CML Semiconductor Products

PRODUCT INFORMATION

FX909 GMSK “Packet Data” Modem

Publication D/909/1 July 1994
Advance Information

Features

- **Automatic Protocol Handling** [Bit and Frame Sync, Block Formatting, CRC and FEC, Interleaving and Scrambling]
- **Half-Duplex Operation at 4,000 to 19,200b/s**
- **GMSK Tx Signal Filtering (BT = 0.3)**
- **Low-Power; 3mA [TYP] Operating**
- **‘Mobitex’ and General-Purpose Packet Data Applications**
- **8-Bit Parallel µController Interface**
- **Bandwidth Efficient**
- **Reduces Host Processing Load**
- **Simple Software Implementation**

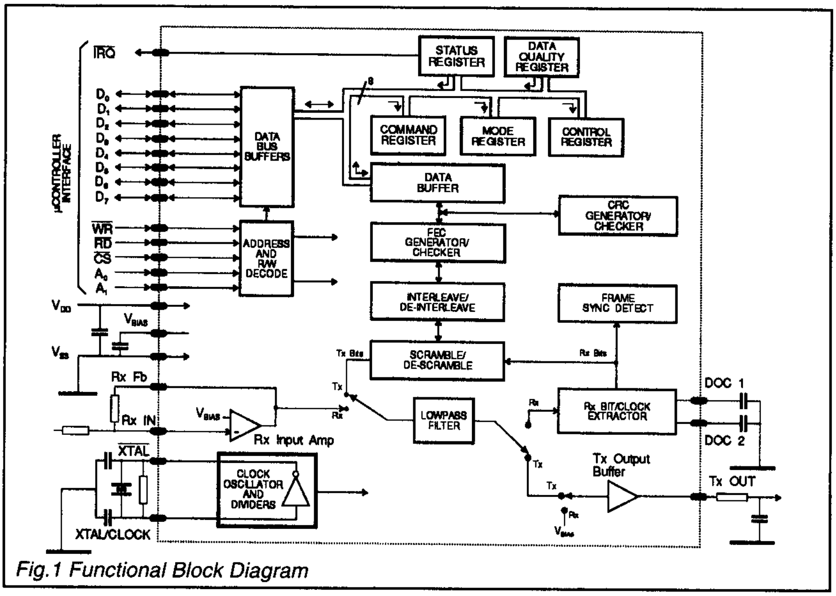


Fig.1 Functional Block Diagram

FX909

Brief Description

The FX909 is a single-chip half-duplex GMSK modem which offers many benefits to the manufacturer and programmer of medium to high-speed radio packet-data links using ‘Mobitex’ or general-purpose packet protocol.

❍ **Automatic handling (Tx/Rx) of Frame structure and Data Blocks ......** will reduce the processing load on the host µController. Requiring service by the µController only once per Tx or Rx Data Block, the FX909 will perform as much as possible of the computationally intensive work involved in the handling of ‘Mobitex’ protocol, including CRC and FEC operations, Frame Sync detection, Interleaving and Scrambling.

❍ **Gaussian Minimum Shift Keying (GMSK) modulation ......** provides the basis for an extremely good relationship between the RF bandwidth, Data bit-rate and Bit-error-rate.

❍ **Low-power, high-speed operation ......** selectable 4,000b/s to 19,200b/s with a typical operating requirement of only 3mA at 5 volts.

❍ **Signal acquisition and tracking ......** allows for the rapid acquisition of received messages, followed by automatic tracking of signal variations.
Both PLL bandwidth and Rx signal level measurement circuitry will react automatically as programmed.

Rx and Tx data and control between the host µController and the FX909 is via an 8-bit bi-directional parallel interface; input and output signals to and from the radio system are in analogue form suitable for connection to the radio's discriminator and frequency modulator.

The FX909 is available in both 24-pin DIL and Surface-Mount packages.

Introduction

Table of Contents

FX909 Circuit Descriptions (See Figure 2)

Data Bus Buffers

Eight bi-directional 3-state logic-level buffers between the modem's internal registers and the controlling µController's data-bus lines.

$\overline{RD}$, $\overline{WR}$ $\overline{CS}$ and Address Inputs A_0 and A_1

Control the transfer of data bytes between the µController and the modem's internal registers, according to the state of the Write and Read enable ($\overline{WR}$ and $\overline{RD}$) inputs, the Chip Select ($\overline{CS}$) input and the Register Address inputs (A_0 and A_1).

The Data Bus Buffers and Address & R/W Decode blocks provide a byte-wide parallel µController interface.

Status and Data Quality Registers

8-bit registers which the µController can read to determine the status of the modem and the received data quality.

Command, Mode and Control Registers

The values written by the µController to these 8-bit registers control the operation of the modem.

Data Buffer

An 18-byte buffer used to hold Rx or Tx data to or from the µController.

CRC Generator/Checker

A circuit which generates (transmit mode) or checks (receive mode) the Cyclic Redundancy Checksum bits which are included in transmitted Data Blocks so that the receive modem can detect transmission errors.

FEC Generator/Checker

In transmit mode this circuit calculates and adds the Forward Error Correction information (4 bits) to each byte presented to it. In receive mode the FEC information is used to correct most transmission errors that may have occurred in a Data Block or in Frame Head control bytes.

Tx Output Buffer

A unity-gain amplifier used in the transmit mode to buffer the output of the Tx Lowpass Filter. In receive mode, the input of this buffer is connected to V_{BIAS}. When changing from Rx to Tx mode the input to this buffer will be connected to V_{BIAS} for 2 bit periods to prevent unwanted signals, from the lowpass filter, appearing at the Tx OUT pin.

FX909 Circuit Descriptions ......

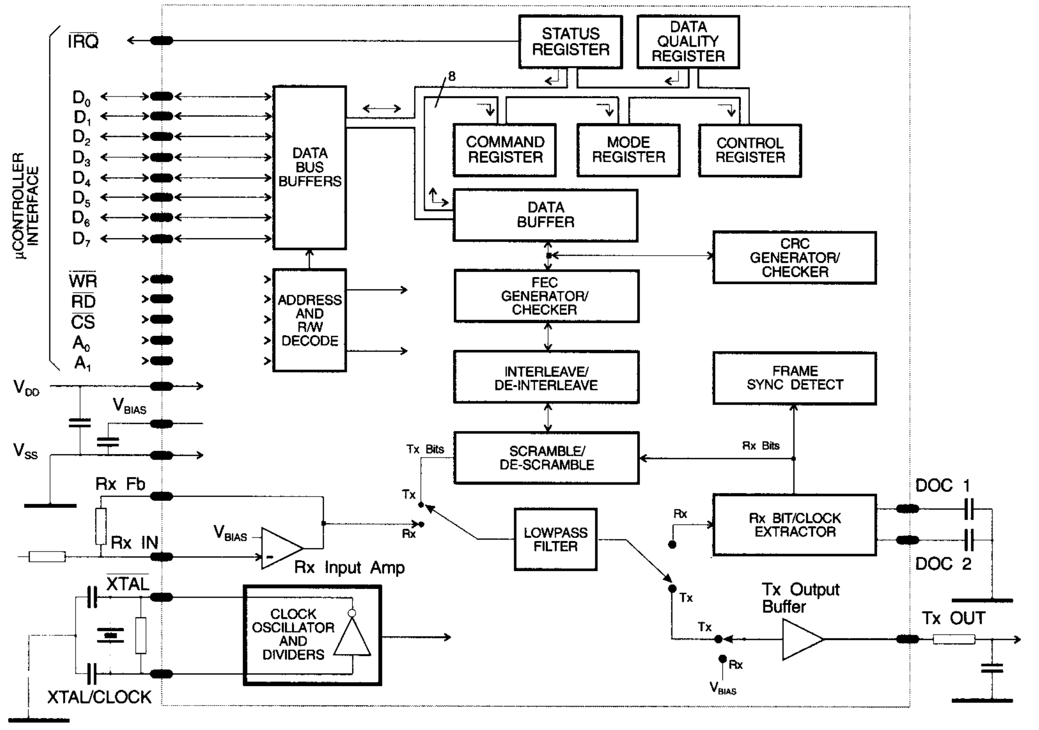


Fig.2 Functional Block Diagram

Rx Input Amp

Allows the received signal input to the modem to be set to the optimum level by suitable selection of the external components.

Interleave/De-Interleave Buffer

Interleaves data bits within a data block before transmission and de-interleaves the received data block so that the FEC system is best able to handle short noise bursts or fades.

Frame Sync Detect

This circuit, which is only active in the receive mode, is used to look for the user-specified 16-bit Frame Synchronisation pattern which is transmitted to mark the start of every frame.

Scramble/De-Scramble

This block may be used to scramble/descramble the transmitted and received Data Block by modulating it with a 511-bit pseudo-random sequence. Scrambling smooths the transmitted spectrum especially when repetitive sequences are to be transmitted.

Rx Bit/Clock Extraction

These circuits, which operate only in receive mode, extract a bit-rate clock from the received signal, and measure the received signal amplitude and its dc offset.

This information is then used to extract the received bits and also to provide an input to the received Data Quality measuring circuit.

Clock Oscillator and Dividers

This circuit derives the transmit bit rate (and the nominal receive bit rate) by frequency division of a reference frequency which may be generated by the on-chip Xtal oscillator or fed from an external source.

(Tx/Rx) Lowpass Filter

This filter, which is used in both transmit and receive modes, is a low-pass transitional gaussian filter.

In Tx mode the filter bandwidth is set for a loss of 3dB at 0.3 times the selected bit rate (BT = 0.3) and the bits are passed through this filter to eliminate the high frequency components which would otherwise cause interference into adjacent radio channels.

In Rx mode this filter is used with an increased BT factor (0.56) to reject HF noise so that the signal is in a suitable condition for extracting the received data.

Pin Functions

FX909 (J4 and L2)	
1	$\overline{\textbf{IRQ}}$**:** A 'wire-ORable' output for connection to the controlling μController's Interrupt Request input. This output has a low-impedance pull-down to V_{SS} when active, and is high-impedance when inactive.
2	$\mathbf{D_7}$**:**
3	$\mathbf{D_6}$**:**
4	$\mathbf{D_5}$**:**
5	$\mathbf{D_4}$**:** 8 bi-directional 3-state μController interface data lines.
6	$\mathbf{D_3}$**:**
7	$\mathbf{D_2}$**:**
8	$\mathbf{D_1}$**:**
9	$\mathbf{D_0}$**:**
10	$\overline{\textbf{RD}}$**:** An active-low logic level input used to control the reading of data from the modem into the controlling μController.
11	$\overline{\textbf{WR}}$**:** An active-low logic level input used to control the writing of data into the modem from the controlling μController.
12	$\mathbf{V_{SS}}$**:** The negative supply rail (ground).
13	$\overline{\textbf{CS}}$**:** An active-low logic level input to the modem used to enable a data Read or Write operation (see Figure 25, Timing).
14	$\mathbf{A_0}$**:** Two logic-level modem register selection inputs.
15	$\mathbf{A_1}$**:**
16	$\overline{\textbf{Xtal}}$**:** The output of the on-chip Xtal oscillator.
17	**Xtal/Clock:** The input to the on-chip Xtal oscillator. Note that attempts at operation of the FX909 without a suitable Xtal or clock input will increase required I_{DD} and place the μController interface into an undefined state.
18	**Doc 2:** Connections to the internal Rx signal level measurement circuitry. Capacitors as described in Figure 3 should be fitted between each of these pins and V_{SS}.
19	**Doc 1:**
20	**Tx Out:** The Tx signal output from the modem.
21	$\mathbf{V_{BIAS}}$**:** The internal circuitry bias line, held at $V_{DD}/2$, this pin must be decoupled to V_{SS} by a capacitor mounted close to the device pins.
22	**Rx In:** The input to the Rx input amplifier.
23	**Rx Fb (Rx Feedback):** The output of the Rx input amplifier, and the input to the (Rx) Lowpass Filter.
24	$\mathbf{V_{DD}}$**:** The positive supply. Levels and voltages within the modem are dependent upon this supply. This pin should be decoupled to V_{SS} by a capacitor mounted close to the device pins.

Application Information

External Components

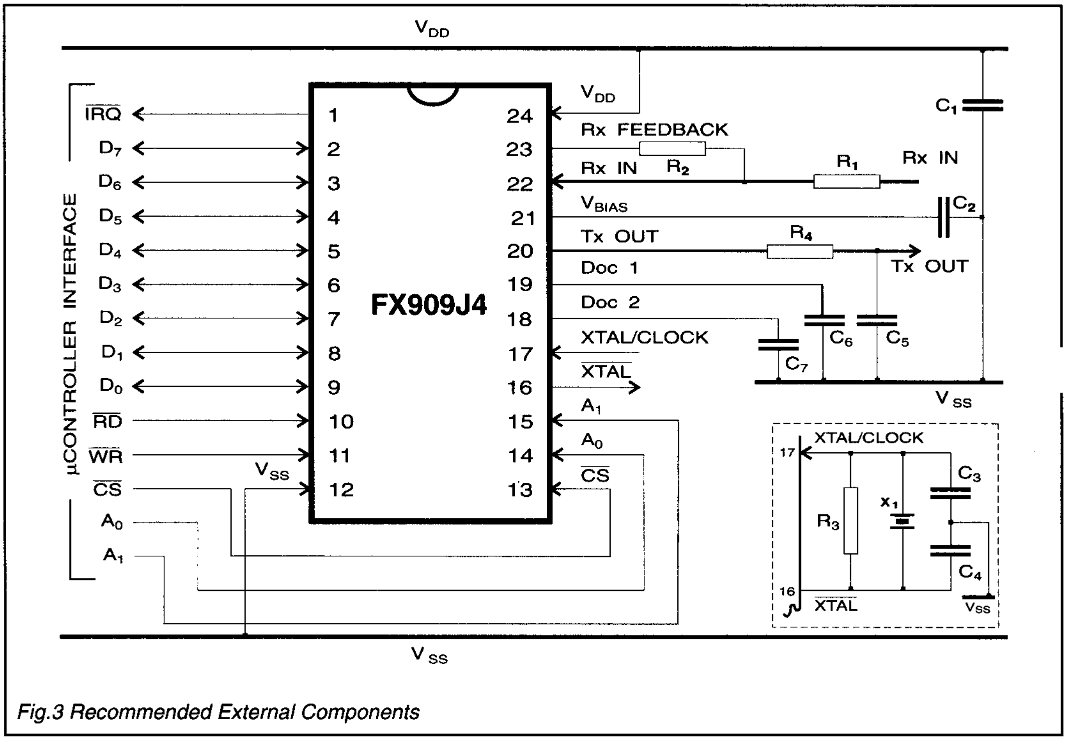


Fig.3 Recommended External Components

Component	Value	Tolerance
R_1	Note 1	±10%
R_2	100kΩ	±10%
R_3	1.0MΩ	±20%
R_4	Note 2	±5%
C_1	0.1µF	±20%
C_2	0.1µF	±20%
C_3	Note 3	±20%
C_4	Note 3	±20%
C_5	Note 2	±10%
C_6	Note 4	±20%
C_7	Note 4	±20%
X_1	Note 5	

Installation Notes

1. Resistors R_1 and R_2, with the Rx Input Amplifier, set the signal input level to the modem.
   The value of R_1 should be calculated to give 1.0v p-p at the Rx Feedback pin for a received 11110000... sequence. The dc level of the received signal should be adjusted so that the signal at the modem's Rx Feedback pin is centred around V_{BIAS}.

2. External components R_4 and C_5 form an RC lowpass filter between the Tx Buffer output (Tx OUT) and the input to the radio's frequency modulator; this is an important part of the Tx signal filtering. These components may form a part of any dc level shifting and gain adjustment circuitry.
   The ground connection (V_{SS}) of C_5 should be positioned to give maximum attenuation of high frequency noise into the modulator.
   R_4 and C_5 should be chosen so that the product of R_4 (Ohms) and C_5 (Farads) is:

   $$\frac{0.34}{\text{bit rate}} \text{ bits per sec}$$

   R_4 should be not less than 47kΩ and the value used for C_5 should take into account parasitic capacitance.

Examples	R_4	C_5
8000b/s	100kΩ	430pF
4800b/s	100kΩ	710pF

   The 'eye' diagram of the transmitted signal (after the external R_4/C_5 network) is shown in Figure 5.

Continued on next page ......

Application Information ......

Installation Notes ......

3. The values used for C_3 and C_4 should be suitable for the frequency of X_1.
   As a guide:
   $C_3 = C_4 = 33pF$ for $X_1 < 5.0MHz$.
   $C_3 = C_4 = 18pF$ for $X_1 > 5.0MHz$.

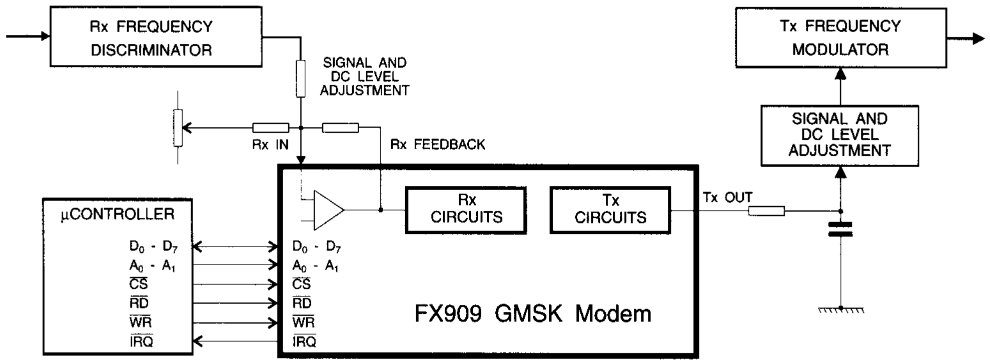
4. External capacitors C_6 and C_7 form part of the received signal level measuring circuit; the values of C_6 and C_7 should satisfy the following:
   C (F) x Data Rate (bps) = 120 x 10^{-6}.

D/Rate(kb/s)	C_6/C_7(nF)	D/Rate(kb/s)	C_6/C_7(nF)
4	30.0	4.8	22.0
8.0	15	9.6	12.0
16.0	6.8	19.2	6.8

5. If the on-chip Xtal oscillator is to be used, then the external components X_1, C_3, C_4, and R_3 are required as shown in Figure 3 (inset).
   If an external clock source is to be used these components are not required; the input should be connected to the Xtal/clock pin and the Xtal pin left unconnected.
   Table 4 (Clock/Data Rates) provides advice on the selection of the correct Xtal value.

External Signal Paths

The diagram below shows signal connections to and from the FX909. Inputs and outputs are shown with dc coupling and level-shifting components; the notes and diagrams on the following page (Figures 6 and 7) describe how, if acceptable, ac coupling may be employed.



g.4 External Signal Paths

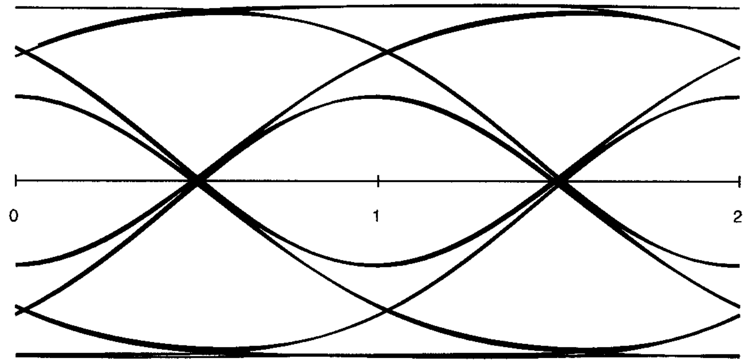


Fig.5 Example Transmit EYE Diagram (after the external RC network)

AC Coupling

For a practical application, ac coupling from the modem's transmit output to the Frequency Modulator and from the receiver's Frequency Discriminator to the receive input of the modem may be desired. There are, however, two problems.

Firstly, ac coupling of the signal degrades the bit-error-rate performance of the modem.
Figure 6 illustrates the typical bit error rates at 8kb/s (without FEC) for differing degrees of ac coupling;

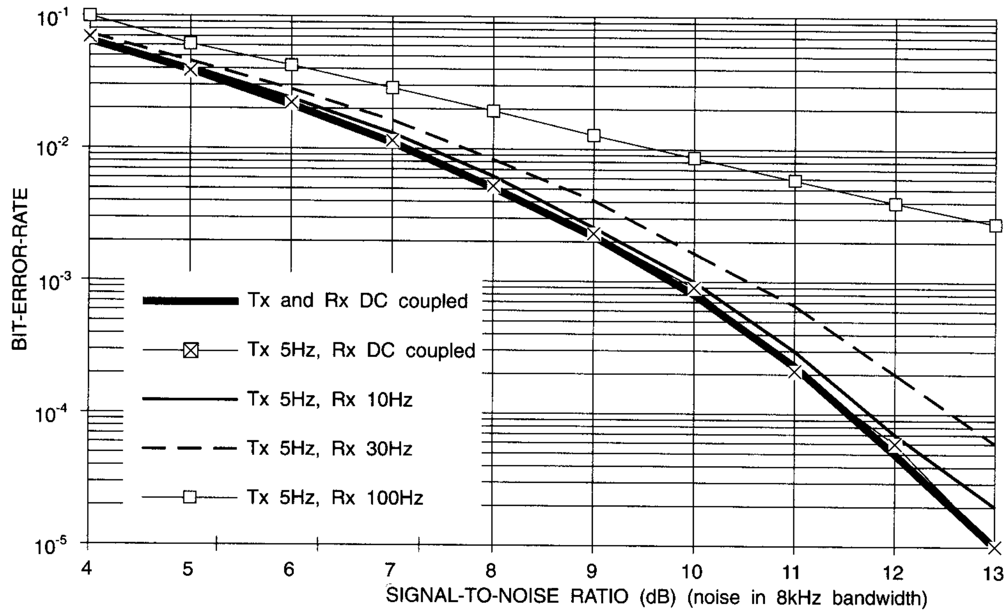


Fig.6 Examples of BER Performance Degradation Due to Varying Degrees of AC Coupling

Secondly, any ac coupling at the receive input will transform any step in the voltage at the discriminator output to a slowly decaying pulse which can confuse the modem's level measuring circuits.
As illustrated below, the time for this voltage step to decay to 37% of its original value is:

$$T = \frac{1}{(2\pi \times f)}$$

Where f is the 3dB cut-off frequency of the ac coupling network and is 8 msec (or 64 bit-times at 8000 bits-sec) for a 20Hz network.

For these reasons the maximum -3dB cut-off frequencies would seem to be around 5Hz in the Tx path and 20Hz in receive at 8kb/s.

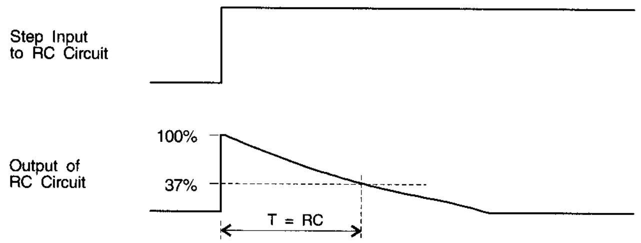


Fig.7 Decay Problems of the AC Coupling Network

Application Information ......

Radio Performance

The maximum data rate that can be transmitted over a radio channel using this modem depends on:

- RF channel spacing.
- Allowable adjacent channel interference.
- Bit rate.
- Peak carrier deviation (modulation index).
- Tx and Rx reference oscillator accuracies.
- Modulator and demodulator linearity.
- Receiver IF filter frequency and phase characteristics.
- Use of error correction techniques.
- Acceptable error rate.

As a guide, 8000 bits/sec can be achieved - subject to local regulatory requirements - over a system with 12.5kHz channel spacing if the transmitter frequency deviation is set to ± 2kHz peak for a repetitive ' 1100... ' pattern and the maximum difference between transmitter and receiver 'carrier' frequencies is less than 1500Hz.

The modulation scheme employed by this modem is designed to achieve high data throughput by exploiting as much as possible of the RF channel bandwidth. This does, however, place constraints on the performance of the radio.

In particular, attention must be paid to:

- Linearity, frequency and phase response of the Tx Frequency Modulator.
- The bandwidth and phase response of the receiver's IF filters.
- Accuracy of the Tx and Rx reference oscillators, as any difference will shift the received signal towards the skirts of the IF filter response and cause a dc offset at the discriminator output.

Viewing the received signal eye (at the FX909 Rx Feedback pin) gives a good indication of the overall transmitter/ receiver performance.

Modem to µController Interface

Data Bus Buffers and Address and Read/Write Decode blocks form a byte-wide parallel µController interface. The diagram below shows how this function can be memory mapped.

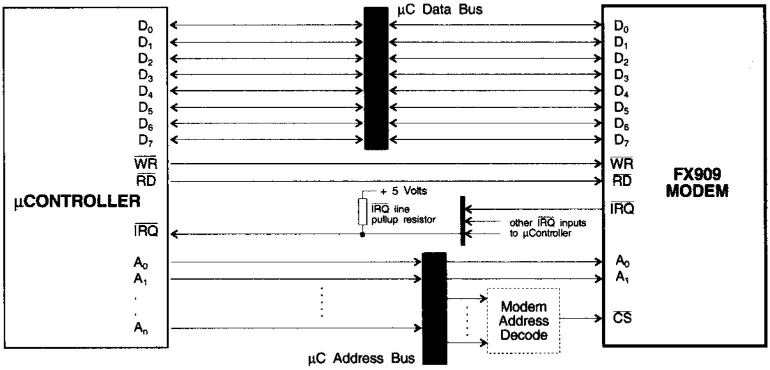


Fig.8 Typical Modem to µController Interface

Application Information ......

Baseband and RF Frequency Requirements

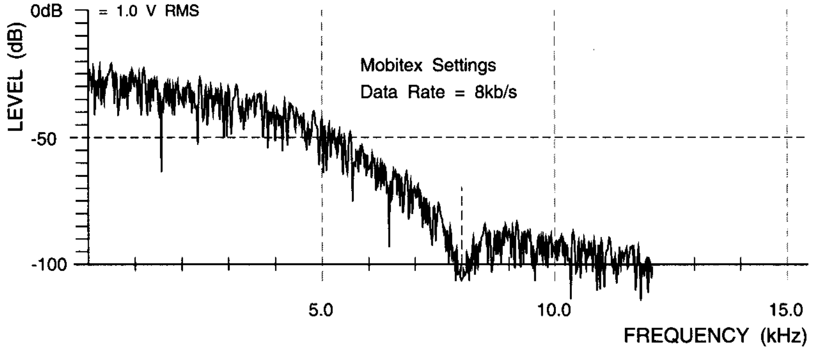


Fig.9 Typical Output Frequency Spectrum for a Random Data Sequence (Tx Out, measured after R_4, C_5 combination)

RF Channel Occupancy

The diagram below shows the theoretical RF bandwidth requirements when interfacing the FX909 baseband (Tx OUT) signal (Figure 9, above) to a radio transmitter. This plot assumes a perfect frequency modulator.

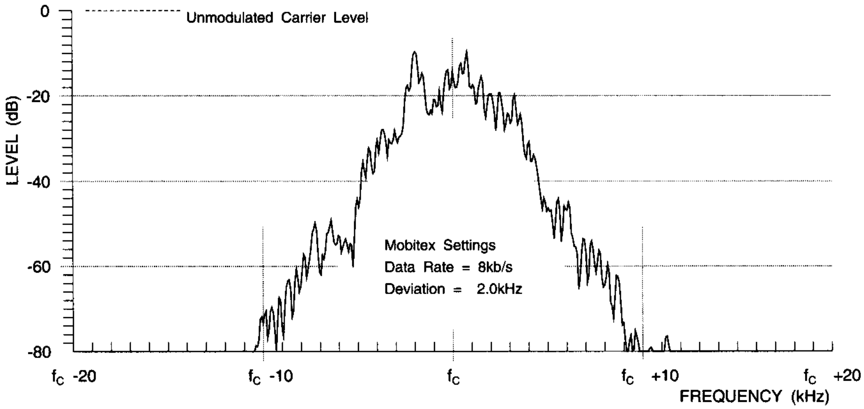


Fig.10 Theoretical Tx RF Frequency Spectrum Resulting from a Random Data Sequence from the FX909

Note that particular repetitive data sequences, such as '1010 ....' will produce spectra which are markedly different to those shown in Figures 9 and 10.

Programming Information

Data Formats

Mobitex Frame and Data Structures

The Mobitex format for transmitted data is in the form of a Frame Head immediately followed by a number of Data Blocks (0 to 32).

The Frame Head consists of 7 bytes ......

2 bytes of Bit Sync:
11001100 11001100 – from base, or
00110011 00110011 – from mobile
(sent L to R).

2 bytes of Frame Sync:
System specific.

2 bytes of Control Data:
– System specific ID and control information.

1 byte of FEC Code (generated by the FX909):
- 4 bits for each of the control bytes:
  – bits 7 - 4 operate on the first control byte.
  – bits 3 - 0 operate on the second control byte.

Each byte in the Frame Head is transmitted bit 7 (MSB) first, bit 0 (LSB) last.

The Data Block consists of ......

18 bytes of Data:
2 bytes of CRC are calculated by the FX909 from the 18 Data Bytes.
4 bits of FEC code are calculated for each of the Data and CRC bytes

The resulting 240 bits are interleaved and scrambled before transmission; see Figure 23, (Interleaving).

Figure 11 shows how the over-air signal is built up from Frame Sync and Bit Sync patterns, Control Bytes and Data Blocks.

The binary data transferred between the modem and the controlling µController is that shown enclosed in the heavily outlined rectangles.

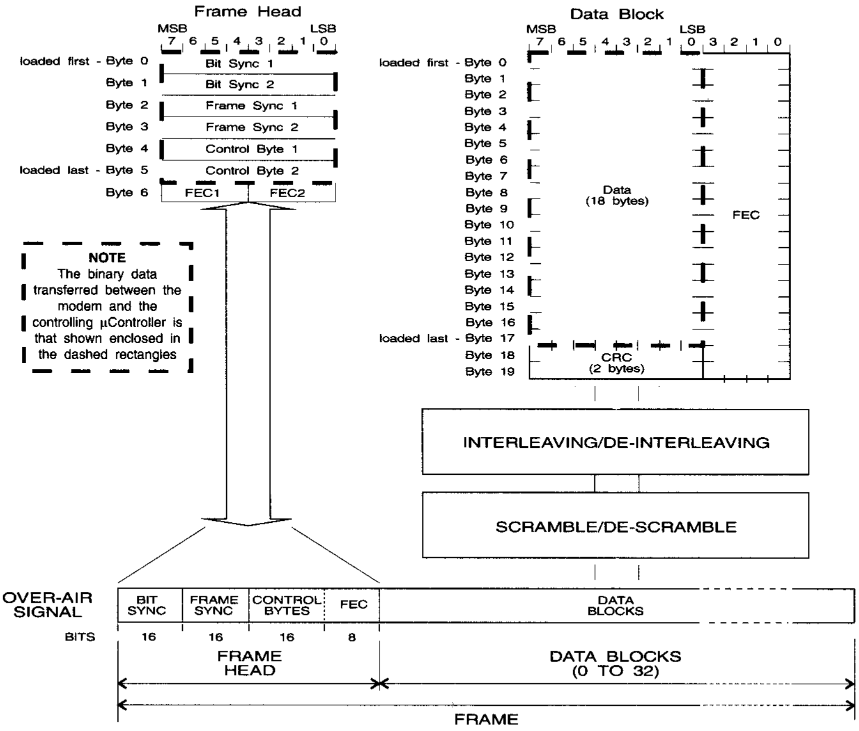


Fig.11 FX909 Mobitex System Data Format

General Purpose Formats

In a proprietary system the user may employ the data elements provided by this device to construct a custom, over-air data structure.

For example, 16 bits of bit sync + 2 bytes of frame sync + 4 bytes of receiver and sender address + n data blocks would be sent as: **TQB (bit and frame sync) + TQB (addresses) + (n x) TDB**.
And received as: **SFS + RSB + RSB + RSB + RSB + (n x) RDB**.

Note that it is important to have established frame synchronization before receiving data to enable the receiving device to decode synchronously. Also the user may add, by way of algorithms performed on the controlling device, additional data correction with the bytes in the data block task

Programming Information ......

Modem/µController Interaction

In general, data is transmitted over-air in the form of messages, or 'Frames', consisting of a 'Frame Head' optionally followed by one or more formatted Data Blocks.

The Frame Head includes a Frame Synchronisation pattern designed to allow the receiving modem to identify the start of a frame.

The following data blocks are constructed from the 'raw' data using a combination of CRC (cyclic redundancy checksum) generation, Forward Error Correction (FEC) coding, Interleaving and Scrambling.

To reduce the processing load on the host µController, the FX909 has been designed to perform as much as possible of the computationally intensive work involved in Frame formatting and de-formatting and, when in receive mode, in searching for and synchronising onto the Frame Sync or Frame Head.

In normal operation the modem will only require servicing by the µController once per received or transmitted data block. Thus, to transmit a block, the controlling µController has only to load the unformatted (raw) binary data into the modem's data buffer then instruct the modem to format and transmit that data. The modem will then calculate and add the CRC bits as required, encode the result with FEC coding, interleave then scramble the bits before transmission.

In receive mode, the FX909 modem can be instructed to assemble a block's worth of received bits, de-scramble and de-interleave the bits, check and correct (using the FEC coding) and check the resulting CRC before placing the received binary data into the Data Buffer for the µController to read. The FX909 modem can also handle the transmission and receptic of un-formatted data; to allow for example, the transmission of special Bit and Frame Synchronisation sequences or test patterns.

The Programmer's View

Register Selection

The FX909 modem appears to the programmer as 4 write-only 8-bit registers shadowed by 3 read-only registers.

Individual registers are selected by the A_1 and A_0 inputs; see Read and Write cycle timing diagrams (Figure 25).

Table 1 Register Selection

A_1	A_0	Write to Modem	Read from Modem
0	0	Data Buffer	Data Buffer
0	1	Command Register	Status Register
1	0	Control Register	DQ Register
1	1	Mode Register	not used

Data Buffer

An 18-byte read/write buffer which is used to transfer data (as opposed to Command, Status, Mode, Data-Quality and Control information) between the modem and the controlling µController.

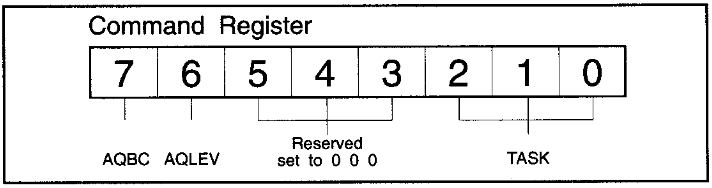
The Data Buffer appears to the µController as a single 8-bit register; the modem ensures that sequential µController 'read' or 'write' actions to the buffer are routed to the correct locations within this buffer.

The µController should only access this buffer when the Status Register BFREE (Buffer Free) bit is at a logi '1'. The buffer should only be written to while in the Tx mode and read from in the Rx mode (except when loading Frame Sync detection bytes in the Rx mode). Data Read/Write Timing information is provided in this document (Figure 25).

Command Register

Writing to this register instructs the modem to perform a specific action or actions, depending upon the setting of the TASK, AQLEV, and AQBC bits (see Figure 12 and Table 2).

Fig.12 The Command Register



When it has no action to perform, the modem will be in an idle state, and if it is in the Tx mode the input to the Tx (Lowpass) Filter will be connected to V_{BIAS}.

When it has no action to perform in the Rx mode the modem will continue to measure the received data quality and extract bits from the received signal, feeding them into the De-Interleave Buffer, but will otherwise ignore the received data.

Command Register

B7
AQBC

Acquire Bit Clock: This bit has no effect in the Tx mode.

In the Rx mode, whenever a byte with the AQBC bit set to logic '1' is written to the Command Register, it initiates an automatic sequence designed to achieve bit-timing synchronisation with the received signal as quickly as possible. This involves setting the Phase Locked Loop of the received bit-timing extraction circuits to their widest bandwidth, then gradually reducing the bandwidth as timing synchronisation is achieved, until it reaches the 'normal' value set by the PLLBW bits of the Control Register.

Setting this bit to logic '0' (or changing it from '1' to '0') has no effect, however note that the acquisition sequence will be re-started every time that a byte written to the Command Register has the AQBC bit set to logic '1'.

The AQBC bit will normally be set at the same time as an SFS (Search for Frame Sync) or SFH (Search for Frame Head) task, however it may also be used independently to re-establish clock synchronisation quickly after a long fade. Alternatively, an SFS or SFH task may be written to the Command Register with the AQBC bit at logic '0' if it is known that clock synchronisation does not need to be re-established.

More details of the Bit Clock Extraction Sequence are given in the Operational Information section of this Data Sheet

B6
AQLEV

Acquire Receive Signal Levels: This bit has no effect in the Tx mode.

In receive mode, whenever a byte with the AQLEV bit set to a logic '1' is written to the Command Register, it initiates an automatic sequence designed to measure the amplitude and dc offset of the received signal as rapidly as possible. This sequence involves setting the measurement circuits to respond quickly at first, then gradually increasing their response time -hence improving the measurement accuracy- until the 'normal' value set by the LEVRES bits of the Control Register is reached. See Figure 24.

Setting this bit to a logic '0' (or changing it from '1' to '0') has no effect; note that the acquisition sequence will be re-started every time that a byte written to the Command Register has the AQLEV bit set to a logic '1'.

The AQLEV bit will normally be set at the same time as an SFS (Search for Frame Sync) or SFH (Search for Frame Head) task is initiated, however it may also be used independently to re-establish signal levels quickly after a long fade. Alternatively, a SFS or SFH task may be written to the Command Register with the AQLEV bit at logic '0' if it is known that there is no need to re-establish the received signal levels. Refer to the Clock Extraction (Operational Information section) notes.

B5
B4
B3

These bits should each be set to a logic '0'.

B2
B1
B0
TASK

Task: Operations such as transmitting a data block are treated by the modem as 'tasks'. Information on Task functions is given on the following pages.
A task is initiated when the µController writes a byte to the Command Register with the Task bits set to anything other than the 'NULL' ('0' '0' '0') code.

The µController should not write a task (other than NULL or RESET) to the Command Register or write to or read from the Data Buffer if the BFREE (Buffer Free) bit of the Status Register is a logic '0'.

Different tasks apply in receive and transmit modes.

Tx Mode: All tasks other than NULL, RESET and TSO instruct the modem to transmit data from the Data Buffer, formatting it as required. For these tasks the µController should wait until the BFREE (Buffer Free) bit of the Status Register is a logic '1', before writing the data to the Data Buffer, then it should write the desired task to the Command Register. If more than 1 byte needs to be written to the Data Buffer, byte number '0' of the block should be written first.

Once the byte containing the desired task has been written to the Command Register, the modem will: Set the BFREE (Buffer Free) bit of the Status Register to a logic '0', take the data from the Data Buffer as quickly as it can -transferring it to the Interleave Buffer for eventual transmission. This operation will start immediately if the modem is 'idle' (i.e. not transmitting data from a previous task), otherwise it will be delayed until there is sufficient room in the Interleave Buffer.

Programming Information ......

Command Register ......

B2
B1
B0
TASK ......

Task: ...... Once all data has been transferred from the Data Buffer the modem will set the BFREE and IRQ bits of the Status Register to a logic '1', (causing the $\overline{\text{IRQ}}$ output to go low if the IRQEN bit of the Mode Register has been set to a logic '1') to tell the µController that it may write new data and the next task to the modem.

In this way the µController can write a task -and the associated data- to the modem while the modem is still transmitting the data from the previous task.

Rx Mode: The µController should wait until the BFREE bit of the Status Register is a logic '1', then write the desired task to the Command Register. Once the byte containing the desired task has been written to the Command Register, the modem will:

Set the BFREE bit of the Status Register to a logic '0'.

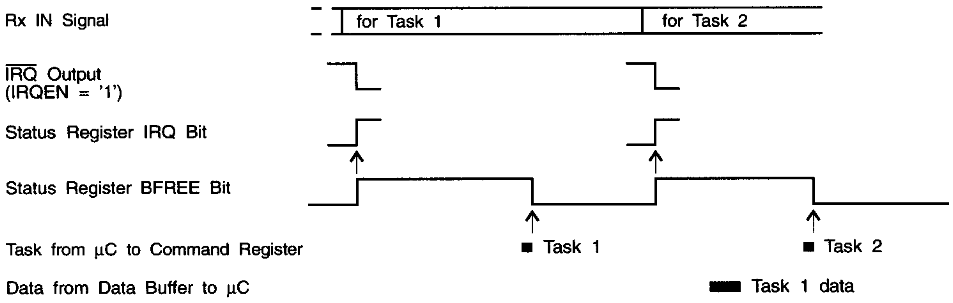
Wait until enough received bits are in the De-Interleave Buffer.

Decode them as needed, and transfer any resulting data to the Data Buffer.

Then the modem will set the BFREE and IRQ bits of the Status Register to logic '1', (causing the $\overline{\text{IRQ}}$ output to go low if the IRQEN bit of the Mode Register has been set to a logic '1') to tell the µController that it may read from the Data Buffer and write the next task to the modem. If more than 1 byte is contained in the Data Buffer, byte number '0' of the data will be read first.

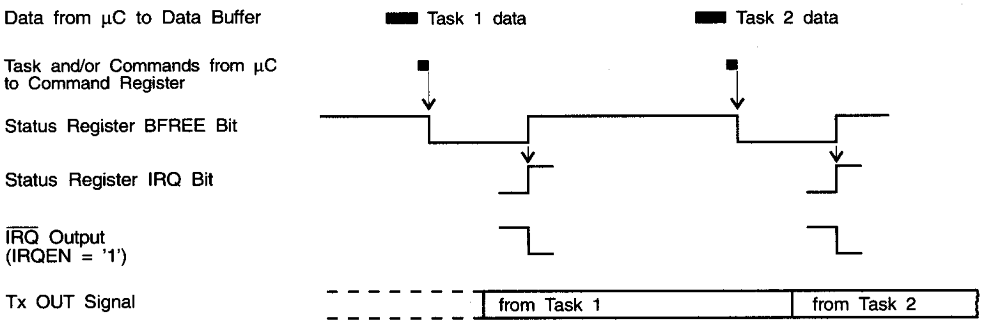
In this way the µController can read data and write a new task to the modem while the received bits needed for this new task are being stored in the De-Interleave Buffer. The above is not true for loading the Frame Sync detection bytes (LFSB); the bytes to be compared with the incoming data must be loaded prior to the task bits being written.

Detailed timings for the various tasks are given in later sections.



Rx IN Signal
for Task 1
for Task 2
$\overline{\text{IRQ}}$ Output (IRQEN = '1')
Status Register IRQ Bit
Status Register BFREE Bit
Task from µC to Command Register
■ Task 1
■ Task 2
Data from Data Buffer to µC
Task 1 data

Fig.13 The Receive Process



Data from µC to Data Buffer
Task 1 data
Task 2 data
Task and/or Commands from µC to Command Register
Status Register BFREE Bit
Status Register IRQ Bit
$\overline{\text{IRQ}}$ Output (IRQEN = '1')
Tx OUT Signal
from Task 1
from Task 2

Fig.14 The Transmit Process

Modem Tasks in Detail

The following describes the setting and format of the Command Register 'task' bits (bits 2, 1 and 0). Note that before a task is programmed the Tx/Rx bit in the Mode Register must be set to the required level.

Command Bits 2	1	0		Receive Mode		Transmit Mode
0	0	0	NULL		NULL	
0	0	1	SFH	Search for Frame Head	T7H	Transmit 7 Byte Frame Head
0	1	0	R3H	Read 3 Byte Frame Head	Reserved	.. ..
0	1	1	RDB	Read Data Block	TDB	Transmit Data Block
1	0	0	SFS	Search for Frame-Sync	TQB	Transmit 4 Bytes
1	0	1	RSB	Read Single Byte	TSB	Transmit Single Byte
1	1	0	LFSB	Load Frame-Sync Bytes	TSO	Transmit Scrambler Output
1	1	1	RESET	Cancel any Current Action	RESET	Cancel any Current Action

Table 2 Modem Task Allocations

Modem Tasks

NULL

No Effect. This task is provided so that an AQBC or AQLEV (Command Register) command can be initiated without loading a new task.

SFH

Search for Frame Head. Causes the modem to search the received signal for a valid Frame Head. The Frame Head will consist of a 16-bit Frame Sync followed by control data which has no uncorrectable errors (see Figure 11 -Data Format).

The search will continue until a valid Frame Head has been found, or until the RESET task is loaded.

The search is carried out by the modem in 3 stages:

1 Attempt to match the incoming bits against the previously programmed (task LFSB) 16-bit Frame Sync pattern (allowing up to any one bit (of 16) in error).

2 When a match has been found, the modem will read the next 3 received bytes as Frame Head bytes; these bytes will be checked using the FEC bits. If the FEC indicates uncorrectable errors the modem will resume the search, looking for a new Frame Sync pattern.

3 If the received bytes are error free or correctable, BFREE and IRQ bits (Status Register) are set to a logic '1' and the CRCFEC bit set to a logic '0'; the two corrected (by the modem) Frame Head Control Data bytes are then placed into the Data Buffer.
The MOBAN bit (Mobile or Base) in the Status Register will be set according to the polarity of the 3 bits that preceded the Frame Sync pattern.

On detecting that the BFREE bit of the Status Register has gone to a logic '1', the µController should read the 2 Frame Head control data bytes from the Data Buffer and then write the next task to the modem's Command Register.

R3H

Read 3 Byte Frame Head. This task, which would normally follow an SFS task, will cause the modem to place the next 3 bytes directly into the Data Buffer and, concurrently, check those 3 bytes as Frame Head Control Data bytes; the modem will set the CRCFEC bit to a logic '1' (high) if errors are detected. Note: This task will not correct any errors.

The BFREE and IRQ bits of the Status Register will be set to a logic '1' when the task is complete; this is to indicate that the µController may read the data from the Data Buffer and write the next task to the Command Register. The CRCFEC bit in the Status Register will be set according to the validity of the received FEC bits.

RDB

Read Data Block. Causes the modem to read the next 240 bits (see Data Formats -Frame and Data Structures) as a Mobitex data block. This task will de-scramble and de-interleave the received bits, FEC correct and CRC check the resulting 18 data bytes placing them in the Data Buffer. When the task is complete the BFREE and IRQ bits of the Status Register are set to a logic '1' to indicate that the µController may read the data from the Data Buffer and write the next task to the Command Register. The CRCFEC bit in the Status Register will be set according to the outcome of the CRC check. Note that in the receive mode the checksum circuits are initialised (ready for operation) on completion of any task other than NULL.

Programming Information ......

Modem Tasks ......

SFS

Search for Frame Sync. This task performs the first part only of an SFH task. It causes the modem to search the received signal for a 16-bit sequence which matches the previously programmed Frame Sync pattern (allowing up to any one bit (in 16) in error). When a match is found the modem will set the BFREE and IRQ bits of the Status Register to a logic '1' and update the MOBAN bit. The µController may then write the next task to the Command Register.

RSB

Read Single Byte. This task, which is intended for special tests and channel monitoring -perhaps preceded by an SFS task, causes the modem to read the next 8 bits and translate them directly (without de-interleaving or FEC) to an 8-bit byte which is placed into the Data Buffer (bit 7 will represent the earliest bit received). The BFREE and IRQ bits of the Status Register will then be set to a logic '1' to indicate that the µController may read the data byte from the Data Buffer and write the next task to the Command Register.

LFSB

Load Frame Sync Bytes. This task is unlike other Rx tasks in that the Data Buffer must be loaded (with the 2 Frame Sync bytes) before the task is issued and the task must only be issued 'between' received messages ; i.e. before the first task for receiving a message and after the last data is read out of the Data Buffer.

It takes 2 bytes from the Data Buffer and loads them into the FX909's internal Frame Sync pattern store. The MSB of byte 0 represents the first bit of a received Frame Sync pattern and the LSB of byte 1 is compared to the last bit of a received Frame Sync pattern that will be looked for when a SFS or SFH task is executing. The LFSB task itself does not initiate a search for a received Frame Sync pattern.

Once the modem has read the Frame Sync bytes from the Data Buffer, the BFREE and IRQ bits of the Status Register will be set to a logic '1', indicating that the µController may write the next task to the modem.

T7H

Transmit 7-Byte Frame Head. Takes 6 bytes of data from the Data Buffer, calculates and appends 8 bits of FEC from bytes 4 and 5 then transmits the result as a complete Frame Head. Bytes 0 and 1 form the bit -sync pattern, bytes 2 and 3 form the frame-sync pattern and bytes 4 and 5 are the Frame Head control bytes. Bit 7 of byte 0 of the Data Buffer is sent first and bit 0 of the FEC byte last.

Once the modem has read the data bytes from the Data Buffer, the BFREE and IRQ bits of the Status Register will be set to a logic '1', indicating that the µController may write the next task and its data to the modem.

TQB

Transmit 4 Bytes. Takes 4 bytes of data from the Data Buffer and transmits them without adding FEC or interleaving, bit 7 of byte 0 first, bit 0 of byte 3 last.

Once the modem has read the data bytes from the Data Buffer, the BFREE and IRQ bits of the Status Register will be set to a logic '1', indicating that the µController may write the next task and its data to the modem.

TDB

Transmit Data Block. Takes 18 bytes of data from the Data Buffer, calculates and applies a 16-bit CRC and forms the FEC for the 18 data bytes and the CRC; the resulting 240 bits are then interleaved and passed through the scrambler, if enabled, before being transmitted as a Data Block.

Note that in transmit mode the CRC checksum circuit is initialised on completion of any task other than NULL.

Once the modem has read the data bytes from the Data Buffer, the BFREE and IRQ bits of the Status Register will be set to a logic '1', indicating that the µController may write the next task data to the modem.

TSO

Transmit Scrambler Output. Intended for channel set-up, this task enables the scrambler and transmits its output (which will be 9-bit pseudo-random). When the modem has started this task the Status Register bits will not be changed and hence an IRQ will not be raised. The µController may write data and the next task to the modem at any time and the scrambler output will stop when the new task has produced its first data. See Mode Register SCREN.

RESET

Stop any Current Action. This 'task' takes effect immediately, and terminates any current action (Task, AQBC or AQLEV) the modem may be performing and sets the BFREE bit of the Status Register to a logic '1', without setting the IRQ bit.

RESET should be used when V_{DD} is applied to set the modem into a known state.

Note that due to delays in the Tx Lowpass Filter filter, it will take approximately 2 bit-times for any change to become apparent at the Tx Out pin.

Programming Information ......

Modem Tasks ......

TSB	**Transmit Single Byte.** Takes a byte from the Data Buffer and transmits the 8 bits, bit 7 first without adding FEC or interleaving. Once the modem has read the data byte from the Data Buffer, the BFREE and IRQ bits of the Status Register will be set to a logic '1', indicating to the µController that it may write the next task and its data to the modem.

Lowpass Filter Delay

The Task Timing figures detailed in Table 3 are based upon: the signal at the input to the Tx lowpass Filter in the transmit mode, or the signal at the input to the de-interleave circuits in the receive mode. As can be seen from the diagram in Figure 15, there is an additional delay of approximately 2 (two) bit-times in both Tx and Rx modes due to the (Tx/Rx) Lowpass Filter.

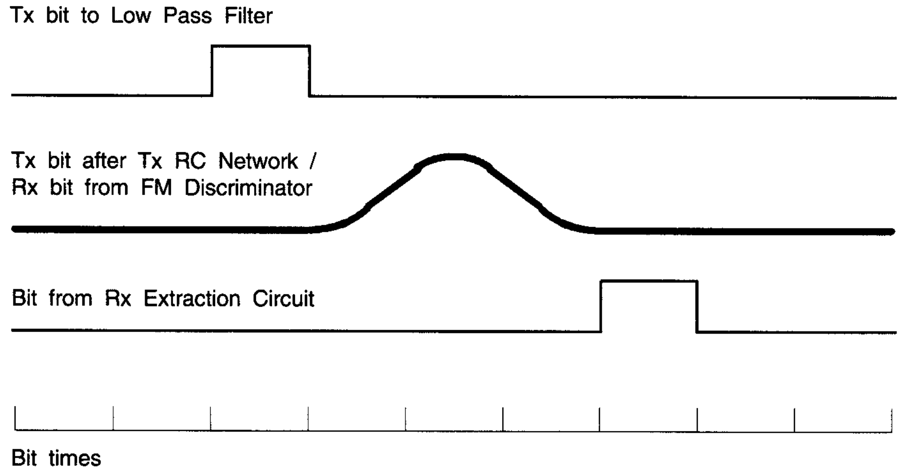

Tx bit to Low Pass Filter
Tx bit after Tx RC Network /
Rx bit from FM Discriminator
Bit from Rx Extraction Circuit
Bit times

Fig. 15 Example of Lowpass Filter Delay Times

Transmit and Receive Task Timing

The diagrams and table on the following page describe Rx and Tx task timing requirements. Note that a new task should not be written to the Command Register for at least 2 bit times after the following operations:
Changing the Rx/Tx bit.
Resetting or after power is applied to the FX909.
This precaution is to ensure that the internal operation of the device is initialised correctly for the new task.
This precaution only applies to the Command Register; other registers may be accessed as normal.

Programming Information ......

Transmit and Receive Task Timing ......

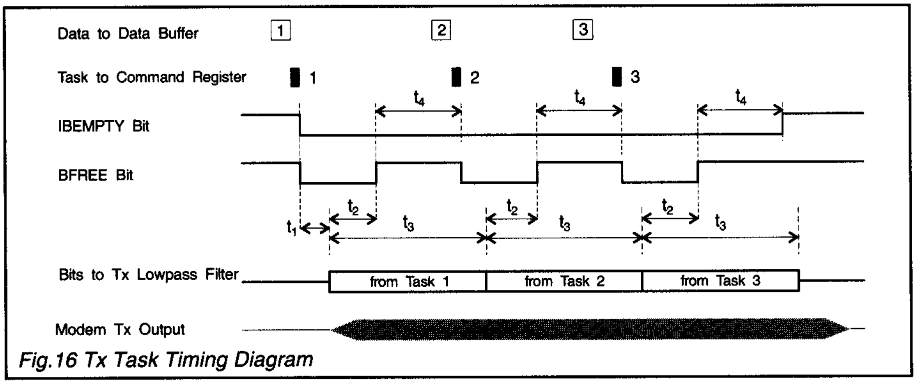


Fig.16 Tx Task Timing Diagram

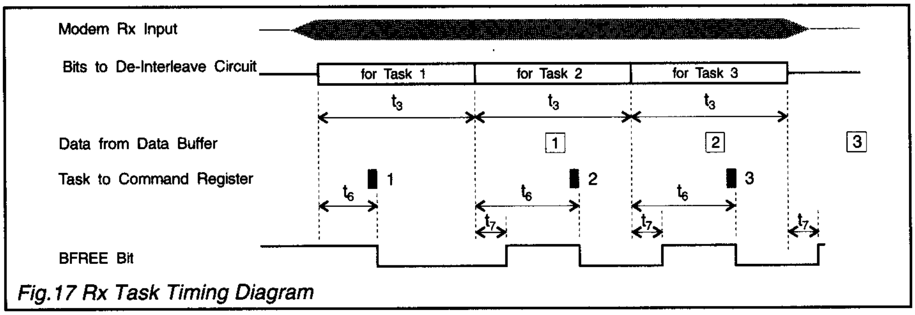


Fig.17 Rx Task Timing Diagram

Timing	Notes	Task	Typical (Bit) Time
t_1	Modem in idle state. Time from writing first task to the application of the first Tx bit to the Tx Lowpass Filter.	Any	1
t_2	Time from the application of the first bit of the task to the Tx Lowpass Filter until BFREE goes to a logic '1' (high).	T7H TQB TDB TSB	36 24 20 1
t_3	Time to transmit all bits of the task. or Time to receive all bits of the task	T7H/SFH TQB R3H TDB/RDB TSB/RSB	56 32 24 240 8
t_4	Maximum time allowed from BFREE going to a logic '1' for the next task (and data) to be written to the modem.	T7H TQB TDB TSB	18 6 218 6
t_6	Maximum time between the first bit of the task entering the de-interleave circuit and the task being written to the modem.	SFH R3H RDB RSB	14 18 218 6
t_7	Time from last bit for task entering the de-interleave circuit to BFREE going to a logic '1'.	Any	1

Table 3 Typical Rx/Tx Task Load Timings

Programming Information ......

Control Register

This 8-bit write-only register controls the modem's bit-rate, response times of the receive clock extraction and signal level measurement circuits and the internal analogue filters.

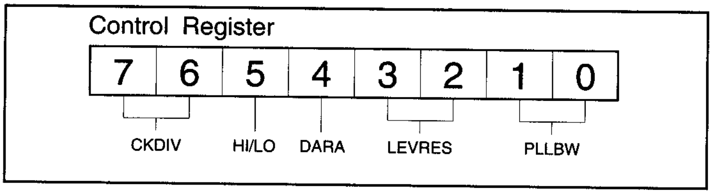


Fig.18 The Control Register

Control Register

Table 4 shows how bit-rates of 4000/8000/16000 or 4800/9600/19200 bits per second may be obtained from common Xtal/clock frequencies.

B7, B6 CKDIV — **Clock Division Ratio:** These bits, together with the HI/LO bit, control a frequency divider driven from the Xtal/clock signal; this ratio and the frequency at the Xtal/Clock pin will determine the transmit and nominal receive bit-rate.

B5 HI/LO — **High or Low Xtal Range Selection:**

		B5		**Xtal/Clock Frequency (MHz)**					
		‘1’	High	8.192	9.8304	4.096 (12.288/3)	4.9152	2.048 (6.144/3)	2.4576 (12.288/5)
			‘0’ Low	4.096 (12.288/3)	4.9152	2.048 (6.144/3)	2.4576 (12.288/5)	1.024	1.2288
B7	**B6**	**Division Ratio:** **Xtal/Clock** **Data Rate**		**Data Rate (bits per second)**					
0	0	256	128			16000	19200	8000	9600
0	1	512	256	16000	19200	8000	9600	4000	4800
1	0	1024	512	8000	9600	4000	4800		
	1	2048	1024	4000	4800				

Table 4 Clock/Data Rates ***Note*** *that device operation is not guaranteed or specified above 19,200 bits/s or below 4,000 bits/s*

B4 DARA — **Data Rate:** Employed in both Rx and Tx, this bit optimises the modem's internal signal filtering circuitry to the relevant bit-rate.
For bit-rates above 10 kb/s this bit (B4) should be set to a logic ‘1’, for bit-rates at or below 10kb/s set to a logic ‘0’.

B3, B2 LEVRES — **Level Measurement Response Time:** These bits are only used in the Rx mode and have no effect in the Tx mode; they set the ‘normal’ response time of the Rx signal amplitude and dc offset measuring circuits. This setting will be temporarily overriden by the automatic sequence of an AQLEV command. See Table 5.
For Mobitex systems, and most general-purpose applications using this modem, these bits should be set to ‘Peak Averaging’, except when the µController detects a receive signal fade, when ‘Hold’ should be selected.

Continued » » »

Programming Information ......

Control Register......

B3, B2 ......
LEVRES

LEVRES ......: The 'Lossy Peak Detect' setting is intended for systems where the µController cannot detect signal fades or the start of a received message; this setting allows the modem to respond quickly to fresh messages and recover rapidly after a fade without µController intervention -this however will be at the cost of reduced Bit-Error-Rate vs Signal-to-Noise performance.

Note that as the measured levels are stored on capacitors C_6 and C_7 via pins Doc 1 and Doc 2, these levels will decay gradually towards V_{BIAS} when the 'Hold' setting is used; the discharge time-constant is approximately 2000 bit-times.

Table 5 details bit-setting application.

B3	B2	Setting	Action
0	0	Hold	Keep current values of amplitude and offset
0	1	Peak Averaging	Track input signal using bit peak averaging
1	0	Peak Detect	Track input signal using peak detection
1	1	Lossy Peak Detect	Track input signal using lossy peak detection

Table 5

B1, B0
PLLBW

PLL Bandwidth: For use in the Rx mode only (no effect in Tx).

In the receive mode these two bits set the 'normal' bandwidth of the Rx Clock Extraction phase locked loop circuit to allow for Rx and Tx Xtal tolerances. This setting will be temporarily overridden by the automatic sequence of an AQBC (Command Register Bit 7) command.

B1	B0	PLL Bandwidth	Suggested Use
0	0	Hold	Signal fades
0	1	Narrow	±20ppm or better Xtals
1	0	Medium	Wide tolerance Xtals or long preamble acquisition
1	1	Wide	Quick acquisition

Table 6

The 'Hold' setting is intended for use during signal fades, otherwise the minimum bandwidth consistent with the Rx and Tx modem bit-rate tolerances should be chosen.

The wide and medium bandwidth settings are intended for systems where the µController cannot detect signal fades or the start of a receive message; as they allow the modem to respond rapidly to fresh messages and recover rapidly after a fade without µController intervention. This action however is at the expense of reduced Bit-Error-Rate vs Signal-to-Noise performance.

Programming Information ......

Mode Register

This 8-bit write-only register controls the basic operating modes of the modem.

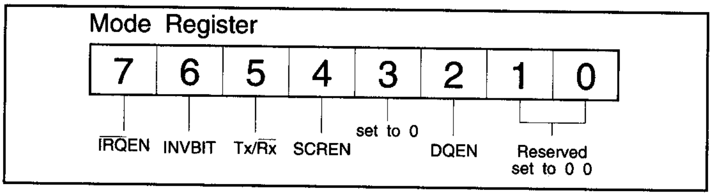


Fig.19 The Mode Register

Mode Register

B7
$\overline{\text{IRQ}}$ EN

$\overline{\text{IRQ}}$ Output Enable: When set to a logic '1' the Interrupt Request output will be pulled low (to V_{SS}) whenever the $\overline{\text{IRQ}}$ bit (BIT 7) of the Status Register is set by the modem to a logic '1'. When set to a logic '0' the Interrupt Request output will not function and will remain in its high-impedance state (see Pin Functions Page).

B6
INVBIT

Invert Bits: When set to a logic '1', all data (sense) voltages to and from the modem's Rx and Tx paths are inverted.

For example:

B6	**Tx/Rx Logic '1'**	**Tx/Rx Logic '0'**
'0'	High (above V_{BIAS})	Low (below V_{BIAS})
'1'	Low (below V_{BIAS})	High (above V_{BIAS})

Data will be affected immediately after B6 is set and so this bit should not be changed whilst the modem is decoding or transmitting data. This bit only operates on data bits, there is no effect upon functional logic inputs.

B5
Tx/Rx

Tx/$\overline{\text{Rx}}$ Mode: When set to a logic '1' places the modem in the Transmit mode; when set to a logic '0' places the modem in the Receive mode. To allow the lowpass filter to stabilize, when changing from Rx to Tx there must be a 2 bit pause before setting a new task.
Note that changing between Transmit and Receive modes will cancel any current task.

B4
SCREN

Scramble Enable: Setting this bit to a logic '1' enables data scrambling; setting it to a logic '0' disables scrambling. The scrambler only takes effect during the transmission or reception of a Mobitex Data Block (see Figure 11 -System Data Format) and during TSO (Transmit Scrambler Output) task.
The scrambler is only operative, if enabled by this mode bit (B4), during TSO, RDB or TDB (see Modem Tasks), it is held in the reset state at all other times. This bit should not be changed whilst the modem is decoding or transmitting a Data Block.

B3

This bit should be set to a logic '0'.

B2
DQEN

Data Quality $\overline{\text{IRQ}}$ Enable: For use in the Rx mode only (no effect in Tx).
In the Rx mode, setting this bit to a logic '1' causes the $\overline{\text{IRQ}}$ bit (of the Status Register) to be set to a logic '1' whenever a new Data Quality reading is ready; the DQRDY bit of the Status Register will also be set to a logic '1' at the same time.

B1
B0

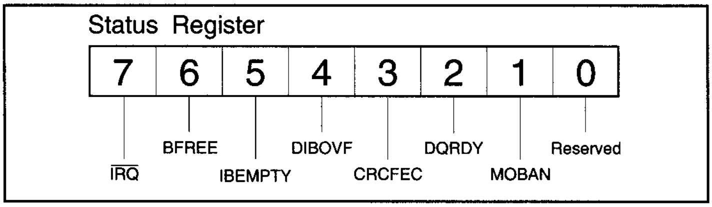
These bits should be set to a logic '0'.

Programming Information ......

Status Register

This register may be read by the µController to determine the current state of the modem.

Fig.20 The Status Register



Status Register

B7
IRQ

Interrupt Request: This bit is set to a logic '1' by:

The Status Register BFREE bit going from a logic '0' to '1', unless this transition is caused by a RESET Task or by a change to the Mode Register's Tx/Rx bit.

or

The Status Register IBEMPTY bit going from a logic '0' to '1', unless this transition is caused by a RESET Task or by a change to the Mode Register's Tx/Rx bit.

or

The Status Register DQRDY bit going from a logic '0' to '1' (if DQEN = '1').

or

The Status Register DIBOVF bit going from a logic '0' to '1'.

This (IRQ) bit is cleared to a logic '0' immediately after a read of the Status Register.
If the IRQEN bit of the Mode Register is a logic '1', the FX909 $\overline{IRQ}$ output pin will be pulled low (to V_{SS}) whenever the Status Register IRQ bit is a logic '1'.

B6
BFREE

Data Buffer Free: BFREE reflects the availability of the Data Buffer; BFREE is cleared to a logic '0' *(Buffer NOT Free)* whenever a task other than NULL, RESET or TSO is written to the Command Register.
In Transmit mode, the BFREE bit will be set to a logic '1' (setting the Status Register IRQ bit to a logic '1) when the modem is ready for the µController to write new data to the Data Buffer and the next task to the Command Register.
In Receive mode, the BFREE bit is set to a logic '1' (setting the Status Register IRQ bit to a logic '1) when it has completed a task and any data associated with that task has been placed into the Data Buffer. The µController may then read that data and write the next task to the Command Register.
The BFREE bit is also set to a logic '1' (but without setting the IRQ bit) by a RESET task or when the Mode Register Tx/Rx bit is changed.

B5
IBEMPTY

Interleave Buffer Empty: In Transmit mode, IBEMPTY is set to a logic '1' (also setting the IRQ bit) when less than two bits remain in the Interleave Buffer. Any transmit task written to the modem after IBEMPTY goes to a logic '1' will be too late to avoid a gap in the transmit output signal (see Figure 16 and Table 3, Tx Task Timing)
IBEMPTY is also set to a logic '1' by a RESET task and by a change of the Mode Register's Tx/Rx bit, but in this case the IRQ bit will not be set.
IBEMPTY is cleared to a logic '0' by writing a task other than NULL, RESET or TSO to the Command Register.
Note that when the modem is in the transmit mode and the Interleave Buffer is empty, a mid-level (V_{BIAS}) voltage will be fed to the Tx Lowpass Filter.
In Receive mode this bit is a logic '0'.

Programming Information ......

Status Register ......

B4
DIBOVF

De-Interleave Buffer Overflow: In Receive mode DIBOVF is set to a logic '1' (also setting the IRQ bit) when a task is written to the Command Register too late to allow continuous reception (see Figure 17 and Table 3 Rx Task Timing).
DIBVOF is cleared to a logic '0' by reading the Status Register, by writing a RESET task to the Command Register or by changing the Tx/Rx bit of the Mode Register.
In Transmit mode this bit is a logic '0'.

B3
CRCFEC

CRC or FEC Error: In Receive mode CRCFEC will be updated at the end of every Data Block task, after checking the CRC, and at the end of receiving Frame Head control bytes, after checking the FEC.
A logic '0' indicates that the CRC was received correctly or that the FEC found no uncorrectable errors.
A logic '1' indicates that errors are present.
CRCFEC is cleared to a logic '0' by a RESET task, or by changing the Tx/Rx bit of the Mode Register.
In Transmit mode this bit is a logic '0'.

B2
DQRDY

Data Quality Reading Ready: In Receive mode DQRDY is set to a logic '1' whenever a Data Quality reading has been completed (see Figure 21 Data Quality graph).
DQRDY is cleared to a logic '0' by a read of the Data Quality Register, or by changing the Tx/Rx bit of the Mode Register.

B1
MOBAN

Mobile or Base Bit-Sync Received: In Receive mode the MOBAN bit is updated at the end of the SFS and SFH tasks. MOBAN is set to a logic '1' whenever the 3 bits immediately preceding a detected Frame Sync are '0' '1' '1' (received left to right), with up to any one bit in error.
MOBAN is set to a logic '0' if the bit pattern is '1' '0' '0', again with up to any one bit in error. Thus if this bit is set to a logic '1' then the received message is likely to have originated from a mobile station, and if set to a logic '0' the call is likely to have originated from a base station.
The Data Formats section of this document describes the different mobile and base sync structures.
In Transmit mode this bit is a logic '0'.

B0

This bit is always set to a logic '0'.

The Data Quality Register

The information presented in this 8-bit register is intended to indicate the quality of the receive signal during a Data Block or 30 single bytes.

In Receive Mode, the modem measures the quality of the received signal by comparing the actual received zero-crossing time against an internally generated time. This value is averaged over 240 bits and at the end of the measurement the Data Quality Register and the Data Quality Reading Ready (DQRDY) bit in the Status Register are updated. An interrupt will only occur at this time if the DQEN bit in the Mode Register = logic '1'.

In Transmit Mode all bits are set to a logic '0'.

To provide synchronisation with Data Blocks, and hence ensure that the DQ Register is updated ready to be read when the RDB task finishes, the measurement process is reset at the end of Tasks SFH, SFS, RDB and R3H.

Figure 21 shows how the value (0 - 240) read from the Data Quality Register varies with received signal-to-noise ratio.

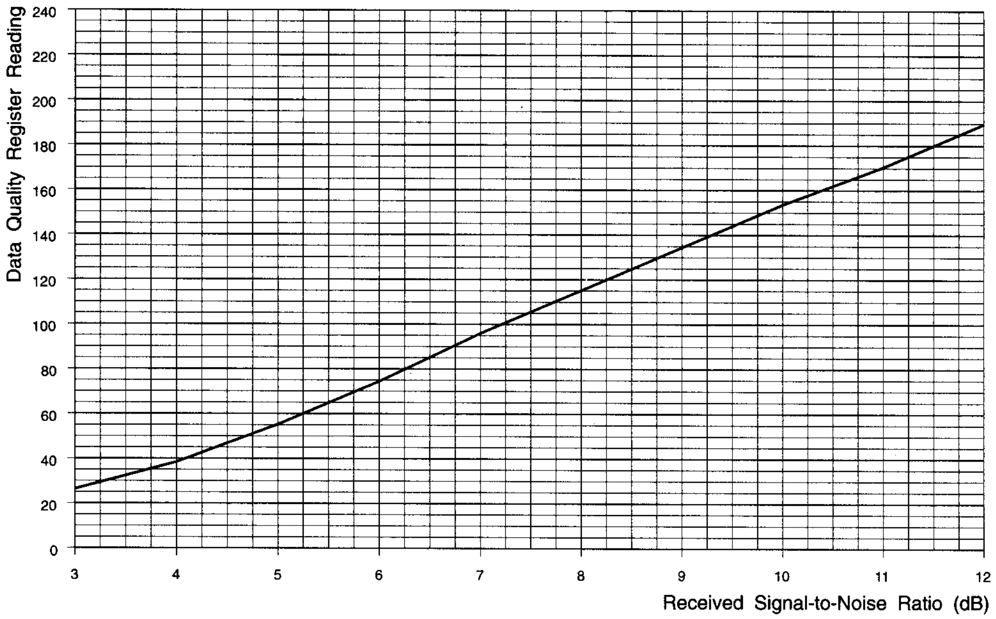
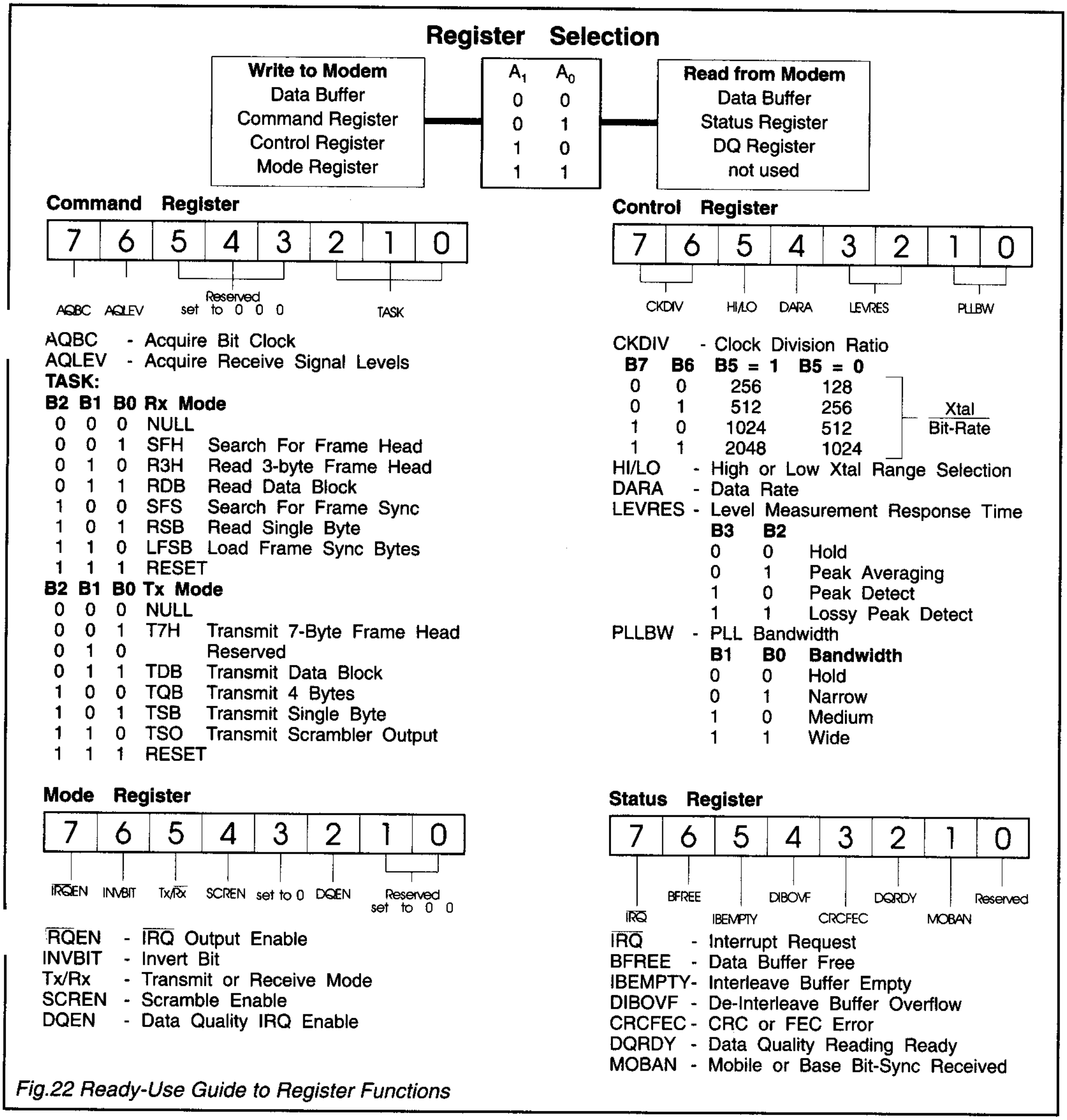


Fig.21 Typical Data Quality Reading vs Rx Signal-to-Noise Ratio (calculated for noise in a bit-rate bandwidth)

Programming Information ......

FX909 Registers

This diagram may provide a useful quick-reference to FX909 register allocations.



Register Selection

Write to Modem	A_1	A_0	Read from Modem
Data Buffer	0	0	Data Buffer
Command Register	0	1	Status Register
Control Register	1	0	DQ Register
Mode Register	1	1	not used

Command Register

AQBC - Acquire Bit Clock
AQLEV - Acquire Receive Signal Levels

TASK:

B2	**B1**	**B0**	**Rx Mode**	
0	0	0	NULL	
0	0	1	SFH	Search For Frame Head
0	1	0	R3H	Read 3-byte Frame Head
0	1	1	RDB	Read Data Block
1	0	0	SFS	Search For Frame Sync
1	0	1	RSB	Read Single Byte
1	1	0	LFSB	Load Frame Sync Bytes
1	1	1	RESET	

B2	**B1**	**B0**	**Tx Mode**	
0	0	0	NULL	
0	0	1	T7H	Transmit 7-Byte Frame Head
0	1	0		Reserved
0	1	1	TDB	Transmit Data Block
1	0	0	TQB	Transmit 4 Bytes
1	0	1	TSB	Transmit Single Byte
1	1	0	TSO	Transmit Scrambler Output
1	1	1	RESET	

Control Register

CKDIV - Clock Division Ratio

B7	**B6**	**B5 = 1**	**B5 = 0**	
0	0	256	128	Xtal / Bit-Rate
0	1	512	256	
1	0	1024	512	
1	1	2048	1024	

HI/LO - High or Low Xtal Range Selection
DARA - Data Rate
LEVRES - Level Measurement Response Time

B3	**B2**	
0	0	Hold
0	1	Peak Averaging
1	0	Peak Detect
1	1	Lossy Peak Detect

PLLBW - PLL Bandwidth

B1	**B0**	**Bandwidth**
0	0	Hold
0	1	Narrow
1	0	Medium
1	1	Wide

Mode Register

$\overline{\text{IRQ}}$EN	-	$\overline{\text{IRQ}}$ Output Enable
INVBIT	-	Invert Bit
Tx/$\overline{\text{Rx}}$	-	Transmit or Receive Mode
SCREN	-	Scramble Enable
DQEN	-	Data Quality IRQ Enable

Status Register

IRQ	-	Interrupt Request
BFREE	-	Data Buffer Free
IBEMPTY	-	Interleave Buffer Empty
DIBOVF	-	De-Interleave Buffer Overflow
CRCFEC	-	CRC or FEC Error
DQRDY	-	Data Quality Reading Ready
MOBAN	-	Mobile or Base Bit-Sync Received

Fig.22 Ready-Use Guide to Register Functions

Operational Information

Operation Details

Cyclic Redundancy Code (CRC)

A 16-bit CRC code is used in the Mobitex Data Block.

In Transmit Mode the CRC is calculated by the modem from the 18 data bytes (see Figure 11 Mobitex System Data Format) using the following generator polynomial:

$$g(x) = x^{16} + x^{12} + x^{5} + 1 \qquad \text{[CCITT CRC-16]}$$

This code detects all (single) error bursts of up to 16 bits in length and about 99.998% of all other error patterns.

The CRC Register is intialised to all logic '1's and the CRC is calculated octet by octet starting with the LSB of byte 0. The CRC calculated is bit-wise inverted and appended to the data bytes with the MSB transmitted earliest.

In Receive Mode a 16-bit CRC code is generated from the 18 data bytes of each Mobitex Data Block as described above and the bit-wise inverted value is compared with the received CRC bytes, if a mis-match is present then an error has been detected.

Forward Error Correction (FEC)

In Transmit Mode, during T7H and TDB, the modem generates a 4-bit Forward Error Correction code for each coded byte.

The FEC is defined by the following H matrix:

$$\begin{array}{c|cccccccccccc|}
 & \multicolumn{8}{c}{\textit{DATA BYTE}} & \multicolumn{4}{c}{\textit{FEC}} \\
 & \text{MSB} & & & & & & & \text{LSB} & \text{MSB} & & & \text{LSB} \\
 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 3 & 2 & 1 & 0 \\
\hline
 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
H = & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & \\
\hline
\end{array}$$

Generation of the FEC consists of logically ANDing the byte to be transmitted with bits 7 to 0 of each row of the H matrix. Even parity is generated for each of the 4 results and these 4 parity bits, in the positions indicated by the last 4 columns of the H matrix, form the FEC code.

Receive Mode: In checking the FEC the received 12-bit word is logically ANDed with each row of the H matrix (earliest bit received compared with the first column). Again even parity is generated for the 4 resulting words and these parity bits form a 4-bit nibble. If this nibble = 0 then no errors have been detected. Other results 'point' to the bit in error or indicate that uncorrectable errors have occured.

This code can correct any single error that has occured in each 12-bit (8 data + 4 FEC) section of the message.

Example of FEC Generation

If the byte to be coded is **'0 0 1 0 1 1 0 0'**, then the FEC is derived as follows:

H Matrix Row	**1**	**2**	**3**	**4**
A	1 1 1 0 1 1 0 0	1 1 0 1 0 0 1 1	1 0 1 1 1 0 1 0	0 1 1 1 0 1 0 1
B	**0 0 1 0 1 1 0 0**	**0 0 1 0 1 1 0 0**	**0 0 1 0 1 1 0 0**	**0 0 1 0 1 1 0 0**
A 'AND' B	0 0 1 0 1 1 0 0	0 0 0 0 0 0 0 0	0 0 1 0 1 0 0 0	0 0 1 0 0 1 0 0
Even Parity	1	0	0	0

With reference to the table above, Row A is bit 7 to 0 of one row of the H matrix and Row B is the byte to be coded. The Even Parity bits refer to the result of 'A' AND 'B'.

Therefore the word formed will be: **'0 0 1 0 1 1 0 0   1 0 0 0'** -sent left to right.

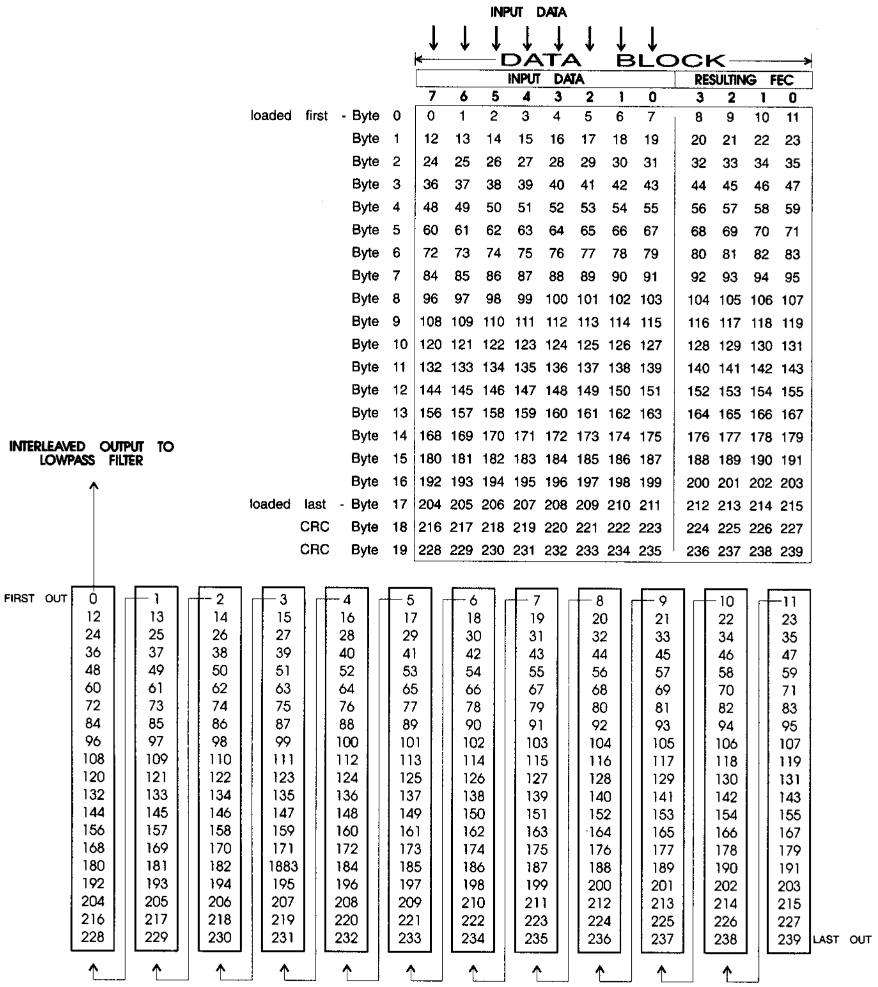
When the same process is carried out on these 12 bits (**'0 0 1 0 1 1 0 0   1 0 0 0'**), using all 12 bits of each H matrix row, the resulting parity bits will be **'0 0 0 0'.**

Operation Details ......

Interleaving

The 240 bits of a Data Block are interleaved by the modem before transmission to give protection against noise bursts and short fades. Interleaving is not performed on any bits in the Frame Head.

Considering the 240 bits to be numbered sequentially before interleaving as 0 to 239 (0 = bit 7 of byte 0, 11 = bit 0 of the FEC for byte 0, and .... ,239 = bit 0 of the FEC for byte 19), then they will be transmitted as shown in Figure 23.



	DATA BLOCK											
	INPUT DATA								RESULTING FEC			
	7	6	5	4	3	2	1	0	3	2	1	0
loaded first - Byte 0	0	1	2	3	4	5	6	7	8	9	10	11
Byte 1	12	13	14	15	16	17	18	19	20	21	22	23
Byte 2	24	25	26	27	28	29	30	31	32	33	34	35
Byte 3	36	37	38	39	40	41	42	43	44	45	46	47
Byte 4	48	49	50	51	52	53	54	55	56	57	58	59
Byte 5	60	61	62	63	64	65	66	67	68	69	70	71
Byte 6	72	73	74	75	76	77	78	79	80	81	82	83
Byte 7	84	85	86	87	88	89	90	91	92	93	94	95
Byte 8	96	97	98	99	100	101	102	103	104	105	106	107
Byte 9	108	109	110	111	112	113	114	115	116	117	118	119
Byte 10	120	121	122	123	124	125	126	127	128	129	130	131
Byte 11	132	133	134	135	136	137	138	139	140	141	142	143
Byte 12	144	145	146	147	148	149	150	151	152	153	154	155
Byte 13	156	157	158	159	160	161	162	163	164	165	166	167
Byte 14	168	169	170	171	172	173	174	175	176	177	178	179
Byte 15	180	181	182	183	184	185	186	187	188	189	190	191
Byte 16	192	193	194	195	196	197	198	199	200	201	202	203
loaded last - Byte 17	204	205	206	207	208	209	210	211	212	213	214	215
CRC Byte 18	216	217	218	219	220	221	222	223	224	225	226	227
CRC Byte 19	228	229	230	231	232	233	234	235	236	237	238	239

FIRST OUT	0	1	2	3	4	5	6	7	8	9	10	11	
	12	13	14	15	16	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	31	32	33	34	35	
	36	37	38	39	40	41	42	43	44	45	46	47	
	48	49	50	51	52	53	54	55	56	57	58	59	
	60	61	62	63	64	65	66	67	68	69	70	71	
	72	73	74	75	76	77	78	79	80	81	82	83	
	84	85	86	87	88	89	90	91	92	93	94	95	
	96	97	98	99	100	101	102	103	104	105	106	107	
	108	109	110	111	112	113	114	115	116	117	118	119	
	120	121	122	123	124	125	126	127	128	129	130	131	
	132	133	134	135	136	137	138	139	140	141	142	143	
	144	145	146	147	148	149	150	151	152	153	154	155	
	156	157	158	159	160	161	162	163	164	165	166	167	
	168	169	170	171	172	173	174	175	176	177	178	179	
	180	181	182	1883	184	185	186	187	188	189	190	191	
	192	193	194	195	196	197	198	199	200	201	202	203	
	204	205	206	207	208	209	210	211	212	213	214	215	
	216	217	218	219	220	221	222	223	224	225	226	227	
	228	229	230	231	232	233	234	235	236	237	238	239	LAST OUT

Fig.23 Interleaving - Input/Output

Scrambling

The Data Block may be transmitted or received as scrambled information in accordance with the setting of the Scramble Enable bit (Mode Register SCREN).

All formatted bits of a Data Block are passed through a 9-bit scrambler. This scrambler is initialised at the beginning of the first data block in every frame.

The 511-bit sequence is generated with a 9-bit shift register with the output of the 5th and 9th stages being Exclusive OR'd and fed back to the input of the 1st stage. The scrambler is enabled when SCREN = logic '1' or when the Transmit Scrambler Output (TSO) task is selected but disabled during all others.

Operational Information ......

Rx/Tx Note

The filtering required to reduce the transmitted bandwidth causes energy from each bit of information to be smeared across 3 bit-times. To ensure that the last bit transmitted is received correctly it is necessary to add an 8 bit 'hang byte' to the end of each message. Thus the tasks required to transmit an isolated Mobitex frame are:

T7H + (n x)TDB + TSB

When receiving this data the extra byte can be ignored as its only function is to ensure integrity of the last bit and not to carry any information itself.

It is suggested that a '00110011' or '11001100' pattern is used for this 'hang byte'.

"Receive Frame" Example

If the FX909 is required to receive a Mobitex Frame the following sequences of control and data will have to be issued:

Sequence Explanation	Action
Prerequisites: V_{DD} Applied; PLLBW, LEVRES, DARA, CKDIV and SCREN set as required; Tx/Rx set to Rx mode - Carrier has been detected	Tx/Rx = '0'
Steps	

1 Set AQLEV and AQBC bits, 2 bit-periods after the carrier has been detected, to initiate the level acquisition and bit clock extraction sequences -	AQLEV = '1'
	AQBC = '1'

2 2 Frame Sync bytes loaded to Data Buffer followed by setting LFSB task -	Write to Data Buffer LFSB
3 FX909 interrupts with IRQ when the 2nd byte is read from the Data Buffer -	IRQ

4 Status Register is read -
Task is set to search for a Mobitex Frame Head -
	BFREE = '1'
	Task = SFH

5 FX909 will interrupt with IRQ when a valid frame sync is detected and the header bytes decoded with no uncorrectable errors -	IRQ

6 Status Register is read -
MOBAN bit checked and 2 frame head control bytes read -
	BFREE = '1'
	MOBAN = '1' or '0'
	Read from Data Buf

7 Task is set to receive a Mobitex Data Block	Task = RDB
8 FX909 will interrupt with IRQ when a data block has been received and the CRC has been calculated -	IRQ

9 Status Register is read, CRC validity checked and 18 data block bytes read -	BFREE = '1'
	CRCFEC = '1'
	Read from Data Buffer

10 Set task if more information is expected:	
GOTO Step 4 if last data block received and another Frame Head is imminently expected	Step 4
GOTO Step 7 if another Mobitex Data Block is expected	Step 7
If the last data block has been decoded and no more information is expected then the task bits need not be set as the FX909 will automatically select the idle state.	

Operational Information ......

"Transmit Frame" Example

If the FX909 is required to send a Mobitex Frame the following sequences of control and data will have to be issued:

Sequence Explanation	Action
Prerequisites: V_{DD} Applied; SCREN, DARA and CKDIV set as required; Tx/Rx set to Tx mode -	Tx/Rx = '1'
Steps	
1 6 bytes forming the frame head are loaded into the data buffer -	Write to Data Buffer
followed by a 2 bit-period pause to let the filter stabilise -	
followed by setting the Transmit 7-Byte Frame Head task -	Task T7H
2 FX909 interrupts with IRQ when the 6th byte is read from the data buffer -	IRQ
Status Register is read -	BFREE = '1'
18 bytes of data are loaded into the data buffer -	Write to Data Buffer
followed by setting the Transmit Data Byte task -	TDB
4 FX909 interrupts with IRQ when the 18th byte is read from the data buffer -	IRQ
5 Status Register is read -	BFREE = '1'
Host may load data and set next task as required:	
GOTO Step 1 if the last data block for this frame has been transmitted and another frame is to be immediately transmitted -	Step 1
GOTO Step 3 if another data block in this frame is to be transmitted -	Step 3
GOTO Step 6 if the last data block for this frame has been transmitted and no more data is to be immediately sent -	Step 6
6 1 byte representing the 'hang byte' is loaded into the Data Buffer -	Write to Data Buffer
followed by setting the Transmit Single Byte task -	TSB
After the 'hang byte' has been transmitted and no more data is to be sent then a new task need not be written and the µController can wait for the IBEMPTY interrupt when it can, if required, shut down the Tx RF circuits.	

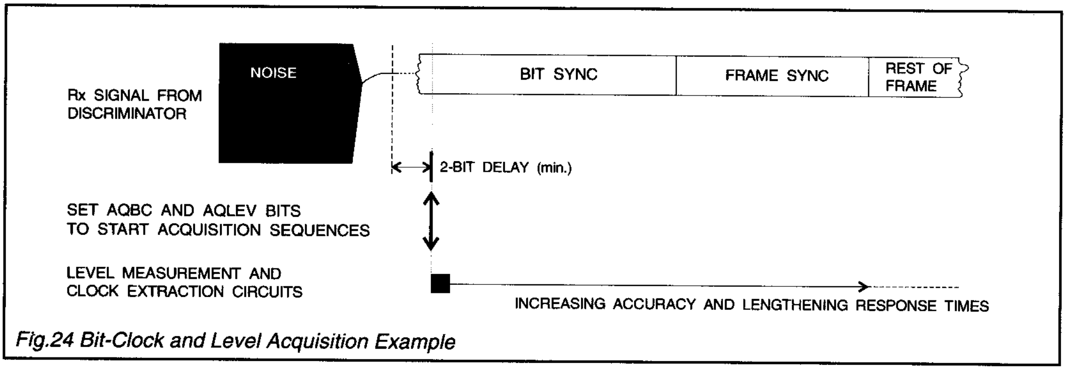

Rx SIGNAL FROM DISCRIMINATOR
NOISE
BIT SYNC
FRAME SYNC
REST OF FRAME
2-BIT DELAY (min.)
SET AQBC AND AQLEV BITS TO START ACQUISITION SEQUENCES
LEVEL MEASUREMENT AND CLOCK EXTRACTION CIRCUITS
INCREASING ACCURACY AND LENGTHENING RESPONSE TIMES

Fig.24 Bit-Clock and Level Acquisition Example

Received Signal Acquisition

Level Measurement and Clock Extraction

To achieve reasonable error rates the FX909 modem needs to make accurate measurements of the received signal amplitude, dc offset and bit-timing. Accurate measurements, especially in the presence of noise, are best made by averaging over a relatively long time period.

In most cases the modem will be used to receive isolated messages from a distant transmitter that is only turned on for a very short time before the message starts; also, the received baseband signal from the radio's frequency discriminator will have a dc offset due to small differences between the receiver and transmitter reference oscillators and hence their 'carrier' frequencies.

To cater for this situation, AQBC and AQLEV (Acquire Bit Clock and Level) commands are provided which, when triggered, cause the modem to follow an automatic sequence designed to perform the measurements as quickly as possible.

Note that due to the delay through the Rx lowpass filter, the AQBC and AQLEV sequences should not be started until about 2-bit times after the Rx carrier has been detected.

In a system where the controlling µController is not able to detect the Rx carrier, the AQBC and AQLEV sequences may be started at any time; possibly when no carrier is being received. However in this case the clock and level acquisition operation will take longer as the circuits will have to recover from the change from a large amplitude noise signal at the output of the frequency discriminator to the wanted signal, probably with a dc offset. In this type of system the time between the turn-on of the transmitter and the start of the Frame Sync pattern should be extended -preferably by extending the Bit Sync sequence to 32 or even 48 bits.

Warning: Clock extraction circuits work by detecting the timing of received edges, i.e. a chang from '0' to '1' or '1' to '0'. They will eventually fail if logic '1's' or logic '0's' are transmitted continuously. Similarly, the level measuring circuits require '00' and '11' bit-pairs to be received at reasonably frequent intervals.

Guides to AQBC and AQLEV sequences are given below. See Figure 24.

Acquire Receive Signal Levels (AQLEV)

The Acquire Receive Signal Levels (AQLEV) sequence starts with a measurement of the average signal voltage over a period of 1 bit-time. The sequence continues by measuring the positive-going and negative-going peaks of the signal. The attack and decay times used in this 'lossy peak detect' mode are such that a sufficiently accurate measurement can be made within 16 bits of a '11001100 ...' pattern (i.e the bit-sync sequence) to allow the bit-clock extraction circuits to operate.

Once the device has detected frame sync. or, if SFH or SFS has not been set, after 30 bits, then the the level measurement circuits will switch to the residual setting: 'Lossy Peak Detect', 'Peak Detect', 'Peak Averaging' or 'Hold'. Note that for normal operation the LEVRES bits would only be set to 'Hold' for the duration of a fade.

Sequence

SFH or SFS is set; Frame Sync is being searched for:
 1 bit of clamp
 Lossy Peak detect until Frame Sync is detected
 Residual setting.

SFH or SFS is not set; Frame Sync is not being searched for:
 1 bit of clamp.
 30 bits of Lossy Peak Detect.
 Residual setting.

Acquire Bit Clock (AQBC)

The Acquire Bit Clock (AQBC) sequence follows a similar pattern; starting with a very fast initial estimate of the received bit timing (Wide), when the AQBC bit is set.

When the device has detected Frame Sync. or, if SFH or SFS has not been set after 16 bits, then the bandwidth is set to Medium. 30 bits later the clock extraction circuits will change to the residual value.

Sequence

SFH or SFS is set; Frame Sync is being searched
 'Wide' setting until Frame Sync detected.
 30 bits of 'Medium' setting.
 Residual setting.

SFH or SFS is not set; Frame Sync is not being searched for:
 16 bits of 'Wide' setting.
 30 bits of 'Medium' setting.
 Residual setting.

Operational Information ......

Control and Data Load Timing

Control Instructions, Task and Data is loaded to the FX909 in a parallel form as detailed in the relevant Programming Information sections of this Data Sheet. Timing information for Read and Write operations is given in Figure 25; timing parameters are provided in the Specification section.

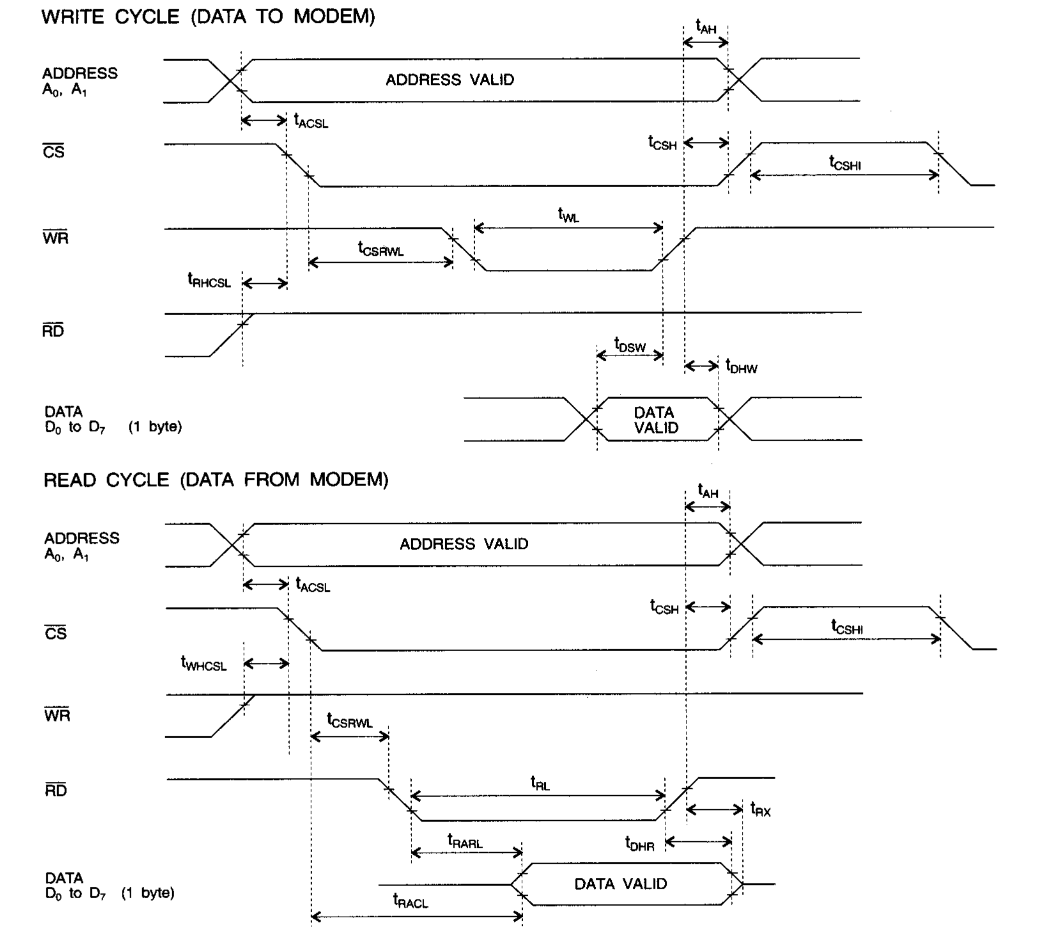


Fig.25 FX909 - μProcessor Read/Write Timing

Specification

Absolute Maximum Ratings

Exceeding the maximum rating can result in device damage. Operation of the device outside the operating limits is not implied.

Supply voltage		-0.3 to 7.0V
Input voltage at any pin (ref V_{SS} = 0V)		-0.3 to (V_{DD} + 0.3V)
Sink/source current (supply pins)		+/- 30mA
(other pins)		+/- 20mA
Total device dissipation @ T_{AMB} 25°C		800mW Max.
Derating		10mW/°C
Operating temperature range:	**FX909J4/L2**	-40°C to +85°C
Storage temperature range:	**FX909J4/L2**	-40°C to +85°C

Operating Limits

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

	Remarks	Min.	Max.	Unit
Supply Voltage (V_{DD})		4.5	5.5	V
Operating Temperature (T_{OP})		-40.0	+85.0	°C
Bit Rate		4,000	19,200	bits/sec
Xtal/Clock Frequency		1.0	10.0	MHz

Operating Characteristics

All device characteristics are measured under the following conditions unless otherwise specified:

V_{DD} = 4.5 to 5.5V, T_{OP} = -40°C to +85°C. Xtal/Clock Frequency = 4.096MHz. Bit Rate = 8,000 bits/sec. Noise Bandwidth = Bit Rate.

Characteristics	See Note	Min.	Typ.	Max.	Unit
Static Values					
I_{DD}	1	-	3.0	-	mA
Tx Output *See also Figure 26*					
Impedance	2	-	1.0	2.5	kΩ
Signal Level	3	0.9	1.0	1.1	Vp-p
Tx Data Delay	4	-	4.0	6.0	bits
Rx Input					
Impedance (Rx In pin)		10.0	-	-	MΩ
Rx Input Amp Voltage Gain		-	500	-	V/V
Input Signal Level	5	0.7	1.0	1.3	Vp-p
Rx Data Delay	6	-	3.5	-	bits
Xtal/Clock					
'High' pulse Width	7	40.0	-	-	ns
'Low' pulse Width	7	40.0	-	-	ns
Input Impedance		10.0	-	-	MΩ
Inverter Gain (I/P = 1mV rms @ 1kHz)		20.0	-	-	dB
µController Interface					
Input logic '1' Level	8, 9	V_{DD} -1.5	-	-	V
Input Logic '0' Level	8, 9	-	-	1.5	V
Input Leakage Current (V_{IN} = 0V to V_{DD})	8, 9	-5.0		+5.0	µA
Input Capacitance	8,9	-	10.0	-	pF
Output Logic '1' Level (IOH = 120µA	9	V_{DD} -0.4			V
Output Logic '0' Level (IOL = 360µA)	9,10	-	-	0.4	V
'Off' State Leakage Current (V = V_{DD})	10	-	-	10	µA

Specification ......

Modem Read/Write Load Timing

Description		Spec. Page Note	Min.	Typ.	Max.	Unit
t_{ACSL}	“Address Valid” to “CS Low” time		0	-	-	ns
t_{AH}	“Address Hold” time		0	-	-	ns
t_{CSH}	“CS Hold” time		0	-	-	ns
t_{CSHI}	“CS High” time	12	6.0	-	-	Xtal/Clock Cycles
t_{CSRWL}	“CS” to “WR” or “RD” Low time		0	-	-	ns
t_{DHR}	“Read-Data Hold” time		0	-	-	ns
t_{DHW}	Write-Data Hold time		0	-	-	ns
t_{DSW}	Write-Data Set-Up” time		90.0	-	-	ns
t_{RHCSL}	“RD High” to “CS Low” time (write cycle)		0	-	-	ns
t_{RACL}	“Read Access” time from “CS Low”	11	-	-	175	ns
t_{RARL}	“Read Access” time from “RD Low”	11	-	-	145	ns
t_{RL}	“RD” Low time		200	-	-	ns
t	“RD High” to “D_0-D_7 3-State” time		-	-	50.0	ns
$_{SL}$	“WR High” to “CS Low” time (read cycle)		0	-	-	ns
t_{WL}	“WR” Low time		200	-	-	ns

Notes:

1. Not including any current drawn from the modem pins by external circuitry.
2. Small signal impedance (dynamic measurement).
3. Measured after external CR (R_4/C_6) filter, for 1111000011110000.. bit sequence; at V_{DD} = 5.0V (output level is proportional to V_{DD}).
4. Measured between issuing first task after idle and the centre of the first bit at Tx Out (see Figure 14, 'The Transmit Process').
5. For optimum performance, measured at the Rx Feedback pin, for a '...11110000...' bit sequence.
6. Measured between centre of last bit of an Rx single byte or Frame Sync at Rx In and an IRQ interrupt to the µController.
7. Timing for an external input to the Xtal/Clock pin.
8. $\overline{WR}$, $\overline{RD}$, $\overline{CS}$, A_0 and A_1 pins.
9. D_0 - D_7 pins.
10. $\overline{IRQ}$ pin.
11. With 30pF (Max.) to V_{SS} on D_0 - D_7 pins.
12. Xtal/Clock cycles at the Xtal/clock pin.

Specification ......

Lowpass Filter Response

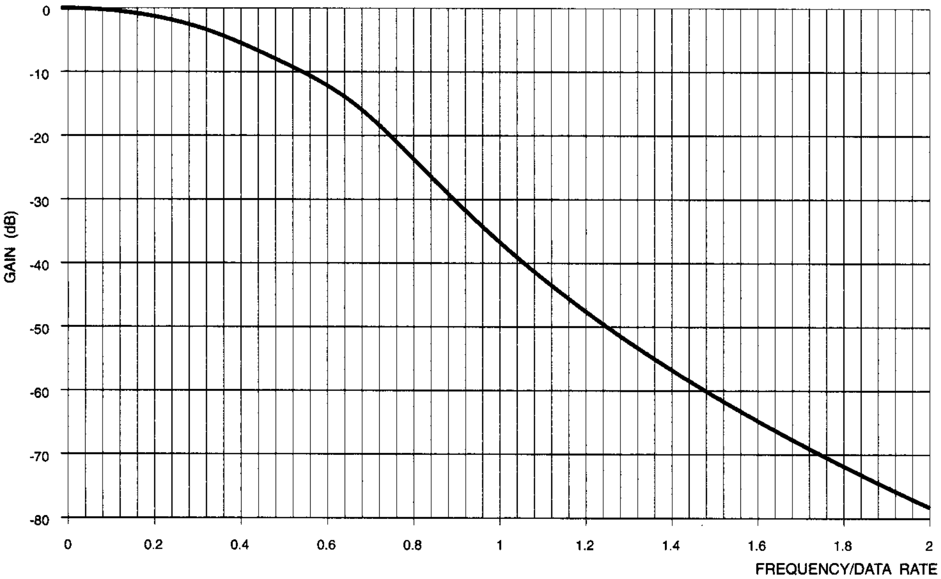


Fig.26 Lowpass Filter Response (Including the External RC Filter Components) in the Tx Mode

Signal-to-Noise Performance

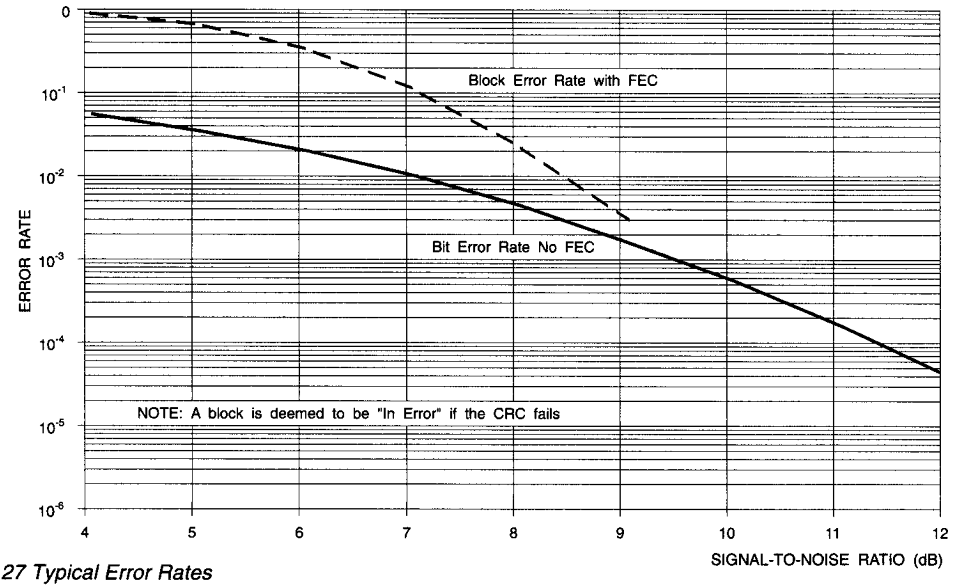


Fig.27 Typical Error Rates

Bit and Block Errors

The figure above shows both bit and block error possibilities for a received signal with a specific signal-to-noise performance. Bit Errors are individually detected errors in the raw data stream.

Block Errors are those where a complete block of data (240 bits; see Figure 11 Data Format) is in error (as indicated by the CRC).

Note that signal-to-noise ratios illustrated in this Data Sheet are calculated for noise in a bit-rate bandwidth.

Package Outlines

The FX909 is available in the package styles outlined below. Mechanical package diagrams and specifications are detailed in Section 10 of this document.

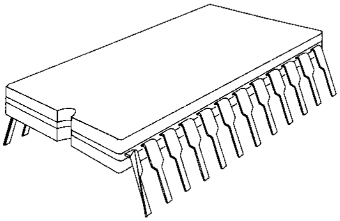
Pin 1 identification marking is shown on the relevant diagram and pins on all package styles number anti-clockwise when viewed from the top.

Handling Precautions

The FX909 is a CMOS LSI circuit which includes input protection. However precautions should be taken to prevent static discharges which may cause damage.

FX909J 24-pin cerdip DIL (J4)

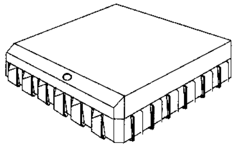
NOT TO SCALE



Max. Body Length	32.03mm
Max. Body Width	14.81mm

FX909LS 24-lead plastic leaded chip carrier (L2)

NOT TO SCALE



Max. Body Length	10.40mm
Max. Body Width	10.40mm

dering Information

FX909J	24-pin cerdip DIL	(J4)
FX909LS	24-lead plastic leaded chip carrier	(L2)

CML does not assume any responsibility for the use of any circuitry described. No circuit patent licences are implied and CML reserves the right at any time without notice to change the said circuitry.