IEEE 802.11
Wireless Access Method and Physical Layer Specifications

Title: AN UNINTELLIGENT RADIO INTERFACE

Presented by :

W. B. Cooper
National Semiconductor
Santa Clara, CA
408-721-5815 (V)
408-721-4115 (Fax)
Burch@Berlioz.nsc.com (Email)

Abstract

This paper provides an example of a simple service interface between a wireless LAN radio PHY and the digital hardware and associated software which implements the controller for the PHY. This proposed interface is between the medium dependent layer and the convergence layer. The implementation is simple in the sense that it implies no intelligence on the radio side of the PHY layer. All synchronization and timing functions are placed in the controller on the MAC side of service interface. The system is assumed to be frequency hopping, but it is believed that the proposed structure would serve for other physical layers as well. The purpose of this paper is to illustrate what is required for direct control of the radio and baseband functions as contrasted to a intelligent radio interface.

Conclusion

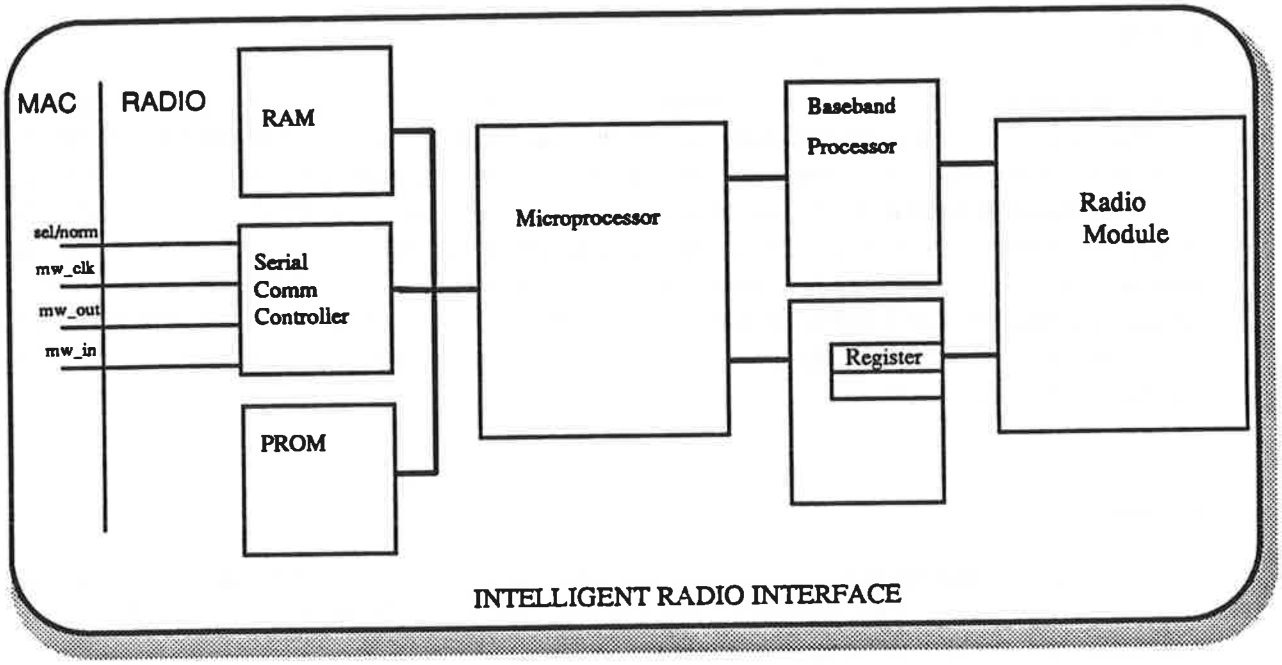
An exposed interface between the medium dependent layer and convergence layers is better suited for a low cost radio than the present DTE/DCE interface presently planned between the convergence and medium independent layers.

1.0 Introduction

The type of command and control structure used between the controller and PHY layers as implications on the complexity and performance of the PHY. The difference between an interface that uses a command set for control, and simple a unintelligent interface is the addition of a controller or state machine for interpretation and translation of the command set into device dependent operations. A simple command structure where PHY implementation independent primitives are passed back and forth over the interface allows:

- A small number of interconnect lines.
- A command set for control of the PHY.
- Command set that is implementation independent.

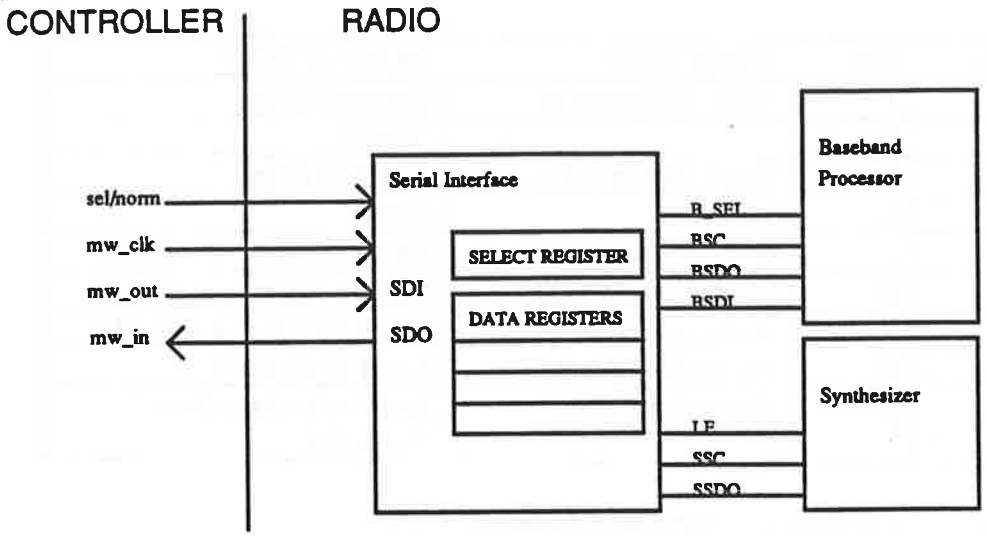
The following block diagram gives an example of an intelligent interface:



There are, however, disadvantages in using architecture which uses a highly structured command interface between the controller and PHY.

- More complexity in the radio.
- Higher power consumption.
- More digital noise sources in the radio.
- Difficulty in synchronizing MAC and PHY functions.

The following block diagram represents an example of what a radio interface would look like with no imbedded control logic. The microprocessor and memory are no longer required. The interface is still serial in order to reduce the number of interconnects. The result is a simpler radio with no intelligence and lower power consumption.



RADIO INTERFACE WITH NO CONTROLLER FUNCTIONS

2.0 Intelligent Interface

A structured intelligent interface for a frequency hopping radio might have a command set that looks something like this:

Command	Operation
Reset	Initializes controller
Sleep Mode Enable	Places radio in low power mode
Load Hop Table	Load hopping pattern
Set Channel Immediate	Go to a particular channel
Set Output Power	Set output power of the transmitter
Select Antenna	Manual diversity control of antenna
Read RSSI	Read the signal strength of the last received message
Loop Back	Loop back digital data
Enter Diagnostic Mode	Stop hopping
Leave Diagnostic Mode	Start hopping

There are many other possible commands that would be applicable, but the idea is to have command set which is interpreted by a controller to remove any implementation dependencies in the PHY.

An interface connector for such a implementation might have a very few number of connections:

SIGNAL	PIN	DIRECTION	DESCRIPTION
tx_clk	1	radio to Controller	Transmit Clock
gnd	2		Ground
tx_data	3	Controller to radio	Transmit Data
vcc	4		Supply
rx_clk	5	radio to Controller	Receive Clock
rx_data	6	radio to Controller	Receive Data
cmd_in	7	radio to Controller	Serial data In to Controller
cmd_clk	8	Controller to radio	Serial Wire clock
cmd_out	9	Controller to radio	Serial Wire data Out of Controller

Table 1: Connector Definition Intelligent PHY

The attractiveness of this structure is that it can remove the synchronization and control issues from the MAC side of the interface and allows for an implementation independent design around which a standards specification can be written. The disadvantage of course is that the commands over the interface must be interpreted and executed in the PHY. An alternative implementation is to provide for direct control of the PHY.

3.0 A Simple PHY with Direct Control of PHY Functions

If the interface between the PHY and Controller to be buried within the design and no exposed service point is to be provided to the user then the control of the PHY need not be as structured. The controller functions that provide for operation of the MAC can also control the PHY. The result can be an overall reduction in the complexity of the design and power consumption.

In addition, if the Controller does not have direct control over PHY then it is difficult to provide synchronization and control functions to efficiently provide for sleep mode operation. For example, if the Access Point Controller could provide a list of addressed mobiles during the header then a longer period in a low power mode could be achieved by the mobile during inactive periods. If a mobile did not here its address during the first header then it could go into a low power mode until the start of the next hop.

The interface to an unintelligent PHY might require a few more connections to bring out to the interface some of the critical control functions. Table II is an example of what an interface connector to an unintelligent PHY might look like.

4.0 Connector Definition

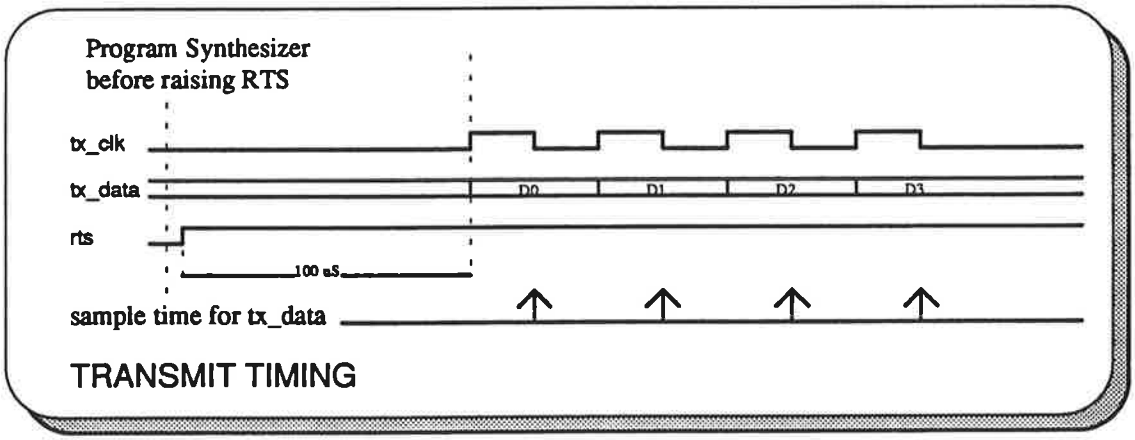
SIGNAL	PIN	DIRECTION	DESCRIPTION
tx_clk	1	radio to Controller	Transmit Clock
gnd	2		Ground
tx_data	3	Controller to radio	Transmit Data
vcc	4		Supply
rx_clk	5	radio to Controller	Receive Clock
rx_data	6	radio to Controller	Receive Data
mw_in	7	radio to Controller	Serial data In to Controller
mw_clk	8	Controller to radio	Serial Wire clock
mw_out	9	Controller to radio	Serial Wire data Out of Controller
rts	10	Controller to radio	Request To Send
cd	11	radio to Controller	Carrier Detect
rx_start	12	Controller to radio	Receiver Start
spare	13		
sel/norm	14	Controller to radio	Select / Normal
fault	15	radio to Controller	Radio summary fault

Table 2: Connector Definition Simple PHY

5.0 Transmit Interface and Timing

The transmit data over the interface is serial in order to reduce the number of interconnects. The data is synchronous with transmit clock. A request to send line is used to activate the transmitter.

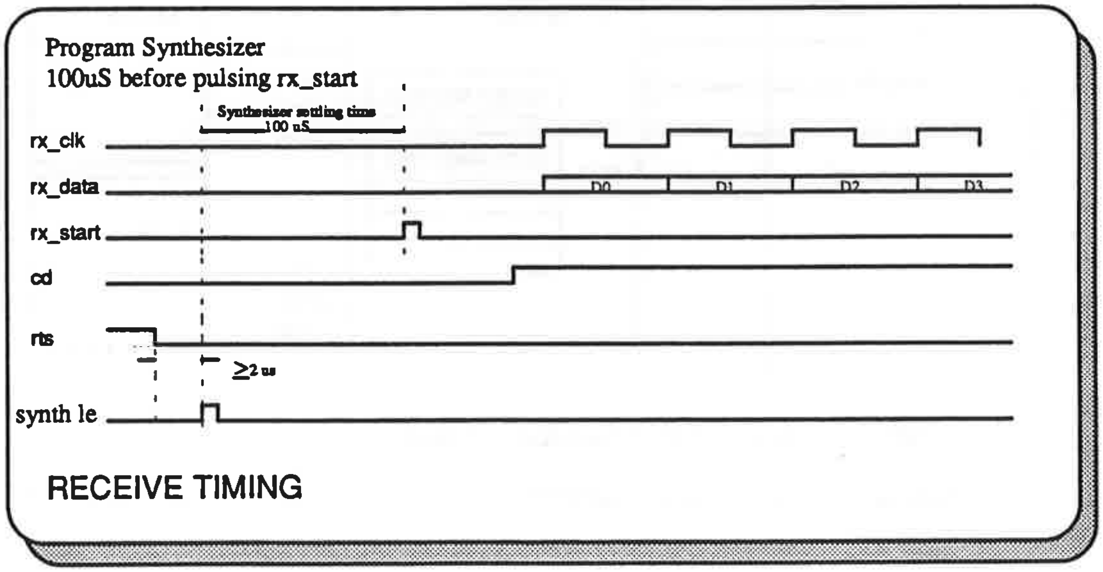
Signal	Direction	Description
tx_clk	r>m	**Transmit Clock.** This clock is provided by the radio to clock data in. The nominal clock rate is 1 MHz It will begin toggling 100uS after RTS is raised.
tx_data	m>r	**Transmit Data.** This is the data to be transmitted. The data is sampled by the radio, on the falling edge of the tx_clk. A preamble of 10... (50 bits) will be inserted prior to any data including opening flag, etc. This 50 bit preamble aids the receive radio in acquiring the received signal. The preamble will precede every new transmission.
rts	m>r	**Request to Send.** This is a signal generated by the Controller to request that the radio switch to transmit mode. The radio will delay 100uS before turning on transmit amps and clocking transmit data. The delay is required for the rf synthesizer to settle. When RTS is low the radio will switch to the receive mode.



6.0 Receive Interface and Timing

The receive side of the radio interface is also serial. The receive clock is extracted from the data by the baseband processor. The radio is in the receive mode whenever RTS is inactive. Receive start resets the acquisition sequence and places the antennas in the diversity selection mode. Carrier detect is a baseband derived signal which indicates that a valid preamble has been detected.

Signal	Direction	Description
rx_clk	r>m	**Receive Clock.** The radio will generate this clock only when it detects carrier. The clock will become active after CD goes high.
rx_data	r>m	**Receive Data.** This is the data received by the radio. The data should be sampled on the falling edge of the rx_clk signal. It is up to the Controller to decide whether the data is valid.
rts	m>r	**Request to Send.** This is a signal generated by the controller to request the radio to switch to the receive mode (rts low). The synthesizer receive frequency can be preloaded anytime prior to applying the latch signal.
cd	r>m	**Carrier Detect.** This signal indicates that the radio is receiving RF energy. CD could be triggered by the radio in response to noise or interference. It is up to the controller to decide this and restart the receiver by toggling the **rx_start** signal.
rx_start	m>r	**Receiver Start.** In response to this signal the receiver reset Internal slicing and DC offset registers and begin searching for CD again.



Program Synthesizer
100uS before pulsing rx_start

RECEIVE TIMING

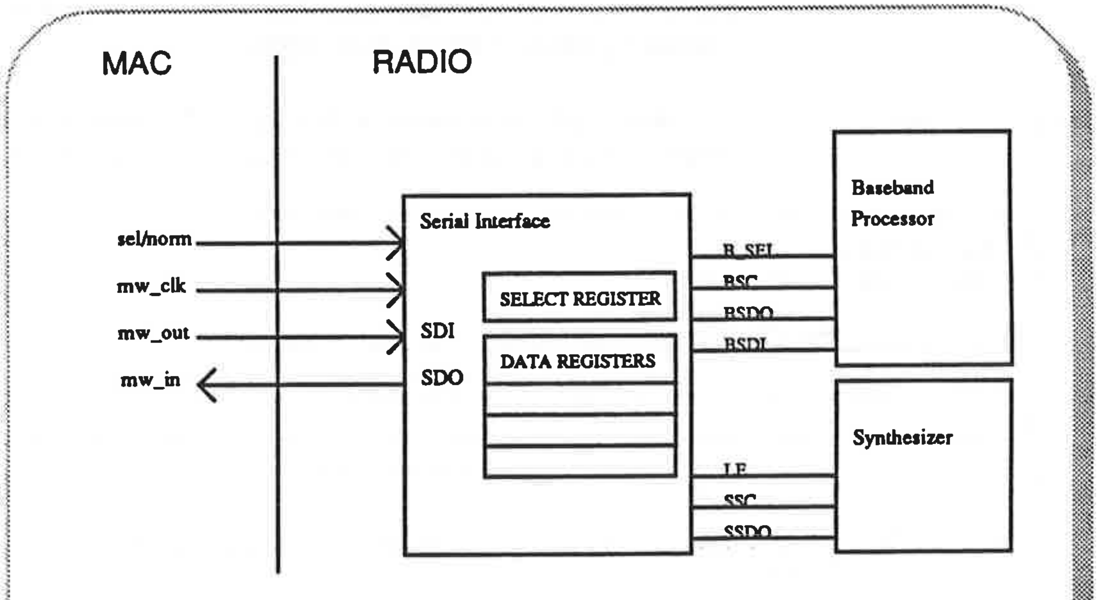
7.0 Serial Interface

The Serial Interface allows the Controller to address one of the following devices:

o the Baseband Processor
o the Synthesizer chip
o the Configuration Data Registers
o other serial devices

Serial Control Signals

Signal	Direction	Description
sel/norm	m>r	**Select/Normal.** This is a signal supplied by the MAC to access the serial Select Register.
sc	m>r	**Serial Clock.** This clock is supplied by the MAC to clock in/out the Serial interface serial data.
sdi	m>r	**Serial Data In.** The serial data input for the Serial Interface.
sdo	r>m	**Serial Data Out.** The serial data output for the Serial Interface.

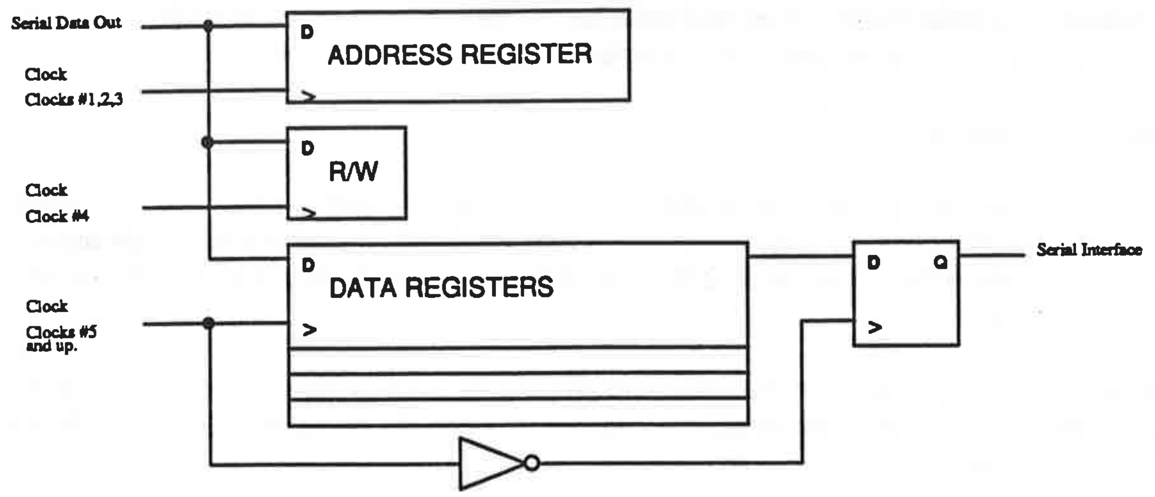


SERIAL INTERFACE BLOCK DIAGRAM

7.2 PHY Data Registers

The Data registers of the PHY are used to control various PHY functions and to read information gathered by the PHY.

The data word for the PHY consists first of three (3) bits of Address to determine which data is to be accessed. This is followed by one bit to indicate read or write and the remaining bits are to/from the data register.



BLOCK DIAGRAM OF DATA REGISTERS

7.2.3 PHY Data Register Description

Status/Fault Register.

Jabber Timer Fault. A jabber timer fault indicates the radio has been transmitting for too long a period of time. The maximum allowed transmit time is 400 ms. When this condition is detected the radio will automatically switch off the transmitter and set the Jabber Timer Fault.

Synthesizer Unlock. This fault indicates the radio is not generating the correct channel frequency.

Transmit Amplifier Fault. This fault indicates the radio output power amplifier has failed. The radio will not be able to transmit when this fault occurs.

Control Register 1.

RF Power Control. These 3 bits control the radio Transmit output power level. The values listed in 6.2.2 are in dB below the peak power level (dBc), i.e. 0 dBc represents maximum output power. At power-up these three bits will be 000. The MAC should set these bits to 011 (for maximum output power).

Antenna Diversity Control. This bit selects one of two antenna patterns for antenna diversity. Due to the multipath environment signal reception on one pattern may be significantly better than the other pattern.

Active/Standby Control. This bit controls the power consumption mode for the 2240 (IF Discriminator), 2216B (LNA/Mixer) and the 2320 (Synthesizer). For normal operation this bit should be high (Active). At power up this bit will be low so it will be necessary for the controller to set this bit high. The controller can set this bit low to put the radio in a sleep mode.

Control Register 2.

Powerdown When this bit is set high the receive functions are in a powerdown mode. At powerup this bit is low (normal power mode). The Controller can set this bit high to put the radio in a sleep mode.

/Baseband Reset. This is the reset control (active low) for the 2410 (Baseband Processor). At powerup this bit will be low, so it will be necessary for the Controller to set this bit high.

/RSSI_en. The RSSI input from the 2240 is sampled every 11.1 us (200/sys clk for the 2410). Each sample is compared against the present peak value, and if larger, it replaces the peak value. The peak value is reset on every falling edge of the /RSSI_en signal. /RSSI_en should be low 5 us before the RSSI measurement is taken to allow for ADC settling.

Loopback Control. The loopback mode is a test mode for loopback of baseband signal. In this mode (active high) RTS must be high to activate this mode.

7.0 THE SYNTHESIZER CHIP

The Synthesizer on the RF/IF Board is used to generate the RF frequencies for the Radio. There are 82 discrete Transmit frequencies and 82 discrete Receive frequencies used. That is, the synthesizer is programmed for different frequencies in the transmit and receive modes. The synthesizer generates these frequencies based on data words received through its serial interface.

The channel frequencies are generated through the following two synthesizer control words:

o the Programmable Reference Divider (R Counter) and Prescalar Select (S Latch)
o the Programmable Divider (N Counter)

8.0 THE BASEBAND PROCESSOR CHIP.

The Baseband Processor on the Baseband Board provides 3 major functions for the Radio. These functions are:

o generate the transmit waveform for the RF signal
o control of the baseband DC correction
o provides access to a digitized version of the RSSI.

Read RSSI Peak Hold Value.
The Received Signal Strength Indicator (RSSI) is a 6-bit value that is log linearly proportional to the input signal level to the IF Demod.