical Outline: *Order as part no. CMX980AL6***




| DIM. | MIN. | TYP. | MAX. |
|---|---|---|---|
| A | 0.390 (9.90) | | 0.398 (10.11) |
| B | 0.390 (9.90) | | 0.398 (10.11) |
| C | 0.055 (1.40) | | 0.063 (1.60) |
| D | 0.465 (11.80) | | 0.480 (12.20) |
| E | 0.465 (11.80) | | 0.480 (12.20) |
| H | 0.002 (0.04) | | 0.006 (0.16) |
| J | 0.009 (0.24) | | 0.018 (0.45) |
| L | 0.016 (0.40) | | 0.030 (0.75) |
| P | | 0.031 (0.80) | |
| T | 0.004 (0.09) | | 0.008 (0.20) |
| X | 0° | | 7° |
| Y | 10° | | 14° |

NOTE : All dimensions in inches (mm.)
Angles in degrees

**Figure 12: 44-pin QFP (L7) Mechanical Outline: *Order as part no. CMX980AL7***