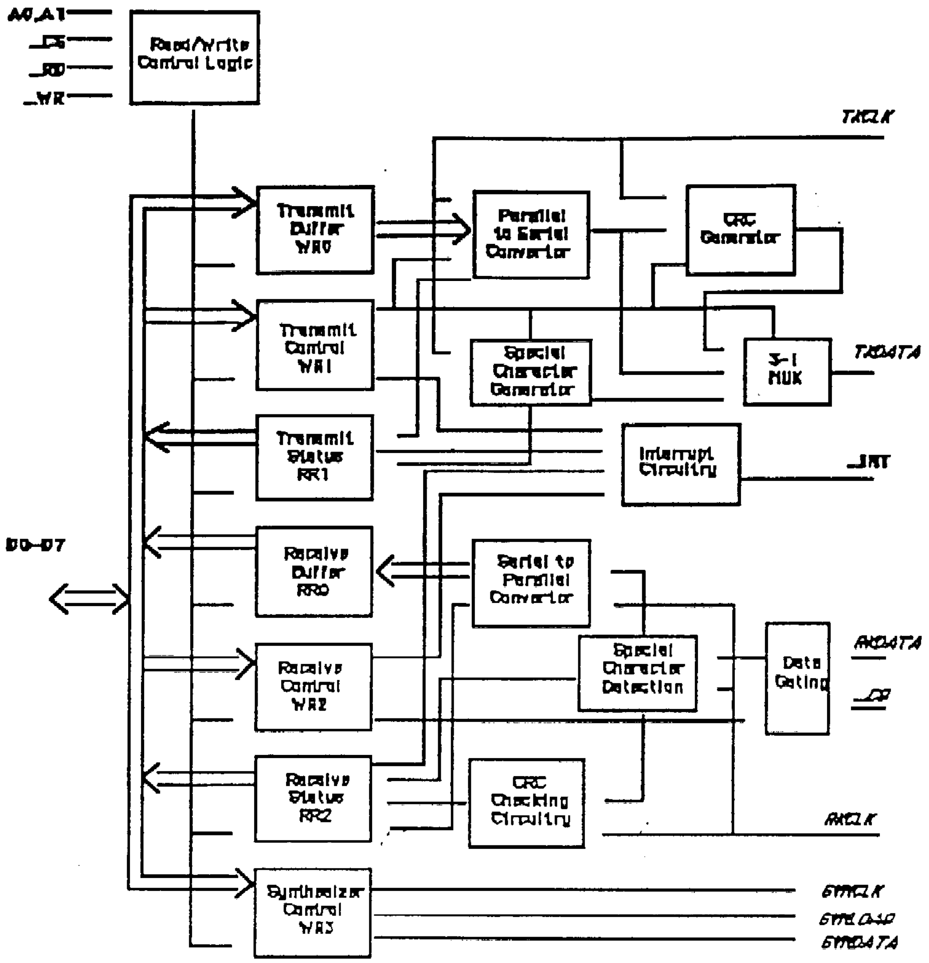

Notes: BOLD=interface signal, *ITALIC*=internal signal

Figure 4 - RDA Series Bus Interface Diagram

3.1 Transmitter Operation

The first step of the transmit process is to bring the radio out of the Standby mode by raising the $\overline{\text{STANDBY}}$ line high, while the TX/$\overline{\text{RX}}$ line is held low (if this has not already been done). After a power on delay, the transmitter frequency can be programmed by writing a sequence of bytes to the Synthesizer Control Register for the RDA radio. After the transmit frequency has settled, transmission is initiated by raising the TX/$\overline{\text{RX}}$ line. This line will turn off the receiver section while enabling the transmitter output amplifiers. After a transmitter turn on delay, the radio will be ready to send valid data.

The RDA radio includes a full HDLC controller that is controlled through a parallel bus interface. There are 4 write registers and 3 read registers on the RDA which are used to set up the radio and to send and receive data packets. The HDLC protocol involves sending packets which include an opening FLAG, a number of DATA bytes, a 16 bit CRC (Cycle Redundancy Check), and a closing FLAG. The FLAG character is a 01111110, and is differentiated from data by a feature called zero insertion and deletion which inserts a zero after any block of 5 successive 1's in the DATA or CRC blocks. As the transmitter is turning on, an ABORT character 11111111 can be sent repeatedly until it is possible to send a packet. The ABORT character can also be sent between packet transmissions.

Once the RDA radio is ready to transmit valid data, an HDLC packet can be transmitted by appropriately sending a FLAG, DATA bytes, a CRC, and a closing FLAG. The RDA radio can be operated in either interrupt or polled modes, and it must be serviced every byte period while transmitting to tell it to send a control character or to provide a new data byte. The 16 bit CRC is automatically generated by the RDA radio and the RDA radio also handles the zero insertion and deletion features. The FLAG and ABORT characters are generated by the RDA and sent by setting bits in control registers so that they do not undergo zero insertion.

The HDLC Transmitter on the RDA produces a serial data stream which then drives the spread spectrum section of the radio. This serial data stream is scrambled to ensure even spectral content of the output in the event that all 1's or all 0's are transmitted. The scrambled data is modulated in the spreading section by a pseudo-random noise sequence. This signal is conditioned further for spectral shape in the spectral conditioner. The resultant signal is modulated by the RF source to produce the final transmit signal. Finally, the signal is amplified to the desired output power.

3.2 Receiver Operation

The first step of the receive process is to bring the radio out of Standby mode by raising the $\overline{\text{STANDBY}}$ line high while the TX/$\overline{\text{RX}}$ line is held low (if this has not already been done). After a power on delay, the receiver frequency can be programmed by writing a sequence of bytes to the Synthesizer Control Register on the RDA radio. After the receive frequency has settled, valid data can be received by the RDA radio.

Signals enter the radio via the antenna connector. A front-end band pass filter rejects signals outside the 902-928MHz band. A low noise amplifier (LNA) performs the first stage of signal amplification. After passing through the LNA, the signal enters the RF Downconverter for translation to an Intermediate Frequency (IF). The IF filter then rejects interference present at frequencies other than the selected frequency channel.Next, the demodulator converts the signal into a form presentable to the de-spreader. The de-spreader, the core of the spread spectrum receiver, correlates the incoming signal and extracts the transmitted clock and data. Finally, the descrambler reverses the scrambling performed by the transmitter, resulting in a serial data stream.

The RDA radios' HDLC controller operates on the recovered serial data stream. Special FLAG and ABORT characters are detected and reported through status bits and interrupts. Any zeros that the transmitter has inserted in the data are removed, and the received packet CRC is computed. If any errors were detected in the received data, a status bit is set. Finally, the received data is converted from serial to parallel form and can be read over the bus interface. The radio can be operated in either interrupt or polled modes, and it must be serviced every byte period while receiving a packet to read the latest data byte or status indication.

3.3 Synthesizer

Common to both the receiver and transmitter is the frequency synthesizer. It generates the RF frequencies needed for both the transmitter and the RF Downconverter in the receiver chain.

In the transmit mode, the synthesizer is programmed to the exact transmit frequency, between 902 and 928 MHz. In the receive mode, however, the synthesizer is programmed to a higher frequency for high side injection into the RF Downconverter. This method of tuning allows the radio to receive a packet in one channel, then transmit the following packet in a different channel if desired. This feature is particularly useful for repeaters.

The synthesizer can tune over a wide range of frequencies. This must be kept in mind when in operation. If the synthesizer is programmed to a frequency outside the 902-928MHz band while the transmitter is enabled, the radio will be in violation of FCC regulations.

3.4 TX/$\overline{\text{RX}}$ Line

The TX/RX line controls the pre-driver stages in the power amplifier and the supply current to the receiver. Since the output amplifier is a Class C (unbiased) type, it draws negligible supply current in the absence of RF power. This has the advantage that there is no need for a high-current, high-power switch to control the output. In battery-operated applications, the high-current supply (VBAT) can be connected directly to the battery.

3.5 $\overline{\text{STANDBY}}$

The $\overline{\text{STANDBY}}$ switch controls the power supply to most of the radio. When the radio comes out of Standby mode, the supply voltages must settle before the radio can be used. Radio timing is fully described in section 4.

3.6 RSSI

The radio provides a Received Signal Strength Indicator (RSSI) derived from the high gain amplifiers in the radio IF section. This indicator measures RF power present in the channel being received. Its voltage varies from approximately 1V to 4V for input signal levels between -100dBm and -30dBm respectively. The RSSI does not indicate a valid spread spectrum code or a low bit error rate. If, for instance, a strong level is indicated on the RSSI, but the carrier detect output is still inactive, then there is interference present in the channel. This feature can be useful for channel selection. This line's output impedance is ~100KΩ.

4 Operation

This section describes the procedures one must follow in order to establish communication between two RDA series radios. Section 4.1 *Overview* briefly covers the major steps necessary to prepare the radio for communication. Section 4.2 *Parallel Interface* presents registers descriptions and timing diagrams that explain how to operate the parallel interface/HDLC controller. Section 4.3 *Radio Control* provides procedures and timing diagrams for placing the radio into each of its fundamental modes: Transmit, Receive and Standby. The final operation section, section 4.4 *Data Interface Timing,* details procedures involved in communicating between two radios.

4.1 Overview

The radio is always in one of three modes: Transmit, Receive or Standby. In Transmit and Receive modes, the radio is transmitting/receiving data on the channel selected during setup. In Standby mode, the radio consumes minimal power and does not perform any RF operation.

Timing is critical when transitioning from one mode to another. In addition, the synthesizer must be reloaded when changing the channel or altering the Transmit/Receive mode. As an example, to bring the radio from Standby to Transmit mode, requires the following steps:

- Set the TX/$\overline{\text{RX}}$ line low.
- Raise the $\overline{\text{STANDBY}}$ line high.
- Wait for the hardware to power-up.
- Program the synthesizer to transmit at the desired frequency.
- Wait for the synthesizer to settle.
- Raise the TX/$\overline{\text{RX}}$ line high to activate the transmitter.
- Wait for the transmitter to reach full power.
- Transmit data packets using the parallel interface to the HDLC controller.

4.2 Parallel Interface Overview

This section describes the steps necessary to send and receive data packets using the parallel interface of the RDA radios. The parallel interface consists of 4 write only registers, 3 read only registers, an interrupt line, a TX/$\overline{\text{RX}}$ line, and a $\overline{\text{STANDBY}}$ line. The registers are accessed as follows:

_CS	_WR	_RD	A1	A0	Action
0	0	1	0	0	Write to WR0 (Transmit Buffer)
0	0	1	0	1	Write to WR1 (Transmit Control)
0	0	1	1	0	Write to WR2 (Receive Control)
0	0	1	1	1	Write to WR3 (Synthesizer Control)
0	1	0	0	0	Read RR0 (Receive Buffer)
0	1	0	0	1	Read RR1 (Transmit Status)
0	1	0	1	0	Read RR2 (Receive Status)

The TX/$\overline{\text{RX}}$ line is used to place the RDA radio into the Transmit mode when it is set high, and places the radio into Receive mode when it is set low. The $\overline{\text{STANDBY}}$ line is used to place the entire RDA radio in a shutdown/reset mode when it is set low. The interrupt line of the RDA parallel interface is optional, and can be used in interrupt driven systems. As an alternative, the RDA radios fully support polled mode operation.

Transmit Mode

When sending an HDLC packet using the RDA radio, there are three parallel interface registers that are used: the Transmit Buffer (WR0), the Transmit Control Register (WR1), and the Transmit Status Register (RR1). The Transmit Control Register (WR1) determines what character will be sent by the RDA radio next. The options are an ABORT, FLAG, or WAKEUP special character, as well as a data byte which is stored in the Transmit Buffer (WR0) and an internally generated two byte CRC. The ABORT, FLAG, WAKEUP, and CRC characters can each be simply generated and sent by setting the appropriate bits in the Transmit Control Register (WR1).

The Transmit Character Request status bit, in the Transmit Status Register (RR1 Bit1) is set as soon as the command to send the last character has been accepted by the RDA radio. At this point, the RDA radio is ready to accept a new command which will identify the next character to send, since the previous character has been loaded and is being shifted out of the radio. The Transmit Character Request bit can be used to generate an interrupt signal from the RDA radio when transmitting. It is qualified by a number of bits in the Transmit Control Register (WR1) when generating an interrupt. These permit the interrupt to be shut off while repeatedly sending special characters such as ABORT, FLAG or WAKEUP. This feature allows a special character to be auto-repeated without triggering Transmit Character Requests which would require repeated intervention by the host processor. The Transmit Character Request bit is cleared by writing to the Transmit Control Register (WR1) or the Transmit Buffer (WR0).

A user data byte is sent by writing to the Transmit Control Register (WR1) to configure it to send data, and loading the data byte into the Transmit Buffer (WR0). The first data byte of a packet should be pre-loaded into the Transmit Buffer (WR0) prior to setting up the Transmit Control Register (WR1) to send data. Transmission of subsequent bytes is initiated by simply writing to the WR0. The Transmit Buffer (WR0) is single buffered, so that while the previous byte is being shifted out of the radio, a new byte should be loaded into the WR0 register. The Transmit Status Register (RR1) contains status bits which show whether the Transmit Buffer (WR0) is full, whether a new character is being requested (Transmit Character Request), and whether a Transmitter Overrun or Underrun has occurred.

Receive Mode

When receiving a HDLC data packet using the RDA radio, there are three parallel interface registers that are used: the Receive Control Register (WR2), the Receive Buffer (RR0), and the Receiver Status Register (RR2). The Receive Control Register (WR2) is used to setup the sources of receiver interrupts, and to setup the gating of the received data with the Carrier Detect signal. The Receiver Status Register (RR2) reports the reception of ABORT, FLAG, WAKEUP, DATA and CRC characters. Another bit in the Receiver Status Register (RR2) is cleared when a valid CRC character has been received, and is set when an invalid CRC has been received. Finally, the Receiver Status Register (RR2) contains a bit which indicates whether a receiver overrun has occurred which indicates that the host processor failed to read a data byte before it was overwritten. The RDA radio contains a single level of output data buffering, so the host processor must be able to read received data bytes within a byte period of their detection.

The Receiver Status Register (RR2) is generally read each time a receive interrupt is received to determine the source. The Receive Buffer (RR0) holds the last data byte that has been received by the RDA radio. The Receive Buffer should be read after the Data Ready bit in the Receiver Status Register (RR2) has been set. A number of bits in the Receiver Control Register (WR2) are used to mask various receive interrupt sources. The most important of these is Bit2 which is used to mask ABORT interrupts. When a receiver is turning on, the received data will most likely be gated high before valid data is received, and this will appear as a continuous stream of ABORT characters. To avoid continuous ABORT interrupts in the power on stage, this interrupt source should be masked, and the host processor should be interrupted on an opening FLAG which signifies the beginning of a valid data packet.

The following two sections show typical sequences of events for sending and receiving data packets. They utilize the interrupt feature of the RDA radio to trigger actions. If a polled mode of operation is desired as an alternative, the appropriate bits in the Transmit Status Register (RR1) and the Receiver Status Register (RR2) can be continuously monitored by the host processor. Many variations can be made to these typical sequences, and the user is referred to Section 4.3 which lists detailed descriptions for the operation of the various bits in the RDA radio parallel interface.

Typical Sequence of Events to Send a Data Packet (Using Interrupts)

- Take the RDA radio out of Standby by setting $\overline{\text{STANDBY}}$ high, and setting TX/$\overline{\text{RX}}$ low.
- Wait for the RDA radio to power up.
- Program the RDA synthesizer by writing a sequence of bytes to WR3 (described in section 4.3.1).
- Wait for the synthesizer to settle.
- Set the TX/$\overline{\text{RX}}$ line high for Transmit mode.
- Setup the RDA radio to send continuous ABORT characters while it is powering on by setting the appropriate bits in the Transmit Control Register (WR1). The interrupt line of the RDA in Transmit mode is tied to the Transmit Character Request bit in the Transmit Status Register (RR1). When the RDA radio is requesting information on what character to send next, the Transmit Character Request bit is asserted. The Transmit Character request should be disabled for the ABORT character so that interrupts are not generated after each character is sent. Both of these operations can be performed by setting WR1 to "00001000".
- Wait for the transmitter to power up and the receiver to lock up.
- Enable Transmit Character Request interrupts for the ABORT character by writing to WR1. This will generate an interrupt after the next ABORT character is sent. This is done by setting WR1 to "00000000".
- When the next RDA interrupt is received, set up the radio to send a FLAG character and enable Transmit Character Requests for the FLAG character by writing to WR1. This will cause an opening FLAG to be sent as the next character. This is done by setting WR1 to "00000001".
- When the next RDA interrupt is received, load the first data byte into WR0. The radio should immediately be setup to send data by writing a "00000100" to WR1.
- When the next RDA interrupt is received, load the next data byte into WR0. It is not necessary to write to WR1 for each byte in the data packet.
- Continue to send data bytes by loading the next byte into WR0 after each interrupt is received.
- After the last data byte has been sent, and when the next RDA interrupt is received, setup the RDA to send the CRC bytes by writing to WR1. This requires a single write to WR1 to trigger the transmission of two consecutive CRC bytes. This is done by writing "00000011" to WR1.
- When the next RDA interrupt is received, write to WR1 so that the RDA will send the closing FLAG character. This is done by writing "00000001" to WR1.
- Setup the RDA radio to send continuous ABORT characters while it is powering off by setting the appropriate bits in the Transmit Control Register (WR1). The Transmit Character request should be disabled for the ABORT character so that interrupts are not generated after each character is sent. Both of these operations can be performed by setting WR1 to "00001000".
- Set the TX/$\overline{\text{RX}}$ line low to place the radio into receive mode.
- Wait for the transmitter to power down.
- Switch the RDA radio off by setting $\overline{\text{STANDBY}}$ low.

Typical Sequence of Events to Receive a Data Packet
(Using Interrupts)

- Take the RDA radio out of Standby by setting $\overline{\text{STANDBY}}$ high, and setting TX/$\overline{\text{RX}}$ low.
- Wait for the RDA radio to power up.
- Program the RDA synthesizer by writing a sequence of bytes to WR3 (described in section 4.3.1).
- Wait for the synthesizer to settle.
- Setup the RDA radio to allow interrupts from a FLAG character, and to gate the received data with Carrier Detect by writing to WR2. Gating the received data with the Carrier Detect ensures that random noise will not inadvertently create a FLAG interrupt when no signal is being detected by the RDA radio. These actions are done by writing "01110110" to WR2.
- Upon receiving an interrupt, the RDA radio has received a FLAG character, and a packet should be coming. There will be approximately a 4 byte-delay until the first data byte is ready to read. Setup the RDA radio to allow interrupts from FLAG and ABORT characters as well as the DATA byte that is expected. Since a valid data packet is being received, the received data should no longer be gated with Carrier Detect. This is done by writing "00010000" to WR2.
- Upon receiving an interrupt, verify that it is from the first data byte being read by checking the Receiver Status Register (RR2). If the Data Ready Bit is set, then read the data byte from the Receive Buffer (RR0).
- Continue to check the Receiver Status Register (RR2) when receiving interrupts. When data bytes are ready, they can be read from RR0, and eventually when the closing FLAG of the HDLC packet is recognized, the final data byte, and the CRC result will all be ready simultaneously.
- Verify that the CRC result in the Receiver Status Register (RR2) is correct.
- The packet has been received, and the RDA should now be setup to interrupt only on a new opening FLAG character, and the received data should be gated with carrier detect. This is done be writing "01110110" to WR2.

4.2.1 Register Descriptions

Transmit Buffer (WR0)
(Line Settings: A1=0, A0=0; $\overline{\text{CS}}$=0, $\overline{\text{RD}}$=1, $\overline{\text{WR}}$=0)

This register contains the next data byte to transmit. It is cleared when STANDBY=0 or TX/RX=0

Transmit Control (WR1)
(A1=0, A0=1; $\overline{\text{CS}}$=0, $\overline{\text{RD}}$=1, $\overline{\text{WR}}$=0)

This register contains the HDLC transmitter control bits. These bits are cleared when $\overline{\text{STANDBY}}$=0 or TX/$\overline{\text{RX}}$=0.

Bit0-Bit2 determine the next character the radio will send. There are five possible selections for transmission: ABORT, FLAG, WAKEUP, DATA or CRC. The ABORT, FLAG and WAKEUP characters as well as DATA (the contents of the Transmit Buffer), are all a single byte long. The internally calculated CRC is two bytes long. A single write to the Transmit Control Register sends both CRC bytes. The following table lists shows the character to be sent next based on Bit0-Bit2 settings.

BIT2	BIT1	BIT0	Next Character to Be Sent
0	0	0	ABORT (11111111)
0	0	1	FLAG (01111110)
0	1	0	WAKEUP (01111111)
0	1	1	CRC (internally calculated 2 bytes)
1	0	0	DATA (Transmit Buffer contents)
1	0	1	ABORT (11111111)
1	1	0	ABORT (11111111)
1	1	1	ABORT (11111111)

Bit3-Bit5 are the Transmit Character Request (TCR) Mask bits. These bits determine whether a new character will be requested after sending an ABORT, FLAG or WAKEUP character. These bits allow the user to send one of these characters continuously without receiving transmit character requests or interrupts after each character.

BIT3	Description
0	TCR is produced when sending ABORTS
1	TCR is masked when sending ABORTS

BIT4	Description
0	TCR is produced when sending FLAGS
1	TCR is masked when sending FLAGS

BIT5	Description
0	TCR is produced when sending WAKEUPS
1	TCR is masked when sending WAKEUPS

Bit6 is the Transmitter Interrupt Mask. Setting this bit to a value of "1" will mask all transmit interrupts.

Bit7 of the Transmit Control Register must be set to "0" for the radio to pass FCC regulations.

Receive Control (WR2)
(A1=1, A0=0; $\overline{\text{CS}}$=0, $\overline{\text{RD}}$=1, $\overline{\text{WR}}$=0)

The Receive Control Register contains the HDLC Receiver control bits. These bits are cleared when $\overline{\text{STANDBY}}$=0 or TX/$\overline{\text{RX}}$=1.

Bit0 must be set to "0" for proper operation.

Bit1-Bit5 set up masks for the possible receive interrupts: DATA ready, ABORT, FLAG, WAKEUP, and CRC.

BIT1	Description
0	Received DATA Byte causes interrupt.
1	Received DATA Byte Interrupt masked.

BIT2	Description
0	Received ABORT causes interrupt.
1	Received ABORT Interrupt masked.

BIT3	Description
0	Received FLAG causes interrupt.
1	Received FLAG Interrupt masked.

BIT4	Description
0	Received WAKEUP causes interrupt.
1	Received WAKEUP Interrupt masked.

BIT5	Description
0	CRC Received causes interrupt.
1	CRC Received Interrupt masked.

BIT6	Description
0	Received DATA passes directly into HDLC Receiver Block.
1	Received DATA is gated high whenever valid carrier is not detected.

When the receiver is on and attempting to receive a packet, it is recommended that Bit6 be set to "1". In this case, if there is no valid input, the HDLC controller will see a constant high level on the received data line which will be interpreted as an ABORT character. If Bit6 were set to "0", the HDLC controller could interpret random noise as a false opening FLAG.

Once a valid opening FLAG is received, Bit6 should be set to "0" to fully utilize the radio's sensitivity and range. At this point, it is known that the data being received is not random, and no longer requires gating by the carrier detect signal.

Bit7 must be set to "0" for proper operation.

Synthesizer Control (WR3)
(A1=1, A0=1; $\overline{CS}$=0, $\overline{RD}$=1, $\overline{WR}$=0)

This register contains the three bits that control the synthesizer. These bits are cleared when STANDBY=0.

Bit0 drives the Synthesizer Clock line (SYNCLK).
Bit1 drives the Synthesizer Data line (SYNDATA).
Bit2 drives the Synthesizer Load line (SYNLD).

See Section 4.3.1 for proper programming of synthesizer control lines.