ATHEROS®
COMMUNICATIONS

Data Sheet

Revision September 2003

AR5212 Multiprotocol MAC/Baseband Processor for 5 GHz and 2.4 GHz Wireless LANs

General Description

The Atheros AR5212 is part of the AR5001X+ solutions for 5 GHz and 2.4 GHz wireless local area networks (WLANs). The AR5212 integrates multiprotocol media access control (MAC), a baseband processor, a PCI/CardBus host interface, and analog-to-digital and digital-to-analog (ADC/DAC) converters. When combined with the AR5111 and AR2111, these chip sets enable a cost effective, low power silicon solution for WLAN station (STA) and access point (AP) applications.

The AR5212 implements half-duplex OFDM, CCK, and DSSS baseband processing supporting all IEEE 802.11a, 802.11b, and 802.11g data rates. In Atheros Turbo Mode™, the AR5212 supports data rates up to 108 Mbps. Additional features include signal detection, automatic gain control, frequency offset estimation, symbol timing, and channel estimation.

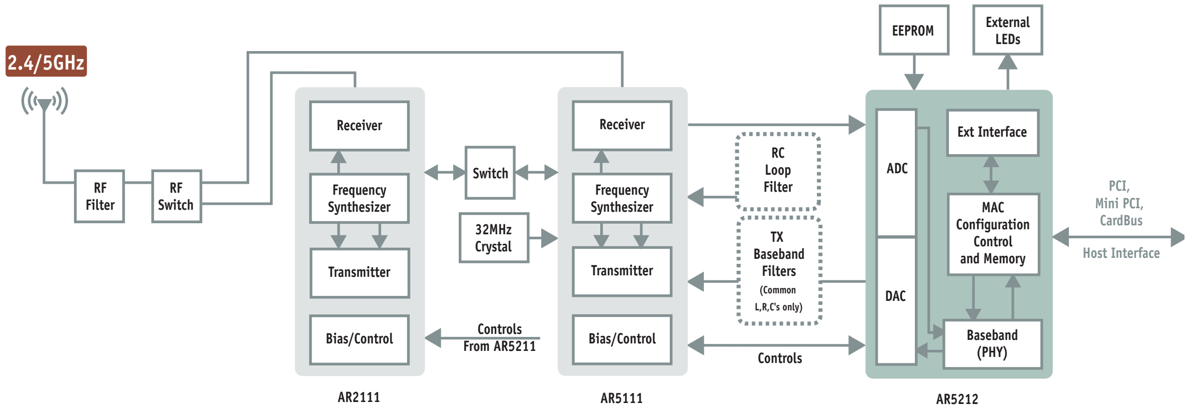
The AR5212 MAC provides support for the IEEE 802.11 wireless MAC protocol. Additional features include support for 802.11i security, receive (Rx) and transmit (Tx) filtering, error recovery, and quality of service (QoS).

The AR5212 handles frame data transfer to and from the host using the PCI/CardBus interface, interrupt generation and reporting, power save, and error and status reporting. The PCI/CardBus interface also provides access to the AR5212, AR5111, and AR2111 registers. External peripheral interface includes serial EEPROM, GPIOs, and LEDs.

Features

- IEEE 802.11a, IEEE 802.11b, and IEEE 802.11g compatible
- BPSK, QPSK, 16 QAM, 64 QAM, DBPSK, DQPSK, and CCK modulation schemes supported
- Data rates of 6 Mbps to 54 Mbps for 802.11a, 1 Mbps to 11 Mbps for 802.11b, 1 Mbps to 54 Mbps for 802.11g, and Atheros Turbo Mode™ offering up to 108 Mbps
- Host interface PCI 2.3 and PC Card 7.1 compatible
- IEEE 1149.1 standard test access port and boundary scan architecture supported
- Standard 0.25 µm complementary metal oxide silicon (CMOS) technology
- 196 BGA plastic package
- With the AR5111/AR2111—Complete all-CMOS solution for both the IEEE 802.11a 5 GHz WLANs, 5.15 to 5.85 GHz, and 802.11b/802.11g 2.4 GHz WLANs, 2.412 to 2.484 GHz, frequency band operations

System Block Diagram



Contents

List of Figures

List of Tables

1. Pin Descriptions

This section contains both a package pinout (see Figure 1-1 on page 14) and a tabular listing of the signal descriptions (see Table 1-1 on page 15).

The following nomenclature is used for signal names:

- NC indicates no connection should be made to this pin.
- _L at the end of the signal name indicates active low signals.
- P at the end of the signal name indicates the positive side of a differential signal.
- N at the end of the signal name indicates the negative side of a differential signal.

The following nomenclature is used for signal types described in Table 1-1:

- IA indicates an analog input signal.
- I indicates a digital input signal.
- IH indicates signals with weak internal pull-up, to prevent signals from floating when left open.
- IL indicates signals with weak internal pull-down, to prevent signals from floating when left open.
- I/O indicates a digital bidirectional signal.
- OA indicates an analog output signal.
- O indicates a digital output signal.
- P indicates a power or ground signal.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	EPRM_EN_L	TDI	TMS	ADC_QN	AGND	ADC_IP	AGND	DAC_QN	AGND	DAC_IP	AGND	RFRESET_L	RFSHIFT	RFDATAOUT_0
B	TRST_L	GND	AGND	ADC_QP	AGND	ADC_IN	AVDD_ADC	DAC_QP	AGND	DAC_IN	XIN	RFLOAD	GND	RFDATAOUT_1
C	TCLK	TDO	GND	AVDD_ADC	PCI_MODE	ADC_RES	NC	AVDD_DAC	AVDD_DAC	AVDD_PLL	XOUT	GND	RFDATAOUT_2	RFDATAOUT_3
D	RX2	GPIO_0	GPIO_1	GND	VDD3_3	GND	VDD2_5	VDD2_5	GND	VDD3_3	GND	ANTC	RFDATAIN5	REFCLK
E	RX5	GPIO_2	GPIO_3	VDD3_3	GND	GND	GND	GND	GND	GND	VDD3_3	TX5	TX2	PLLBYPASS
F	EPRM_DO	EPRM_DI	RXCLEAR	GND	GND	GND	GND	GND	GND	GND	GND	ANTD	NC	NC
G	EPRM_CS	RFDATAIN2	VDD2_5	VDD2_5	GND	GND	GND	GND	GND	GND	VDD2_5	HITUNE5	GPIO_4	GPIO_5
H	EPRM_CLK	ATTEN5	VDD2_5	VDD2_5	GND	GND	GND	GND	GND	GND	VDD2_5	SLEEP	LED_0	LED_1
J	PCI_CLKRUN_L	PCI_AD31	PCI_AD30	GND	GND	GND	GND	GND	GND	GND	GND	ATTEN2	ANTB	ANTA
K	PCI_AD29	PCI_AD28	PCI_AD27	VDD3_3	GND	GND	GND	GND	GND	GND	VDD3_3	PCI_AD2	PCI_AD1	PCI_AD0
L	PCI_PME_L	PCI_AD26	PCI_AD25	GND	VDD3_3	GND	VDD2_5	VDD2_5	GND	VDD3_3	GND	PCI_AD5	PCI_AD4	PCI_AD3
M	PCI_CBE3_L	PCI_AD24	GND	PCI_AD21	VDD3_3	PCI_CBE2_L	PCI_DEVSEL_L	PCI_GNT_L	PCI_IDSEL	VDD3_3	PCI_AD13	GND	PCI_AD7	PCI_AD6
N	PCI_REQ_L	GND	PCI_AD22	PCI_AD20	PCI_AD18	PCI_FRAME_L	PCI_TRDY_L	PCI_STOP_L	PCI_PAR	PCI_CBE1_L	PCI_AD14	PCI_AD11	PCI_AD9	PCI_AD8
P	PCI_AD23	PCI_SERR_L	PCI_RST_L	PCI_AD19	PCI_AD17	PCI_IRDY_L	PCI_CLK	PCI_INT_L	PCI_PERR_L	PCI_AD16	PCI_AD15	PCI_AD12	PCI_AD10	PCI_CBE0_L

Figure 1-1. **Package Pinout (Top View)**

Table 1-1. **Signal to Pin Relationships and Descriptions**

Symbol	Pin	Type	Source or Destination	Description
ADC/DAC				
ADC_IP	A6	IA	AR5111	Differential (P for Positive and N for Negative) baseband in phase (I) and quadrature (Q) inputs from the AR5111 to the analog/digital converter (ADC) using the external anti-aliasing filters. See the AR5111 data sheet.
ADC_IN	B6	IA	AR5111	
ADC_QN	A4	IA	AR5111	
ADC_QP	B4	IA	AR5111	
ADC_RES	C6	IA	AR5111	Connect a 6.19 KΩ ± 1% resistor to ground for bandgap.
DAC_IN	B10	OA	AR5111	Differential baseband I and Q outputs from the DAC to the AR5111 using the external reconstruction filters. See the AR5111 data sheet.
DAC_IP	A10	OA	AR5111	
DAC_QN	A8	OA	AR5111	
DAC_QP	B8	OA	AR5111	
PCI Interface				
PCI_CBE3_L	M1	I/OH	PCI bus	PCI multiplexed bus command and byte enables. Active low. During the address phase of a transaction, these signals define the bus command. During the data phase, they are used as byte enables.
PCI_CBE2_L	M6	I/OH	PCI bus	
PCI_CBE1_L	N10	I/OL	PCI bus	
PCI_CBE0_L	P14	I/OH	PCI bus	
PCI_CLK	P7	IL	PCI bus	PCI bus clock.
PCI_CLKRUN_L	J1	IH	PCI bus	PCI bus clock run. Active low. Provides for starting and stopping the PCI clock. Refer to "PCI Clkrun" on page 35.
PCI_DEVSEL_L	M7	I/OH	PCI bus	PCI dev select. Active low.
PCI_FRAME_L	N6	I/OL	PCI bus	PCI frame. Active low.
PCI_GNT_L	M8	IH	PCI bus	PCI grant. Active low.
PCI_IDSEL	M9	IL	PCI bus	PCI ID select. Not used by CardBus, external 10-K pull-up resistor required for CardBus application.
PCI_INT_L	P8	O	PCI bus	PCI interrupt. Active low.
PCI_IRDY_L	P6	I/OH	PCI bus	PCI initiator ready. Active low.

PCI_MODE	C5	IL	PCI bus	Select PCI or CardBus interface
				L = CardBus (default - internal pull down)
				H = PCI

PCI_PAR	N9	I/OH	PCI bus	PCI parity.
PCI_PERR_L	P9	I/OH	PCI bus	PCI parity error. Active low.
PCI_PME_L/ CSTSCHG	L1	O	PCI bus	PCI power management event. Active low. CardBus CSTSCHG. Active high (These signals are not supported in the AR5212).
PCI_REQ_L	N1	O	PCI bus	PCI request. Active low.
PCI_RST_L	P3	IH	PCI bus	PCI reset, reset the AR5212. Active low.

Table 1-1. **Signal to Pin Relationships and Descriptions (continued)**

Symbol	Pin	Type	Source or Destination	Description
PCI_SERR_L	P2	I/OH	PCI bus	PCI system error. Active low.
PCI_STOP_L	N8	I/OH	PCI bus	PCI stop. Active low.
PCI_TRDY_L	N7	I/OH	PCI bus	PCI target ready. Active low.
PCI_AD31	J2	I/OL	PCI bus	PCI_AD[31:0] is a multiplexed address and data bus. During the first clock of a transaction, PCI_AD[31:0] contains a physical byte address (32 bits). During subsequent clocks, PCI_AD[31:0] contains data.
PCI_AD30	J3	I/OL	PCI bus	
PCI_AD29	K1	I/OL	PCI bus	
PCI_AD28	K2	I/OL	PCI bus	
PCI_AD27	K3	I/OL	PCI bus	
PCI_AD26	L2	I/OL	PCI bus	
PCI_AD25	L3	I/OL	PCI bus	
PCI_AD24	M2	I/OL	PCI bus	
PCI_AD23	P1	I/OL	PCI bus	
PCI_AD22	N3	I/OL	PCI bus	
PCI_AD21	M4	I/OL	PCI bus	
PCI_AD20	N4	I/OL	PCI bus	
PCI_AD19	P4	I/OL	PCI bus	
PCI_AD18	N5	I/OL	PCI bus	
PCI_AD17	P5	I/OL	PCI bus	
PCI_AD16	P10	I/OL	PCI bus	
PCI_AD15	P11	I/OL	PCI bus	
PCI_AD14	N11	I/OL	PCI bus	
PCI_AD13	M11	I/OL	PCI bus	
PCI_AD12	P12	I/OL	PCI bus	
PCI_AD11	N12	I/OL	PCI bus	
PCI_AD10	P13	I/OL	PCI bus	
PCI_AD9	N13	I/OL	PCI bus	
PCI_AD8	N14	I/OL	PCI bus	
PCI_AD7	M13	I/OL	PCI bus	
PCI_AD6	M14	I/OL	PCI bus	
PCI_AD5	L12	I/OL	PCI bus	
PCI_AD4	L13	I/OL	PCI bus	
PCI_AD3	L14	I/OL	PCI bus	
PCI_AD2	K12	I/OL	PCI bus	
PCI_AD1	K13	I/OL	PCI bus	
PCI_AD0	K14	I/OL	PCI bus	

Table 1-1. **Signal to Pin Relationships and Descriptions (continued)**

Symbol	Pin	Type	Source or Destination	Description
AR5111 Interface				
RFRESET_L	A12	O	AR5111	Reset output to the AR5111. Active low.
RFDATAOUT_3	C14	O	AR5111 Serial_in3	Serial interface bus to the AR5111. A termination resistor of 68 Ω can be required to connect between this pin and Serial_in3.
RFDATAOUT_2	C13	O	AR5111 Serial_in2	Serial interface bus to the AR5111. A termination resistor of 68 Ω can be required to connect between this pin and Serial_in2.
RFDATAOUT_1	B14	O	AR5111 Serial_in1	Serial interface bus to the AR5111. A termination resistor of 68 Ω can be required to connect between this pin and Serial_in1.
RFDATAOUT_0	A14	O	AR5111 Serial_in0	Serial interface bus to the AR5111. A termination resistor of 68 Ω can be required to connect between this pin and Serial_in0.
RFLOAD	B12	O	AR5111 Serial_load	Serial interface control signal for loading the buffer register to the control register in the AR5111. A termination resistor of 68 Ω can be required to connect between this pin and Serial_load.
RFSHIFT	A13	O	AR5111 Serial_clk	Clock control for the AR5111 serial interface. A termination resistor of 68 Ω can be required to connect between this pin and Serial_clk.
RFDATAIN5	D13	IL	AR5111 DataOut	Serial interface input from the AR5111.
AR2111 Interface				
RFDATAIN2	G2	IL	AR2111 DataOut	Serial interface input from the AR2111.
EEPROM				
EPRM_CLK	H1	O	EEPROM	EEPROM clock.
EPRM_CS	G1	O	EEPROM	EEPROM chip select.
EPRM_DO	F1	O	EEPROM	EEPROM data output from the AR5212.
EPRM_DI	F2	IL	EEPROM	EEPROM data input to the AR5212.
EPRM_EN_L	A1	IL	EEPROM	EEPROM enable input to the AR5212 to select initialization values from the EEPROM to be loaded to the PCI/CardBus configuration registers. Active low. Refer to "Serial EEPROM Interface" on page 22.
JTAG				
TCLK	C1	IL	—	Test clock. Default low.
TDI	A2	IH	—	Test data input. Default high.
TDO	C2	O	—	Test data output.
TMS	A3	IH	—	Test mode select. Default high.
TRST_L	B1	IH	—	Test reset. Active low. Default high.

Table 1-1. **Signal to Pin Relationships and Descriptions (continued)**

Symbol	Pin	Type	Source or Destination	Description
General				
REFCLK	D14	I	AR5111	32 MHz reference clock for the PLL.
PLLBYPASS	E14	IL		PLL bypass. 0 = use PLL 1 = use REFCLK
SLEEP	H12	O		Output to indicate that the AR5212 is in sleep mode.
LED_1	H14	O	LED	Output to control network LED. See "LED" on page 29.
LED_0	H13	O	LED	Output to control Power/Network LED.
GPIO_5	G14	I/OL	—	General purpose I/O [5-0]. Default to inputs, control using the GPIOCR register. Input from the GPIOs can be read using the GPIODI register, and output to the GPIOs is provided by the GPIODO register. See "GPIO" on page 29.
GPIO_4	G13	I/OL	—	
GPIO_3	E3	I/OL	—	
GPIO_2	E2	I/OL	—	
GPIO_1	D3	I/OL	—	
GPIO_0	D2	I/OL	—	
ANTA	J14	O	Antenna	Antenna switch control. Output to control antenna switching. See "Switching" on page 31 for control of these signals.
ANTB	J13	O	Antenna	
ANTC	D12	O	Antenna	
ANTD	F12	O	Antenna	
RX5	E1	O	External LNA (Optional)	Control signals for switching on the optional external low noise amplifier (LNA) for 5 GHz.
RX2	D1	O	External LNA (Optional)	Control signals for switching on the optional external low noise amplifier (LNA) for 2.4 GHz.
TX5	E12	O	External PA (Optional)	Control signals for switching on the optional external power amplifier (PA) for 5 GHz. Active high, default inactive (low).
TX2	E13	O	External PA (Optional)	Control signals for switching on the optional external power amplifier (PA) for 2.4 GHz. Active high, default inactive (low).
ATTEN5	H2	O	Attenuation switch	External Rx attenuation, 5 GHz.
ATTEN2	J12	O	Attenuation switch	External Rx attenuation, 2.4 GHz.
RXCLEAR	F3	O		The wireless medium is clear.
HITUNE5	G12	O	External filter	External filter control.
XIN	B11	I		Low frequency sleep crystal. If this feature is not enabled, B11 can be tied to VDD (making this pin compatible with the AR5211).
XOUT	C11	O		Low frequency sleep crystal. If this feature is not enabled, C11 can be tied to GND (making this pin compatible with the AR5211).

Table 1-1. **Signal to Pin Relationships and Descriptions (continued)**

Symbol	Pin	Type	Source or Destination	Description
Power				
VDD3_3	D5, D10, E4, E11, K4, K11, L5, L10, M5, M10	P	3.3 V	Digital 3.3 V power supply.
VDD2_5	D7, D8, G3, G4, G11, H3, H4, H11, L7, L8	P	2.5 V	Digital 2.5 V power supply.
GND	B2, B13, C3, C12, D4, D6, D9, D11, E5-10, F4-11, G5-10, H5-10, J4-11, K5-10, L4, L6, L9, L11, M3, M12, N2	P	0 V	Ground.
AVDD_PLL	C10	P	2.5 V	Analog 2.5 V power supply for PLL.
AVDD_ADC	B7, C4	P	2.5 V	Analog 2.5 V power supply.
AVDD_DAC	C8, C9	P	2.5 V	Analog 2.5 V power supply.
AGND	A5, A7, A9, A11, B3, B5, B9	P	0 V	Analog ground.
NC	C7, F13, F14			No connection, must be open.

2. Functional Description

2.1 Overview

The AR5212 consists of three major functional blocks: PCI/CardBus interface, MAC, and digital PHY.

The IEEE 802.11 MAC functionality is partitioned between the host and the AR5212. IEEE 802.11 MAC data service is provided by the MAC of the AR5212, while the host software, with the aid of the AR5212 MAC, controls Tx and Rx queue processing.

The physical layer (PHY) is partitioned between the AR5212, the AR5111/AR2111, and external RLC filters for the ADC/DAC blocks. Refer to the *AR5111 Radio-on-a-Chip for 5 GHz Wireless LANs* data sheet for information on these filters. The baseband digital processing functions are implemented by the digital PHY of the AR5212. The radio frequency (RF) and baseband analog processing are provided by the AR5111/AR2111.

The configuration block, the AR5111/AR2111 interface, PLL, ADC, DAC, EEPROM interface, JTAG, antenna control, LED controls and GPIO complete the AR5212 functionality. See Figure 2-1.

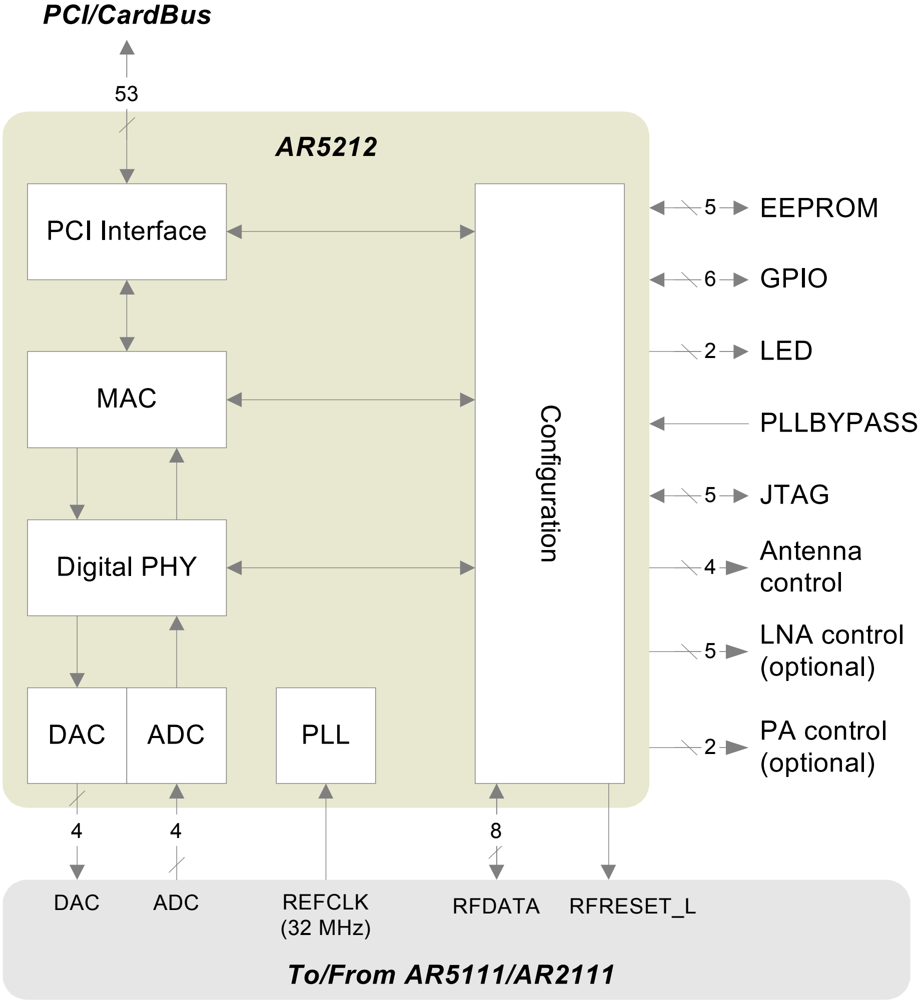


Figure 2-1. **Functional Block Diagram of the AR5212**

2.1.1 Configuration Block

The configuration block provides configuration, control, and status for each of the major functional blocks. This block comprises registers that can be accessed by the other blocks, and also by the host using the PCI/CardBus interface. The functionality of this block is described in "Register Descriptions" on page 79.

2.1.2 AR5212 Address MAP

Internal registers of the various functional blocks and the peripheral interface of the AR5212 are accessible by the host using the PCI/CardBus interface. The locations of these registers are defined as offset addresses. The host memory is mapped to the AR5212 address space by specifying the base address location in the PCI Base Address Register (see "Base Address" on page 85.) The combination of the host base address and the offset address allows access to a particular AR5212 internal register. Table 2-1 lists the offset addresses for the AR5212 internal registers and peripheral interface.

Table 2-1. **AR5212 Offset Addresses**

Offset Location	Usage	Description
0x0000 - 0x07FF	PCI/CardBus Interface and DRU registers	Control and status registers for the PCI CardBus Interface and DRU.
0x0800 - 0x0FFF	QCU registers	Control and status registers for the QCU.
0x1000 - 0x1FFF	DCU registers	Control and status registers for the DCU.
0x4000 - 0x4FFF	PCI clock domain registers	Registers that remain active while in sleep mode.
0x5000 - 0x50FF	CIS tuple	Card information structure (CIS) for PC Card operation. The CIS is loaded from the EEPROM.
0x6000 - 0x6FFF	EEPROM access register	Memory locations of the EEPROM are mapped to this address range of the AR5212, and allow access to the EEPROM.
0x8000 - 0x87FF	PCU registers	Control and status registers for the PCU.
0x8800 - 0x88FF	Key table	The look-up table used for encryption and decryption is stored in this register.

2.1.3 Operational Descriptions

The operations of the major functional blocks are described in the following sections:

- "Queue Control Unit (QCU)" on page 53
- "DCF Control Unit (DCU)" on page 57
- "Protocol Control Unit (PCU)" on page 61
- "Digital PHY Block" on page 71

The following sections describe additional functionality of the AR5212:

- "Serial EEPROM Interface" on page 22
- "PLL" on page 28
- "Reset" on page 28
- "AR5111/AR2111 Interface" on page 29
- "GPIO" on page 29
- "Antenna Controls (Switching)" on page 30

2.2 Serial EEPROM Interface

The AR5212 provides a serial interface for accessing an external EEPROM. This EEPROM is used for storing configuration information for the PCI/CardBus, application-specific and vendor-specific information.

The off-chip EEPROM is either a 4-Kb device, organized as 256 entries of 16 bits each (256x16); an 8-Kb device, organized as 512 entries of 16 bits each (512x16); or a 16-Kb device, organized as 1024 entries of 16 bits each (1024x16). The hardware automatically detects the EEPROM size.

At reset, some of the PCI/CardBus configuration registers are loaded from the EEPROM while others are programmed by the host or initialized by the AR5212 hardware. If the EPRM_EN_L pin is active upon power up, initialization values from the EEPROM are loaded to the PCI/CardBus configuration registers. If the EPRM_EN_L pin is inactive, the PCI/CardBus configuration registers contain the default values as listed in Table 2-2 on page 24.

To ensure the EEPROM contents are valid, a 16-bit word (EEPROM_MAGIC) at location 0x3D is checked. If the values do not match 0x5aa5, the contents of the EEPROM are ignored, and the default values are loaded.

More information regarding the PCI/CardBus interface and EEPROM is provided in "PCI Configuration Space Registers" on page 79 and "EEPROM Interface Registers" on page 149. See "External Serial EEPROM Interface Timing" on page 186 for timing of the EEPROM interface.

EEPROM reads and writes transfer only 16 bits of data; therefore, the software needs to perform two read or write operations to transfer a full 32-bit double-word.

To perform an EEPROM read:

- Write the desired address into the "EEPROM Address Register (E_ADDR)" on page 149.
- Write a '1' to bit [0] of the "EEPROM Command Register (E_CMD)" on page 150.
- Poll the "EEPROM Status Register (E_STS)" on page 150 until either the E_READ_DONE or E_READ_ERR bit is set.

  If the E_READ_DONE bit is set in E_STS, then the read data is available in the "EEPROM Data Register (E_DATA)" on page 149.

  If the E_READ_ERR bit is set, then the read failed for one of four reasons:

  – The EEPROM protection entry disallowed the read.

  – A violation of the EEPROM interface protocol occurred while communicating with the EEPROM and, as a result, the read operation could not be completed.

  – The E_SIZE field of the "EEPROM Configuration Register (E_CFG)" on page 151 is set to use the automatically determined EEPROM size, but the automatically determined EEPROM size is unknown (see EEPROMSIZE, bits [4:3], in "PCI Clock Domain Configuration/Status Register (PCICFG)" on page 142

  – The address in the E_ADDR register is out of range for the selected EEPROM size.

To perform an EEPROM write:

- Write the desired address into the "EEPROM Address Register (E_ADDR)" on page 149.
- Write the desired data into the "EEPROM Data Register (E_DATA)" on page 149.
- Write a '1' to bit [1] of the "EEPROM Command Register (E_CMD)" on page 150.
- Poll the "EEPROM Status Register (E_STS)" on page 150 until either the E_WRITE_DONE or E_WRITE_ERR bit is set.

  If the E_WRITE_DONE bit is set in E_STS, then the write completed normally.

  If the E_WRITE_ERR bit is set, then the read failed for one of three reasons:

  – The EEPROM protection entry disallowed the write.

  – A violation of the EEPROM interface protocol occurred while communicating with the EEPROM and, as a result, the write operation could not be completed.

  – The E_SIZE field of the "EEPROM Configuration Register (E_CFG)" on page 151 is set to use the automatically determined EEPROM size, but the automatically determined EEPROM size is unknown (see EEPROMSIZE, bits [4:3], in "PCI Clock Domain Configuration/Status Register (PCICFG)" on page 142.

  – The address in the E_ADDR register is out of range for the selected EEPROM size.

NOTE: The EEPROM requires as much as 15 ms to perform a write. The EEPROM logic is structured so that the E_WRITE_DONE bit in the E_STS register will not be asserted until the EEPROM signals that the write has completed.

- Software should expect a long delay between the initiation of an EEPROM write operation and its completion, as signalled by the E_WRITE_DONE bit in E_STS.
- Once the E_STS register indicates that the write has completed, software may initiate another EEPROM operation (a read or write) immediately.

The EEPROM contents are divided into three sections:

- A 64-entry (128-byte) section that contains the initial values for various PCI configuration registers, as well as the EEPROM read/write protection key, described in the following section.
- A 128-entry (256-byte) section that contains the Card Information Structure (CIS).
- A variable length section available for OEM use. The length is dependent on the EEPROM size:
  - A 64-entry (128-byte) for the 4-Kb part
  - A 320-entry (640 byte) for the 8-Kb part
  - An 832-entry (1664 byte) for the 16-Kb part

  Items such as the card's MAC address, serial number, calibration information, and other vendor-specific data are expected to be stored in this section.

2.2.4 EEPROM PCI/CardBus Partition

The PCI configuration register values are shown in Table 2-2.

The default is the value the register will assume if the EEPRM_EN_L pin is tied high to disable loading from the EEPROM at chip cold reset.

If EEPRM_EN_L is asserted, values in the EEPROM will be loaded into the corresponding PCI/CardBus configuration registers.

The EEPROM_MAGIC value at address 0x3D acts as a sanity check for the EEPROM contents and the physical PCB connections between AR5212 and the EEPROM. On exit from reset, the AR5212 EEPROM logic first will read the EEPROM_MAGIC entry. If the value of this entry is equal to 0x5aa5, then the EEPROM contents are assumed to be valid and the EEPROM logic will proceed to load the remainder of the EEPROM. If, however, the EEPROM_MAGIC value is not equal to 0x5aa5, then the EEPROM logic will assume that the EEPROM is either unprogrammed or corrupt. When this occurs, the EEPROM logic will cease reading from the EEPROM and will assume the default values as if the EEPRM_EN_L input to the AR5212 was de-asserted.

Table 2-2. **EEPROM Address, Parameters, Corresponding Registers and Default Values**

EEPROM Address	Parameters	Corresponding PCI/CardBus Register	Default
0x00	DEVICE_ID[15:0]	Device ID	0x1107
0x01	VENDOR_ID[15:0]	Vendor ID	0x168C
0x02	CLASS_CODE[23:8]	Class Code	0x0200
0x03	CLASS_CODE[7:0], REVISION_ID[7:0]	Class Code, Revision ID	0x0001
0x04	Reserved (bits 15 to 8), HD_TYPE[7:0]	Header Type	0x0000
0x05	CIS_PTR[15:0]	CIS Pointer	0x5001
0x06	CIS_PTR[31:16]	CIS Pointer	0x0000

Table 2-2. **EEPROM Address, Parameters, Corresponding Registers and Default Values**

EEPROM Address	Parameters	Corresponding PCI/CardBus Register	Default
0x07	SSYS_ID[15:0]	Subsystem ID	0x0000
0x08	SSYS_VEND_ID[15:0]	Subsystem Vendor ID	0x0000
0x09	MAX_LAT[7:0], MIN_GNT[7:0]	Max_Lat, MinGnt	0x0000
0x0A	INT_PIN[7:0], Reserved (bits 8 to 0)	Interrupt Pin, Reserved	0x0100
0x0B	Reserved	—	—
0x0C	PM_CAP[31:16]	CFG_PMCAP	0x0002
0x0D	Reserved (bits 15 to 2), PM_DATA_SCALE[1:0]	Reserved, CFG_PMCSR	0x0002
0x0E	PM_DATA_D0[7:0], PM_DATA_D3[7:0]	CFG_PMDATA, CFG_PMDATA	0x8003
0x0F	Reserved (bits 15 to 7), RFSILENT_FORCE, SLEEP_CLK_SEL, RFSILENT_GPIO_SEL[2:0][1], RFSILENT_POLARITY[1], CLKRUN_ENABLE[2]	—	0x0001
0x10-0x1C	Reserved	—	—
0x1D	MAC_ID[15:0]	—	N/A
0x1E	MAC_ID[31:16]	—	N/A
0x1F	MAC_ID[47:32]	—	N/A
0x20-0x3C	Reserved	—	—
0x3D	EEPROM_MAGIC[15:0]	—	N/A
0x3E	Reserved (bits 15 to 1), KEY_TABLE_RD_PROTECT	—	0x0000
0x3F	EEPROM_PROTECT[15:0]	—	0x0000

[1]Refer to "RFSilent" on page 32.
[2]Refer to "PCI Clkrun" on page 35 for details on CLKRUN_ENABLE.

2.2.5 CIS Tuples Partition

EEPROM addresses 0x40-0xBF contain the card information structure (CIS) tuples that define various CardBus capabilities. They are read by the host at boot time to guide hardware initialization and driver selection. Tuple 0 starts at 0x40 and each tuple takes two addresses, with the even address containing bits 15:0 and the odd address containing bits 31:16.

Information of the tuples is loaded sequentially into the on-chip tuple memory (32-bit wide) after chip reset. (Refer to the "AR5212 Address MAP" on page 22).

2.2.6 Vendor Data Partition

EEPROM addresses 0xC0 and above are allocated for vendor data. The vendor data can only be accessed using the E_ADDR, E_DATA, and E_CMD EEPROM access registers. The amount of memory available for vendor data depends on the EEPROM size shown in Table 2-3.

Table 2-3. **EEPROM Size**

Address	Contents
0xC0-0xFF	Vendor data (4-Kb EEPROM)
0xC0-0x1FF	Vendor data (8-Kb EEPROM)
0xC0-0x3FF	Vendor data (16-Kb EEPROM)

2.2.7 EEPROM Autosizing Mechanism

As noted above, the first procedure on exit from reset is to read the EEPROM_MAGIC entry. The physical presence and programmed state of the EEPROM, as well as its size, are determined automatically.

To determine the EEPROM size, the hardware proceeds through the set of supported EEPROM sizes (4-Kb, 8-Kb, and 16-Kb sizes are supported), starting at the 16-Kb size and proceeding to 4-Kb and then 8-Kb.

For each size, the logic then attempts to read the EEPROM_MAGIC entry at address 0x3D, using the currently assumed EEPROM size. If the EEPROM_MAGIC entry is read correctly, then the logic assumes that the EEPROM is present and that it has correctly determined its size. The logic now proceeds to load the remainder of the EEPROM. If the EEPROM_MAGIC value is incorrect, the logic retries the read using the next supported EEPROM size.

If the EEPROM_MAGIC value cannot be read correctly for any of the supported EEPROM sizes, then the logic assumes the EEPROM is not present on the PCB, or is present but unprogrammed. In either case, the logic will proceed to use the default values as described in the EEPROM section descriptions above.

2.2.8 EEPROM Read/Write Protection Mechanism

The EEPROM contains a 16-bit EEPROM_PROTECT value at address 63 (0x3F). This mask consists of eight 2-bit submasks. Each submask covers a portion of the overall 256/512/1024-entry EEPROM address space. (See Table 2-4, Table 2-5, and Table 2-6).

- A submask can have four values that determine the access types permitted to the associated protection region:
  - 00: read and write access allowed
  - 01: write-only access allowed
  - 10: read-only access allowed
  - 11: no access allowed
- In general, bits [3:2] of EEPROM_PROTECT that control access to the EEPROM_PROTECT entry itself, are set to deny write access.
- Note that EEPROM_PROTECT is read automatically by the HW, and therefore, SW cannot override it, other than by rewriting the EEPROM itself to contain a different protection mask.
- Rewriting EEPROM_PROTECT requires either that bits [3:2] in EEPROM_PROTECT already be set to allow write access, or that the EPRM_EN_L pin of the AR5212 be de-asserted (high).
- In production use, the expected situation would be to write-protect the protection region that contained the protection mask EEPROM_PROTECT and to tie the EPROM_EN_L pin low (asserted) on the PCB. This prevents modifications to the protection mask by any means.

Table 2-4. **EEPROM_PROTECT Mapping for 4-Kb EEPROM**

EEPROM Entry (Word)	EEPROM_PROTECT bits	Contents
0 to 31	1:0	Configuration registers
32 to 63	3:2	Reserved EEPROM_PROTECT
64 to 127	5:4	CIS tuples 0-31
128 to 191	7:6	CIS tuples 32-63
192 to 207	9:8	OEM data (reserved for Atheros)
208 to 223	11:10	OEM data (reserved for Atheros)
224 to 239	13:12	OEM data (reserved for Atheros)
240 to 255	15:14	OEM data (reserved for Atheros)

Table 2-5. **EEPROM_PROTECT Mapping for 8-Kb EEPROM**

EEPROM Entry (Word)	EEPROM_PROTECT bits	Contents
0 to 31	1:0	Configuration registers
32 to 63	3:2	Reserved EEPROM_PROTECT
64 to 127	5:4	CIS tuples 0-31
128 to 191	7:6	CIS tuples 32-63
192 to 255	9:8	OEM data (reserved for Atheros)
256 to 319	11:10	OEM data
320 to 383	13:12	OEM data
384 to 511	15:14	OEM data

Table 2-6. **EEPROM_PROTECT Mapping for 16-Kb EEPROM**

EEPROM Entry (Word)	EEPROM_PROTECT bits	Contents
0 to 31	1:0	Configuration registers
32 to 63	3:2	Reserved EEPROM_PROTECT
64 to 127	5:4	CIS tuples 0-31
128 to 191	7:6	CIS tuples 32-63
192 to 255	9:8	OEM data (reserved for Atheros)
256 to 511	11:10	OEM data
512 to 767	13:12	OEM data
768 to 1023	15:14	OEM data

2.3 PLL

The AR5212 requires a 32 MHz clock. This clock is provided by the AR2111 using the REFCLK input to the PLL. Table 2-7 shows the various clock frequencies that are generated, based on the 32 MHz clock, for the functional blocks within the AR5212.

Table 2-7. **Operating Frequency**

Frequency	Functional Block
0-33 MHz	PCI clock domain registers (see “PCI Clock Domain Registers” on page 139).
40 MHz	Core operating frequency when operating in IEEE 802.11a mode.
22 MHz	Core operating frequency when operating in IEEE 802.11b mode.
44 MHz	Core operating frequency when operating in IEEE 802.11g mode.
80 MHz	Core operating frequency when operating in enhanced turbo mode.

When the PLLBYPASS pin is active (high) the REFCLK bypasses the PLL and is directly used as the clock for test.

2.4 Reset

AR5212 reset is controlled by the host using the PCI_RST_L pin or using the “Reset Control Register (RC)” on page 140. The reset caused by the PCI_RST_L signal is referred to as the cold reset, and the reset caused by setting the appropriate bits inside the RC register is referred to as the warm reset. The PCI/CardBus interface itself can be reset by the host writing to the RPCI bit inside the reset control register. This causes the PCI/CardBus interface to reset for 64 PCI clock cycles, after which the RPCI bit is automatically cleared. The RFRESET_L pin will be set low whenever the PCI_RST_L pin is pulled low or when RBB is set, putting the baseband in warm reset.

- The function of each RC register bit is provided.
  - RMAC bit—Writing a '1' to this bit will put the MAC block in warm reset. The PCI/CardBus Interface and the baseband logic are unaffected.
  - RBB bit—Writing a '1' to this bit will put the baseband logic and AR5111/AR2111 in warm reset. The MAC and PCI/CardBus Interface are unaffected.
  - RPCI bit—Writing a '1' to this bit will cause the logic to assert a warm reset to the PCI/CardBus Interface for 64 clocks, and then automatically remove the reset so that software can once again have access to the AR5212.

  Reset of the PCI/CardBus Interface affects all PCI/CardBus Interface logic, except the “PCI Configuration Space Registers” on page 79. The PCI configuration registers are not reset, and the EEPROM data is not reloaded. Only a true cold reset, that is, an assertion of the PCI_RST_L pin (see page 15), affects the PCI configuration registers and causes a load from the EEPROM.

- The recommended practice for AR5212 is to reset both the PCI/CardBus Interface and the MAC at the same time.
- Software must adhere to the restrictions described below to ensure reliable reset operation.

  When the RC register is written, no other AR5212 register accesses may be active. Because AR5212 posts register writes, if software cannot guarantee that all earlier register writes are complete, then software must read any direct memory access (DMA) register to ensure that all pending register writes are complete, such as the register “Receive Queue Descriptor Pointer (RXDP)” on page 94. Only after a DMA register read completes may software write to the RC register.

  After writing the new value to the RC register, software must read (poll) the RC register until its value matches the value just written. Normally this will take less than a millisecond, but can take up to 2 milliseconds. Note that when polling RC, the RPCI bit always will read back as zero, even if the write to RC sets the RPCI bit.

This occurs because the hardware automatically clears this bit when the PCI/CardBus Interface reset completes.

Recommended flow to bring the MAC into warm reset:

1. If active register writes are possible, read the "Receive Queue Descriptor Pointer (RXDP)" on page 94 to flush any pending register writes.
2. Set the RMAC and RPCI bits in the "Reset Control Register (RC)" on page 140 (write a value of 0x11). If the baseband is to be reset as well (the typical case), then a value of 0x13 must be written to RC instead.
3. Delay for at least 10 ms to allow the reset of the PCI/CardBus Interface to complete.
4. Poll RC until its value is 0x1 (or 0x3, as appropriate). Note again that the expected value for RC has the RPCI bit as zero.

The MAC (and baseband if its bit was set) is now in warm reset.

Recommended flow to take the MAC out of warm reset:

1. Write a value of 0x0 to RC.
2. Poll RC until its value is 0x0.

The MAC and baseband are now ready for use.

2.5 AR5111/AR2111 Interface

A serial interface bus is provided to set the control registers inside both the AR5111 and AR2111. The 32 MHz clock required by the interface is provided by the AR5111. Reset of the AR5111 and AR2111 is controlled using the RFRESET_L pin of the AR5212, which in turn is controlled by the PCI_RST_L pin or RBB bit in the RC Register.

2.6 GPIO

The AR5212 provides six bi-directional general purpose I/O ports. Each GPIO can be independently configured as input or output using the GPIO control register (see "GPIO Control Register (GPIOCR)" on page 144). Information presented at the GPIO inputs can be read from the GPIO data input register (GPIODI), (see "GPIO Data Input Register (GPIODI)" on page 146). GPIO outputs are driven from the GPIO data output register (GPIODO), (see "GPIO Data Output Register (GPIODO)" on page 145).

2.7 LED

Two output pins are provided for LED control. LED_0 can be used to indicate the power supply status or network activity, and LED_1 can be used to indicate network event status. Control of the LED outputs is provided by the PCI clock domain register (see "PCI Clock Domain Configuration/Status Register (PCICFG)" on page 142).

Table 2-8 depicts the functionality that can be achieved by the LED_0 and LED_1 outputs with the control of software.

NOTE: In the table, a setting X indicates that the setting does not affect any changes.

Table 2-8. **LED Functionality**

LEDCTL[1:0]	Description	LED_0	LED_1
XX	Power save mode (default from power up or reset)	Slow-rate blink	OFF
00	Awake from power save mode, can be used to indicate power is applied. The hardware automatically enter this state after exit from power save mode before any other activity. Changes from power save mode to this state might not be visible on the LEDs if the software assumes control of the LED blinking by writing to the LEDCTL bits.	ON	OFF
01	Looking for network association	Alternate blink between LED_0 and LED_1	Alternate blink between LED_1 and LED_0
10	Associated or joined with network; no activity LEDSLOW=0	Slow-rate blink	Slow-rate blink

Table 2-8. **LED Functionality (continued)**

10	Associated or joined with network; blink rate increases with activity LEDMODE[2:0] =000, blink LEDs proportional to the count of all Tx bytes and those Rx bytes that pass the Rx filter (default from power up or reset) LEDMODE[2:0] =001, blink rate proportional to bytes/second on network activity over the air LEDMODE[2:0] =010, blink LED_0 for each Tx byte and LED_1 for each Rx byte Blink rate controls by LEDBLINK[2:0]	blink	blink
XX	Power off or PCI/CardBus slot disabled	OFF	OFF

2.8 *Antenna Controls (Switching)*

The selection on which antenna to transmit or receive is made by the PCU. The algorithm was designed to support two antenna schemes:

1. Dual omni-directional antenna (AP or STA)
2. Up to 14 sectored and 1 omni antenna (AP only)

The omni AP, omni STA, and sectored AP modes are globally controlled by the antenna control fields listed in Table 2-9. Placing the device into a particular mode consists of setting the four fields to the values listed in Table 2-10 on page 31.

Table 2-9. **Antenna Control Registers**

Field	Size (bits)	Description
DEFANT (See register "Default Antenna (DEF_ANTENNA)" on page 164.)	4	When listening for the start of a frame sequence, DEFANT selects either Rx antenna 1 or 2. When transmitting, DEFANT selects the Tx antenna if "USE_DEFANT" on page 154 is set.
USE_DEFANT (See register "STA Address 1 (STA_ID1)" on page 154.)	1	When AntModeXmit in the Tx descriptor is 0: 0 = Use LAST_TX_ANT in the key cache. 1 = Use the LSB of DEF_ANTENNA.
DEFANT_UPDATE (See register "STA Address 1 (STA_ID1)" on page 154.)	1	0 = Do not update the DEF_ANTENNA after each Tx frame. 1 = Update the DEF_ANTENNA after each Tx frame.
RTS_USE_DEF (See register "STA Address 1 (STA_ID1)" on page 154.)	1	0 = Transmit RTS on the antenna selected by the descriptor. 1 = Transmit RTS on the DEF_ANTENNA.

Per packet Tx antenna is controlled by the AntModeXmit field of the "DMA Tx Descriptor, Words 2, 3, 4 and 5," on page 39, which is always set to zero for the above bits to have effect and to be able to use DEFANT for transmit.

Listening is always conducted using the DEFANT. This is also true for transmitting ACK. After transmitting a directed frame, antenna selection is locked to the Tx antenna to receive the responding ACK so that the entire frame exchange is conducted under the same antenna. CTS is always received on the same antenna as the data frame following it.

2.8.1 Omni AP Mode

When operating in omni AP mode, the last Tx antenna for the destination STA is selected (held in the key cache) and toggles every other Tx failure. This same antenna is used when listening for CTS or ACK.

DEFANT controls which antenna is selected when listening for the start of a frame sequence from a STA. It is up to the AP SW to change this selection by writing to DEFANT. See Table 2-10 for mode configuration.

2.8.2 Omni STA Modes

Operating in omni STA modes is similar to omni AP mode, except the last Tx antenna is recorded in the DEF_ANTENNA. Omni STA Mode II uses keycache to store which antenna to use for transmitting and the Default Antenna register stores which antenna to use for receive. This allows use of hardware fast transmit switching while leaving software to determine the receive end. RTS would be transmitted on the same antenna as that used by the associated data frame. See Table 2-10 for mode configuration.

2.8.3 Sectored AP Mode

For sectored AP mode, SW would have complete control of Tx antenna selection through setting the AntModeXmit in the Tx descriptor to the desired antenna. The 1 omni antenna is used for listening and transmitting RTS. See Table 2-10 on page 31 for mode configuration.

Table 2-10. **Mode Configuration**

Fields	Omni AP Mode	Omni STA Mode	Omni STA Mode II	Sectored AP Mode	Sectored STA Mode
AntModeXmit	0000	0000	0000	0001–1110	0001-1110
USE_DEFANT	0	1	0	0	1
DEFANT	0001 or 0010	0001/0010	0001/0010	0001/0010	0001-1110
DEFANT_UPDATE	0	1	0	0	0
RTS_USE_DEF	0	1	0	1	1

2.8.4 Switching

The switch table (see Table 2-11) contains 11 entries, each of which is 6 bits wide, and is indexed by:

- The antenna selected by the MAC.
- The state of the transceiver (idle, receive, or transmit).
- Controls for two external Rx attenuation.

When fast-receive antenna diversity is enabled for 802.11b operation, the baseband will temporarily override the antenna selected by the MAC once a packet has been detected.

Table 2-11 also shows location of the registers.

Table 2-11. **Switch Table**

Chip State	Ant Select	1st Rx Atten	2nd Rx Atten	Register Location (address and bits)	Register Name
idle	—	—	—	0x9910, bits [9:4]	BB_ANTENNA_CONTROL
Tx	1	—	—	0x9960, bits [5:0]	BB_SWITCH_TABLE1
Rx	1	no	no	0x9960, bits [11:6]	
Rx	1	yes	no	0x9960, bits [17:12]	
Rx	1	no	yes	0x9960, bits [23:18] (unused)	
Rx	1	yes	yes	0x9960, bits [29:24] (only 2.4 GHz)	

Table 2-11. **Switch Table**

Chip State	Ant Select	1st Rx Atten	2nd Rx Atten	Register Location (address and bits)	Register Name
Tx	2	—	—	0x9964, bits [5:0]	BB_SWITCH_TABLE2
Rx	2	no	no	0x9964, bits [11:6]	
Rx	2	yes	no	0x9964, bits [17:12]	
Rx	2	no	yes	0x9964, bits [23:18] (unused)	
Rx	2	yes	yes	0x9964, bits [29:24] (only 2.4 GHz)	

Each 6-bit register controls the following AR5212 outputs:

- ATTEN5
- ATTEN2
- ANTD
- ANTC
- ANTB
- ANTA

The most significant bit of the register is ATTEN5. The least significant bit of the register is ANTA. ATTEN5 and ATTEN2 control the external attenuation. ANTD, ANTC, ANTB, and ANTA select antenna 1 or antenna 2. The actual signals used are application-specific (refer to the *AR5001 Reference Design Schematics*, *AR5001 Manufacturing Library Reference*, and the *AR5001 Sample Manufacturing Test Flow* for implementation and configuration of the antenna control signals).

In the 5 GHz band, only the first external attenuation is supported. The second will never be enabled, therefore the second stage enabled table entries are unused at 5 GHz.

In the 2.4 GHz band, two external attenuations are supported. As the received signal strength increases, the first Rx attenuation will be enabled. As the strength of the Rx signals increases even more, the second Rx attenuation will be enabled.

Reset will cause the AR5212 to enter the idle state. Bits [9:4] of BB_ANTENNA_CONTROL are reset to all zeros, and will be applied to the six outputs of the AR5212.

Register BB_ANTENNA_CONTROL contains bits other than [9:4]. Therefore, unless the other bits are known from initialization, it is recommended that bits [9:4] be altered with a read-modify-write cycle.

2.9 RFSilent

The RFSilent support enables a GPIO pin to force the radio frequency (RF) logic to halt transmission immediately and enter a reset condition. Two values in the EEPROM control which GPIO pin is used and select the polarity of the input.

The RFSILENT_GPIO_SEL[2:0] field in word 0xF, bits 2 to 4, of the EEPROM determines which of the GPIO pins will be used as the RFSilent input. Values from 0-5 are valid, corresponding to GPIO pins 0-5, respectively.

Note that in addition to specifying the GPIO pin number in the RFSILENT_GPIO_SEL value, software also must configure the selected GPIO pin as an input. This is done by setting the appropriate driver control bits of the GPIO control register (GPIOCR) such that the RFSilent pin is configured as an input.

The RFSILENT_POLARITY field in word 0xF, bit 1, of the EEPROM determines whether the RF logic is forced into reset when the selected GPIO pin is at a high voltage or a low voltage.

Setting RFSILENT_POLARITY to zero means that the RF logic will be forced into reset when the GPIO pin selected by the field RFSILENT_GPIO_SEL is at a low voltage.

Setting RFSILENT_POLARITY to one means that the RF logic will be forced into reset when the GPIO pin selected by the field RFSILENT_GPIO_SEL is at a high voltage.

3. PCI/CardBus Interface Description

This section provides a summary of the AR5212 PCI/CardBus interface. This interface is compatible with PCI 2.3 and PC Card 7.1 standards, and functions as the host interface for the AR5212, providing data and command transfer between the host software, the MAC, and the configuration registers. For details, refer to the PCI 2.3 and CardBus 7.1 standards specifications.

3.1 PCI/CardBus Registers

The PCI/CardBus configuration registers are used at system boot time for the host to detect the type of device present, and to perform low-level PCI/CardBus configuration, such as assigning a base address to the device. An external serial EEPROM provides configuration information for the device. At reset, some of the PCI/CardBus configuration registers are loaded from the off-chip serial EEPROM, whereas others must be programmed by the host. Configuration, control, and status registers for the various functional blocks of the AR5212 are mapped to the memory space of the PCI/CardBus interface, and thus can be accessed by the host. The PCI configuration registers are provided in detail in "PCI Configuration Space Registers" on page 79.

3.2 Signal Descriptions

The AR5212 PCI/CardBus interface pins are described in"Pin Descriptions" on page 13. Table 3-1 shows the interface pins grouped by functional types. One of these signals, PCI_MODE, is used to select either a PCI bus interface configuration or a CardBus interface configuration. As shown in Table 3-1, all other signals are common to both a CardBus or standard PCI interface, with the exception of PCI_IDSEL which is used only for PCI applications. The power management wake-up signaling (PCI_PME_L/CSTCHG) is not supported by the AR5212 for either bus configuration. The PCI/CardBus interface is active during all power management states.

Table 3-1. **Types of Interface Signals**

Signal Name	Signal Type	Use	I/O	Description
PCI_AD[31:0]	Address and data	Both	I/OL	PCI_AD[31:0] is a multiplexed address and data bus.
PCI_CBE[3:0]_L		Both	I/OH	PCI multiplexed bus command and byte enables. Active low. The AR5212 supports the following commands as bus manager and target device: 0110 = memory read 0111 = memory write 1100 = memory read multiple 1110 = memory read line 1111 = memory write and invalidate In addition, as target device the AR5212 also supports the configuration commands: 1010 = configuration read 1011 = configuration write
PCI_PAR		Both	I/OH	PCI parity.
PCI_RST_L	System	Both	IL	PCI reset, reset the AR5212. Active low.
PCI_CLK		Both	IL	PCI bus clock.
PCI_CLKRUN_L		Both	IH	PCI bus clock run. Active low. Provides for starting and stopping the PCI clock. Refer to "PCI Clkrun" on page 35.

Table 3-1. **Types of Interface Signals (continued)**

Signal Name	Signal Type	Use	I/O	Description
PCI_IDSEL	Interface control	PCI only	IL	PCI ID select. Not used by CardBus, external 10K pull-up resistor required for CardBus application.
PCI_DEVSEL_L		Both	I/OH	PCI dev select. Active low.
PCI_FRAME_L		Both	I/OL	PCI frame. Active low.
PCI_IRDY_L		Both	I/OH	PCI initiator ready. Active low.
PCI_STOP_L		Both	I/OH	PCI stop. Active low.
PCI_TRDY_L		Both	I/OH	PCI target ready. Active low.
PCI_GNT_L	Arbitration	Both	IH	PCI grant. Active low.
PCI_REQ_L		Both	OH	PCI request. Active low.
PCI_INT_L	Interrupt	Both	OH	PCI interrupt. Active low.
PCI_SERR_L	Error reporting	Both	I/OH	PCI system error. Active low.
PCI_PERR_L		Both	I/OH	PCI parity error. Active low.
PCI_PME_L/CSTSCHG	Event notification	Both	OH	PCI - power management event. Active low. CardBus - CSTSCHG. Active high (These signals are not supported in the AR5212).

PCI_MODE	AR5212	Bus type select	IL	Select PCI or CardBus interface
				L = CardBus (default - internal pull down)
				H = PCI

3.3 Host Interface Unit Interrupts

The MAC provides per-QCU Tx interrupts. This means that each QCU generates the following four interrupts:

- TXOK — A frame was sent successfully
- TXERR — A frame could not be sent successfully (retry limit reached, etc.)
- TXDESC — A frame was sent (successfully or not) and the InterReq bit in the frame's Tx descriptor was set
- TXEOL — The QCU has reached the physical end of the Tx descriptor list (generated only by reaching a descriptor with a NULL LinkPtr; not generated just because the VEOL bit in the Tx descriptor was set)

With 10 QCUs, this leads to 40 Tx-related interrupts. Because the maximum size of an atomic register read is 32 bits, the MAC provides hardware support for simulating an atomic read of an interrupt status register (ISR) that is more than 32 bits wide.

To do so, the MAC provides several ISRs: a single primary ISR and several secondary ISRs. The primary ISR contains all of the bits except the Tx-related bits (TXOK, TXERR, TXDESC, TXEOL). The Tx-related bits are generated by logically ORing the corresponding QCU bits from the appropriate secondary ISR, after they are masked with the corresponding bits from the secondary interrupt mask register. Software can check the non-Tx-related interrupts and determine whether any Tx-related bits are set in the secondary ISRs with a single read of the primary ISR. In many cases, the software does not need to read the secondary ISRs; just knowing that some bits are set often is sufficient. The same logical ORing is used for several other ISR bits as well.

In addition, to make the read of all ISRs appear atomic, the MAC implements shadow copies of all the secondary ISRs. On the same cycle in which software reads the primary ISR, the MAC copies the contents of all secondary ISRs into the shadow registers. Software can then read the shadow copies of the secondary ISRs and receive a consistent view of the overall ISR state when the primary ISR was read, thereby simulating an atomic read of all ISRs.

The MAC provides two ways to access the primary and secondary ISRs:

- Write-one-to-clear access

  When used, reads of the ISRs neither copy data to the shadow copies nor clear the ISR being read.

  Software can write to both the primary ISR and to the secondary ISRs. For each such write, the ISR bits for which the write data bit is a one are cleared. ISR bits for which the write data is a zero are unaffected. A write of one to a bit that is a logical OR of bits in a secondary ISR will clear the secondary ISR bits from which the primary ISR bit is generated.

  For example, a write of a one to the TXOK bit (bit 6) in the "Primary Interrupt Status Register (ISR_P)" on page 103 will clear all 10 TXOK bits in "Secondary Interrupt Status Register 0 (ISR_S0)" on page 105 (bits 9:0 of ISR_S0).

- Read-and-clear access.

  When used, only the primary ISR may be read. Each read of the primary ISR triggers a copy into the shadow registers, as described above, and clears all primary and secondary ISR bits as well, all as a single atomic operation. Writes to the primary and secondary ISRs are ignored in this mode.

Software may intermix write-one-to-clear and read-and-clear ISR accesses.

See "Host Interface and Receive Registers" on page 91 for information on the interrupt registers.

3.4 PCI Clkrun

The AR5212 supports the optional PCI clkrun capability, as described in the *PCI Mobile Design Guide, Version 1.1*. PCI_CLKRUN_L is a PCI interface signal that controls the state of the PCI clock as supplied by the PCI host. It is used by both the PCI host and the PCI device to signal that a PCI transaction is starting. This, in turn, forces the host to maintain the PCI clock if it is running, or start the PCI clock if is halted, allowing either the PCI host or PCI device to start the PCI clock prior to each PCI transaction and stop the clock when the transaction is complete. The usage of PCI_CLKRUN_L is transparent to the AR5212 operation and can be enabled or disabled at any time. When enabled, the impact on system performance is negligible but power savings can be significant.

The PCI_CLKRUN_L signal is controlled by the CLKRUNEN field of the "PCI Clock Domain Configuration/Status Register (PCICFG)" on page 142. When cleared, PCI_CLKRUN_L is forced low and the AR5212 forces the PCI clock to run continuously. When set, the AR5212 requests the PCI clock to run only when a PCI transaction is pending and allows the system to halt PCI clock at other times.

The CLKRUN_ENABLE field, in word 0xF, bit 0 of the EEPROM (see Table 2-2 on page 24) controls the initial value of CLKRUNEN following reset. If clear, CLKRUNEN will be clear following reset. If set, clkrun will be enabled following reset, allowing the lowest possible power usage prior to initialization. While reset is asserted, PCI_CLKRUN_L will be forced low.

4. Medium Access Control (MAC)

The MAC consists of the following major functional blocks: 10 queue control units (QCUs), 10 distributed coordination function (DCF) control units (DCUs), a single DMA receive unit (DRU), and a single protocol control unit (PCU). See Figure 4-1.

Functionality of the MAC block includes:

- Tx frame data transfer from the host to the AR5212 using the PCI bus.
- Rx frame data transfer from the AR5212 to the host using the PCI bus.
- Register access to all AR5212, AR5111, and AR2111 registers.
- Interrupt generation and reporting.
- Sleep mode (power-down) sequencing.
- Miscellaneous error and status reporting functions.

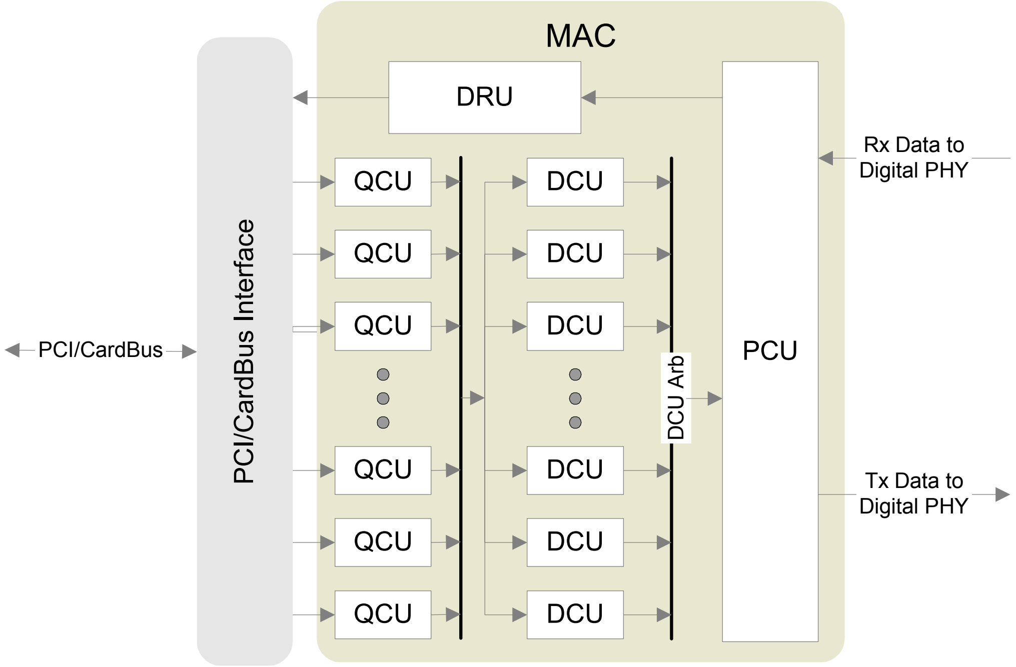


Figure 4-1. **MAC Block Diagram**

4.1 Overview

The MAC block supports full bus-mastering descriptor-based scatter/gather DMA.

Frame transmission begins with the QCUs. QCUs manage the DMA of frame data from the host through the PCI/CardBus Interface, and determine when a frame is available for transmission (see "Queue Control Unit (QCU)" on page 53).

Each QCU targets exactly one DCU. Ready frames are passed from a QCU to its targeted DCU. The DCU manages the enhanced distributed coordination function (EDCF) channel access procedure on behalf of the QCUs associated with it.

Once the DCU gains access to the channel, it passes the frame to the PCU, which encrypts the frame and sends it to the baseband logic. The PCU handles both processing responses to the transmitted frame, and reporting the transmission attempt results to the DCU.

Frame reception begins in the PCU, which receives the incoming frame bitstream from the digital PHY. The PCU decrypts the frame and passes it to the DRU, which manages Rx descriptors and writes the incoming frame data and status to the host memory through the PCI/CardBus Interface.

4.2 Descriptor

The MAC is responsible for transferring frames between the host memory (accessed using the PCI/CardBus interface) and the AR5212. For all normal frame send/receive activity, the host provides a series of descriptors to the MAC, and the MAC then parses the descriptors and performs the required set of data transfers.

See “Descriptor Usage” on page 49 for detailed information on descriptor usage and processing.

4.3 Descriptor Format

The transmit descriptor format is composed of eight 32-bit words and the receive descriptor is composed of six, 32-bit words, shown in Table 4-1. The first two words of the descriptor point to the next descriptor in the linked list, and to the data buffer associated with the descriptor. The next two or four words carry additional control information that affects how the MAC processes the frame and its data. The final two words are used by the MAC to report status information back to the host.

Table 4-2 and Table 4-3 provide Tx and Rx descriptor control information, respectively. Table 4-5 and Table 4-4 provide descriptions of the status information for the Tx and Rx descriptors, respectively. A descriptor is required to be aligned on a 32-bit boundary in host memory, although best performance is achieved if the descriptor is aligned on a cache-line boundary.

Table 4-1. **DMA Descriptor Format**

Word	Bits	Field Name	Description
0	31–0	LinkPtr	Link pointer. Contains the address of the next descriptor to be used. Must be 32-bit aligned (bits 1–0 must be 0).
1	31–0	BufPtr	Data buffer pointer. Contains the starting address of the data buffer associated with this descriptor. A Tx data buffer can begin at any byte address. A Rx data buffer must be aligned on a 32-bit boundary in host memory, although best performance is achieved if the Rx data buffer is cache-line aligned. (Cache-line size varies from system to system.)
2–5 (transmit) 2–3 (receive)	31–0	Host-to-DMA engine control information	Additional control information is passed from host to DMA engine. The format of these words varies depending on whether the descriptor is being used to transmit a frame from host to PCU, or Rx a frame from PCU to host. (See Table 4-2 on page 39, and Table 4-3 on page 44 for details.)
6, 7 (transmit) 4–5 (receive)	31–0	DMA completion status information	Status information reported by the DMA engine when it has finished processing a descriptor. As with the control information, the format of the status information differs between Tx and Rx descriptors. (See Table 4-5 on page 48, and Table 4-4 on page 45 for details.)

Table 4-2. **DMA Tx Descriptor, Words 2, 3, 4 and 5**

Word	Bits	Field Name	Description
2	11–0	FrameLen	Frame length. Specifies the length, in bytes, of the entire MAC frame, including the frame check sequence (FCS), initialization vector (IV), and integrity check value (ICV) fields. This field must appear in the descriptor for the first segment of a frame, and is ignored for all following descriptors of that frame.
	15–12	Reserved	Reserved for future use.
	21–16	TPC	Transmit power control. These bits are passed unchanged to the baseband, where they are used to control the transmit power for the frame.
	22	RTSEn	Request to send (RTS) enable. If set, the PCU transmits the frame using the RTS/CTS protocol. If clear, the PCU transmits the frame using the contention/backoff protocol. At most, one of the RTSEn and CTSEn bits may be set; it is illegal to set both bits.
	23	VEOL	Virtual end-of-list flag. When set, indicates that the QCU should stop processing frames. See "End of Queue/End of List Detection" on page 55 for more details. This field must appear in the final descriptor of a frame and must be clear for all other descriptors of the frame.
	24	ClearDestMask	Clear destination mask bit flag. 1 = Clear the Tx filter bit "Transmit Filter Data Register (D_TXBLK_DATA)" on page 136 at the index specified by the EncryptKeyIdx field (see "DestIdx") 0 = Not to clear Tx mask bit This field must appear in the descriptor for the first segment of a frame, and is ignored for all following descriptors of that frame.

Table 4-2. **DMA Tx Descriptor, Words 2, 3, 4 and 5 (continued)**

Word	Bits	Field Name	Description
2	28–25	TxAntMode	Transmit antenna selection mode: 0000 = Auto 0001 = Omni 0 0010 = Omni 1/Sector 0 0011 = Sector 1 0100 = Sector 2 0101 = Sector 3 0110 = Sector 4 0111 = Sector 5 1000 = Sector 6 1001 = Sector 7 1010 = Sector 8 1011 = Sector 9 1100 = Sector 10 1101 = Sector 11 1110 = Sector 12 1111 = Invert current antenna This field must appear in the descriptor for the first segment of a frame, and is ignored for all following descriptors of that frame.
	29	InterReq	Interrupt request flag. 1 = Generate an InterReq interrupt at the completion of this frame 0 = Do not generate an InterReq interrupt at the completion of this frame This field must be valid and identical for all descriptors of the frame. That is, all descriptors for the frame must have this flag set, or all descriptors for the frame must have this flag clear.
	30	DestIdxValid	Destination index valid flag. Specifies whether the contents of the DestIdx field are valid. 1 = Filter frame based on encrypt key. 0 = Do not filter frame based on encrypt key. This field must appear in the descriptor for the first segment of a frame, and is ignored for all following descriptors of that frame.
	31	CTSEn	Proceed frame with CTS flag. If set, the PCU first sends a CTS before sending the frame described by the descriptor. This is used for 802.11g to quiet legacy stations before sending a frame the legacy stations cannot interpret. At most, one of the RTSEn and CTSEn bits may be set; it is illegal to set both bits.

Table 4-2. **DMA Tx Descriptor, Words 2, 3, 4 and 5 (continued)**

Word	Bits	Field Name	Description
3	11–0	BufLen	Data buffer length. Specifies the length, in bytes, of the data buffer associated with this descriptor. This field must be valid for all descriptors.
	12	More	More descriptors in this frame flag. 1 = The current frame is continued in the next descriptor. 0 = No more descriptors for the current frame. This field must be valid for all descriptors.
	19–13	DestIdx	Destination table index. Specifies an index into an on-chip table of encryption keys, location of the Tx, address, and key. This field must appear in the descriptor for the first segment of a frame, and is ignored for all following descriptors of that frame.
	23–20	FrmType	Frame type indication. Indicates to the PCU what type of frame is being sent. Supported values: 000 = Normal frame. 001 = Announcement traffic indication message (ATIM) frame. 010 = PS poll frame. 011 = Beacon. 100 = Probe response frame. 101–111 = Reserved. This field must appear in the descriptor for the first segment of a frame, and is ignored for all following descriptors of that frame.
	24	NoAck	No ACK flag. 1 = Do not wait for ACK. This field must appear in the descriptor for the first segment of a frame, and is ignored for all following descriptors of that frame.
	31:25	RES	Reserved.

Table 4-2. **DMA Tx Descriptor, Words 2, 3, 4 and 5 (continued)**

Word	Bits	Field Name	Description
4	14–0	RTSCTSDur	RTS/CTS duration value. Specifies the contents of the Duration field in the MAC header for RTS and CTS frames. The MAC inserts this value into the Duration field of the RTS generated if the RTSEn flag is set, or into the Duration field of the CTS generated if the CTSEn flag is set.
	15	DurUpdateEn	Frame duration update control. If set, the MAC updates (overwrites) the Duration field in the frame based on the current transmit rate. If clear, the MAC does not alter the contents of the frame's Duration field. Note that the MAC changes only the frame's Duration field. It does not alter the Duration field in the RTS/CTS that precedes the frame itself if RTSEn or CTSEn is set. The Rate to Duration table must be filled with valid values when this bit is set.
	19–16	TXDataTries0	Number of frame data exchange attempts permitted for transmission series 0. A "frame data exchange attempt" means a transmission attempt in which the actual frame is sent on the air. This is in contrast to the case in which the frame has RTS enabled and the RTS fails to receive a CTS. In this case, the actual frame is not sent on the air, so this does not count as a frame data exchange attempt. A value of zero is illegal for the TXDataTries0 field.
	23–20	TXDataTries1	Number of frame data exchange attempts permitted for transmission series 1. A value of zero means skip this transmission series.
	27–24	TXDataTries2	Number of frame data exchange attempts permitted for transmission series 2. A value of zero means skip this transmission series.
	31–28	TXDataTries3	Number of frame data exchange attempts permitted for transmission series 3. A value of zero means skip this transmission series.

Table 4-2. **DMA Tx Descriptor, Words 2, 3, 4 and 5 (continued)**

<table>
<tr><th>Word</th><th>Bits</th><th>Field Name</th><th>Description</th></tr>
<tr><td>5</td><td>4–0</td><td>TXRate0</td><td>Transmit rate for transmission series 0.
Specifies the rate at which the frame should be transmitted:
<table>
<tr><th>MAC Rate Encoding</th><th>Protocol</th><th>Link Rate (Mbit/sec.)[1]</th></tr>
<tr><td>0x0b</td><td>802.11a/802.11g (OFDM)</td><td>6</td></tr>
<tr><td>0x0f</td><td></td><td>9</td></tr>
<tr><td>0x0a</td><td></td><td>12</td></tr>
<tr><td>0x0e</td><td></td><td>18</td></tr>
<tr><td>0x09</td><td></td><td>24</td></tr>
<tr><td>0x0d</td><td></td><td>36</td></tr>
<tr><td>0x08</td><td></td><td>48</td></tr>
<tr><td>0x0c</td><td></td><td>54</td></tr>
<tr><td>0x1b</td><td>802.11b (CCK)</td><td>1 L[2]</td></tr>
<tr><td>0x1a</td><td></td><td>2 L</td></tr>
<tr><td>0x1e</td><td></td><td>2 S</td></tr>
<tr><td>0x19</td><td></td><td>5.5 L</td></tr>
<tr><td>0x1d</td><td></td><td>5.5 S</td></tr>
<tr><td>0x18</td><td></td><td>11 L</td></tr>
<tr><td>0x1c</td><td></td><td>11 S</td></tr>
<tr><td>0x03</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x07</td><td></td><td></td></tr>
<tr><td>0x02</td><td></td><td></td></tr>
<tr><td>0x06</td><td></td><td></td></tr>
<tr><td>0x01</td><td></td><td></td></tr>
<tr><td>0x00, 0x04, 0x05, 0x010–0x17, 0x1f</td><td>Reserved</td><td>Reserved</td></tr>
</table>
[1] Rates are specified assuming non-turbo mode. When in turbo mode (for those protocols that support turbo mode), the link rate is double that listed in this table.
[2] L = with long preamble; S = with short preamble.</td></tr>
<tr><td></td><td>9–5</td><td>TXRate1</td><td>Transmit rate for transmission series 1.
See TxRate0 “Description” for transmission rate.</td></tr>
</table>

Table 4-2. **DMA Tx Descriptor, Words 2, 3, 4 and 5 (continued)**

Word	Bits	Field Name	Description
5	14–10	TXRate2	Transmit rate for transmission series 2. See TxRate0 "Description" for transmission rate.
	19–15	TXRate3	Transmit rate for transmission series 3. See TxRate0 "Description" for transmission rate.
	24–20	RTSCTSRate	Specifies the rate at which the RTS will be sent if RTSEn is set, or at which the CTS will be sent if CTSEn is set.
	31–25	RES	Reserved for future use. Must be set to zero.

The DMA Rx logic (the DRU block) manages Rx descriptors and transfers the incoming frame data and status to the host through the PCI/CardBus Interface.

The Rx descriptor format for words 2 and 3 is described in Table 4-3.

Table 4-3. **DMA Rx descriptor, Words 2 and 3**

Word	Bits	Field Name	Description
2	31–0	Reserved	Word 2 is not used. Must be set to zero.
3	11–0	BufLen	Data buffer length. Specifies the length, in bytes, of the data buffer associated with this descriptor. Receive data buffers must have a length that is an integral multiple of four bytes. The maximum size of the buffer is 4095 bytes.
	12	Reserved	Reserved for future use. Must be set to zero.
	13	InterReq	Interrupt request flag. Indicates whether the DMA engine should generate an interrupt after reception of the frame is completed. 1 = Generate an InterReq interrupt at the completion of this frame. 0 = Do not generate an InterReq interrupt at the completion of this frame.
	31–14	Reserved	Reserved for future use. Must be set to zero.

The RX descriptor format for words 4 and 5 is described in Table 4-4.

Table 4-4. **DMA Rx descriptor Completion Status Format Words 4 and 5**

Word	Bits	Field Name	Description
4	11–0	DataLen	Received data length. Specifies the length, in bytes, of the data actually received into the data buffer associated with this descriptor. Valid for all descriptors.
	12	More	More descriptors in this frame flag. 1 = The current frame is continued in the next descriptor. 0 = No more descriptors for the current frame. Valid for all descriptors.
	14–13	Reserved	Reserved for future use. Must be set to zero.
	19–15	RxRate	Reception rate indication. Indicates that rate at which this frame was transmitted from the source. Encoding matches those used for the “XmitRate” field in Table 4-2 on page 39. Valid only for the final descriptor of a frame, and only if the FrmRcvOK flag is set, or if the FrmRcvOK flag is clear and the PHYErr flag is clear.
	27–20	RcvSigStrength	Receive signal strength indication. The value of this field indicates the signal strength observed while the frame was being received. Valid only for the final descriptor of a frame, and only if the FrmRcvOK flag is set.
4	31–28	RcvAntenna	Receive antenna indication. This value indicates what antenna the packet was received on. Possible values are: 0000 = Auto 0001 = Omni 0 0010 = Omni 1/Sector 0 0011 = Sector 1 0100 = Sector 2 0101 = Sector 3 0110 = Sector 4 0111 = Sector 5 1000 = Sector 6 1001 = Sector 7 1010 = Sector 8 1011 = Sector 9 1100 = Sector 10 1101 = Sector 11 1110 = Sector 12 Valid only for the final descriptor of a frame, regardless of the state of the FrmRcvOK flag.

Table 4-4. **DMA Rx descriptor Completion Status Format Words 4 and 5**

Word	Bits	Field Name	Description
5	0	Done	Descriptor completion flag. 1 = The MAC has finished processing the descriptor and has updated the status information. 0 = The MAC has not finished processing the descriptor. Valid only for the final descriptor of a frame.
	1	FrRxOK	Frame reception success flag. 1 = Frame received successfully. 0 = An error occurred during frame reception. Valid only for the final descriptor of a frame.
	2	CRCErr	Cyclic redundancy code (CRC) error flag. 1 = Reception of frame failed because of an incorrect CRC value. 0 = Frame received without a CRC error. Valid only for the final descriptor of a frame, and only if the FrmRcvOK flag is clear.
	3	DecryptCRCErr	Decryption CRC failure flag. 1 = Decrypt CRC failed. 0 = No decrypt CRC error. Valid only for the final descriptor of a frame, and only if the FrmRcvOK flag is clear.
	4	PHYErr	PHY error flags. If not equal to zero, then reception of the frame failed because the PHY encountered an error. PHY errors include: 00000 =ERROR TRANSMIT_UNDERRUN 00100 =ERROR PANIC 00101 =ERROR RADAR_DETECT 00110 =ERROR ABORT 00111 =ERROR TX_INTERRRUPT_RX 10001 =ERROR OFDM TIMING 10010 =ERROR OFDM SIGNAL_PARITY 10011 =ERROR OFDM RATE_ILLEGAL 10100 =ERROR OFDM LENGTH_ILLEGAL 10101 =ERROR OFDM POWER_DROP 10110 =ERROR OFDM SERVICE 10111 =ERROR OFDM RESTART 11001 =ERROR CCK TIMING 11010 =ERROR CCK HEADER_CRC 11011 =ERROR CCK RATE_ILLEGAL 11110 =ERROR CCK SERVICE 11111 =ERROR CCK RESTART Valid only for the final descriptor of a frame, and only if the FrmRcvOK flag is clear.

Table 4-4. **DMA Rx descriptor Completion Status Format Words 4 and 5**

Word	Bits	Field Name	Description
5	5	MichaelErr	Michael integrity check error flag. If set, then the frame's "Michael" integrity check value did not verify correctly. Valid only when all of the following are true: ■ The descriptor is the final one of the frame ■ The FrRxOK bit is clear ■ The frame was encrypted using TKIP ■ The frame is not a fragment ■ The frame is not a QoS frame Michael error will only be set if DecryptCRCErr is not set.
	7–6	RES	Reserved for future use. Must be set to zero.
	8	KeyIdxValid *or* PHYErrCode[0] (lower)	If the FrRxOK bit is set, then this field contains the decryption key table index valid flag. If set, indicates that the PCU successfully located the frame's source address in its on-chip key table and that the KeyIdx field reflects the table index at which the destination address was found. If clear, indicates that the PCU failed to locate the destination address in the key table and that the contents of KeyIdx field are undefined. If the FrRxOK bit is clear and the PHYErr bit is set, then this field contains bit [0] of the PHY error code. In both cases, this field is valid only for the final descriptor of a frame.
	15–9	KeyIdx *or* PHYErrCode[4:1] (upper)	■ If the FrRxOK bit is set, then this field contains the decryption key table index valid flag. If set, indicates that the PCU successfully located the frame's source address in its on-chip key table and that the KeyIdx field reflects the table index at which the destination address was found. If clear, indicates that the PCU failed to locate the destination address in the key table and that the contents of KeyIdx field are undefined. ■ If the FrRxOK bit is clear and the PHYErr bit is set, then this field contains bits [4:1] of the PHY error code, the upper three bits are zero. In both cases, this field is valid only for the final descriptor of a frame.
	30–16	RxTimestamp	A snapshot of bits 14:0 of the TSF_L32 register. Valid only for the final descriptor of a frame.
	31	KeyCacheMiss	Key cache miss indication. If set, shared key zero was indicated in the Rx frame but the source address was not in the key cache. Shared key zero was used instead. Valid only for the final descriptor of a frame, and only if the FrmRcvOK flag is clear.

Table 4-5 and Table 4-6 describe the completion status of the Tx and Rx descriptors.

Table 4-5. **DMA Tx descriptor Completion Status Formats Words 6, and 7**

Word	Bits	Field Name	Description
6	0	FrTxOK	Frame transmission success flag. If set, the frame was transmitted successfully. If clear, an error occurred during frame transmission. Valid only for the final descriptor of a frame.
	1	ExcTries	Excessive tries flag. If set, transmission of the frame failed because the try limit was reached before the frame was transmitted successfully. Valid only for the final descriptor of a frame, and only if the FrTxOK flag is clear.
	2	FIFOUnderrun	Transmit FIFO underrun flag. If set, transmission of the frame failed because the DMA engine was not able to supply the PCU with data quickly enough and so at least one of the 802.11 transmit timing constraints was violated. Valid only for the final descriptor of a frame, and only if the FrTxOK flag is clear.
	3	Filtered	Frame transmission filter indication. If set, indicates that frame was not transmitted because the corresponding destination mask bit was set when the frame reached the PCU. Valid only for the final descriptor of a frame, and only if the FrTxOK flag is clear.
	7–4	RTSFailCnt	RTS failure count. Reports the number of times an RTS was sent but no CTS was received for the final transmission series (see the FinalTSIdx field). For frames that have the RTSEn bit clear, this count always will be zero. Note that this count is incremented only when the RTS/CTS exchange fails. In particular, this count is not incremented if the RTS/CTS exchange succeeds but the frame itself fails because no ACK was received. Valid only for the final descriptor of a frame, regardless of the state of the FrTxOK flag.
	11–8	DataFailCnt	Data failure count. Reports the number of times the actual frame (as opposed to the RTS) was sent but no ACK was received for the final transmission series (see the FinalTSIdx field). Valid only for the final descriptor of a frame, regardless of the state of the FrTxOK flag.

Table 4-5. **DMA Tx descriptor Completion Status Formats Words 6, and 7 (continued)**

Word	Bits	Field Name	Description
	15–12	VirtCollCnt	Virtual collision count. Reports the number of virtual collisions that occurred before transmission of the frame ended. The counter value saturates at 0xf. A virtual collision refers to the case, as described in the 802.11e QoS specification, in which two or more output queues are contending for a TXOp simultaneously. In such cases, all lower-priority output queues experience a "virtual collision" in which the frame is treated as if it had been sent on the air but failed to receive an ACK.
	31–16	SendTimestamp	A snapshot of the PCU's timestamp (TSF value), expressed in TUs (that is, bits [25:10] of the PCU's 64-bit TCF). Intended for use by the driver in implementing the frame lifetime requirements associated with the 'aMaxTransmitMSDULifetime' MAC attribute. Valid only for the final descriptor of a frame, regardless of the state of the FrTxOK flag.
7	0	Done	Descriptor completion flag. Set to one by the DMA engine when it has finished processing the descriptor and has updated the status information. Valid only for the final descriptor of a frame, regardless of the state of the FrTxOK flag. The driver is responsible for tracking what descriptors are associated with a frame. When the DMA engine sets the Done flag in the final descriptor of a frame, the driver must be able to determine what other descriptors belong to the same frame and thus also have been consumed.
	12–1	SeqNum	Transmit sequence number. Indicates the sequence number the PCU assigned to the frame. Valid only for the final descriptor of a frame, regardless of the state of the FrTxOK flag.
	20–13	AckSigStrength	Signal strength indication for the ACK. Indicates the signal strength for the ACK frame (or other response) that the PCU received after it transmitted the frame.Valid only for the final descriptor of a frame, and only if the FrTxOKflag is set.
	22–21	FinalTSIdx	Final transmission attempt series index. Specifies the number of the transmission series (see the TXDataTriesN and TXRateN fields in Table 3.5 and Table 3.6) that caused frame transmission to terminate. Valid only for the final descriptor of a frame.
	23	RES	Reserved.
	24	TXAnt	Transmit antenna indication. If the frame was sent successfully, this field indicates which antenna was used for the successful transmission attempt.
	31–25	RES	Reserved for future use. Must be set to zero.

4.4 *Descriptor Usage*

The Tx and Rx descriptors define the interface between the host and the MAC, and govern the transfer of frame data, control, and status between the MAC and host memory.

For both frame transmit and receive, the host is responsible for managing linked lists of Tx and Rx descriptors stored in host memory. In normal operation, the host first informs the MAC of the location of the head of the Tx and Rx descriptor linked lists using the TXDP and RXDP registers. See "QCU Tx Descriptor Pointer Registers (Q_TXDP)" on page 120, and "Receive Queue Descriptor Pointer (RXDP)" on page 94, respectively.

From that point on, the MAC traverses the Tx descriptor list, performs the necessary data transfer operations over the PCI/CardBus and sends the frame to the PCU.

Each time the PCU indicates that a frame has been received, the PCU notifies the DRU, which traverses the list of Rx descriptors and performs the necessary data transfers to store the received frames data into host memory.

"Tx Descriptor Processing" on page 50 and "Rx descriptor Processing" on page 51 describe the details of this process.

4.4.1 Tx Descriptor Processing

To cause the MAC to send a frame into the network, the host creates a linked list of Tx descriptors that describe the frame to be sent. This list can be just one element long if the frame data is not spread across multiple memory blocks. The host then passes this list to the MAC. This process proceeds as follows:

1. The host creates the linked list of Tx descriptors in host memory. Each descriptor must include all fields listed in Table 4-1 on page 38. If the linked list contains more than one descriptor, all descriptors except the last must have their More flag set; the final (or only) descriptor must have its More flag cleared (see Table 4-2 on page 39).
2. The host passes the head of the linked list to the MAC. This step can occur in two different ways.
   - If there is an existing linked list of Tx descriptors, the host must append the new descriptor list to the end of the existing one.
   - If there is no existing list, the host must inform the MAC of the new descriptor list head by writing a pointer to the head element of the descriptor list into the TXDP register (see "QCU Tx Descriptor Pointer Registers (Q_TXDP)" on page 120 or.)

   In both cases, the host also must write a one to the TxE bit in the CR register (see "Command Register" on page 81) to ensure the MAC recognizes that new Tx descriptors are available.

The MAC processes descriptors as follows:

3. Reads the next descriptor from host memory and stores it on-chip.
4. Uses the control information in words 2 through 5 of the first descriptor to generate a control command word, which is passed to the PCU to establish the parameters for the next frame transmission.
5. Uses the BufPtr (see Table 4-1 on page 38) and BufLen (see Table 4-2 on page 39) fields to fetch any data associated with the descriptor. This data is transferred to the PCU's Tx FIFO.
6. Inspects the More flag (see Table 4-2 on page 39) to determine if this frame includes additional descriptors.
   - If the More flag is set, the MAC fetches the next descriptor for the frame from the address specified in the LinkPtr field and begins processing it at Step 5. The host is required to ensure that all descriptors for a given frame are available at the time the MAC begins processing the first descriptor of the frame. (It is illegal for the host to create a situation in which a descriptor's More flag is set, but the value of its LinkPtr field is null or points to an invalid next descriptor.)
   - If the more bit is clear (see Table 4-2 on page 39), this is the final descriptor of the frame. The MAC updates the Tx completion status fields (see Table 4-5 on page 48) and performs Step 7.

7. When the last descriptor of the frame has been processed, the MAC inspects the final descriptor's LinkPtr field to determine how to proceed.
   - If the LinkPtr is non-null, the MAC assumes it points to the first descriptor of a new frame, which is then fetched and the process repeats from Step 3.
   - If the LinkPtr is null, the MAC pauses and waits for the transmit enable (TxE) bit in the CR register to be set. When the TxE bit is set and the TXDP is not written by the host, the MAC reloads the LinkPtr field of the current descriptor. It then follows the link field to the new descriptor on the list. If the TXDP is written by the host, the MAC starts again at Step 3 by reading the descriptor pointed to by the TXDP register.

4.4.2 Rx descriptor Processing

Rx descriptors are processed much like Tx descriptors, except that there is only a single Rx descriptor queue. Additionally, unlike Tx descriptors, the size of an incoming frame or the number of frames that is received over a period of time is unknown to the host. Therefore, the DRU is able to handle having Rx data available from the PCU without having a Rx descriptor available.

The host creates a linked list of Rx descriptors and passes the head of the list to the MAC. If there are presently no available Rx descriptors, the host must create a linked list of Rx descriptors and write a pointer to the head of the Rx descriptor list into the RXDP register. The host also must write a '1' into the receive enable (RxE) bit in the CR register. The data buffer associated with a Rx descriptor must be aligned on a double-word boundary and must have a size that is an integral multiple of the system cache line size (see "Cache Line Size" on page 84).

The MAC handles received frames as follows:

1. It reads the next Rx descriptor from host memory and stores it on-chip.
2. It writes the received frame data into the data buffer associated with the descriptor until either the frame completes or until the data buffer is full (as indicated by the BufLen field included in word 3 of the Rx descriptor).
3. Sets the done flag for the descriptor and updates the BufLen field.
   - If there is more data associated with the frame (that is, the descriptor's data buffer filled before the frame was received completely), the MAC sets the More flag (see Table 4-4 on page 45).
   - If the frame is complete, the MAC clears the More flag (see Table 4-4 on page 45) and updates the remainder of the Rx descriptor completion status bits in words 4 and 5 of the current descriptor.
4. Regardless of whether the current frame is complete or not, the MAC inspects the LinkPtr word of the current descriptor (see Table 4-1 on page 38).
   - If it is non-null, the MAC fetches the next Rx descriptor from the specified location and continues, either with the current frame or with a new frame.
   - If the LinkPtr field is null, the MAC pauses and waits for the RxE bit in the CR register to be set. When the RxE bit is set, and the host has not written to the RXDP register, the MAC reloads the LinkPtr field of the current descriptor. The MAC then follows this link field to the next new descriptor that has been added to the list. If the RXDP has been written to by the host, the MAC starts again at Step 1 by reading the descriptor pointed to by the RXDP register.

As Step 4 describes, it is not inherently an error for the MAC to run out of Rx descriptors in the middle of a frame (or between frames). When this situation occurs, frame data begins backing up in the PCU's receive FIFO until the host provides additional Rx descriptors, thus permitting the MAC to proceed. However, if the PCU's receive FIFO overflows before the host provides additional Rx descriptors, a receive FIFO overrun error occurs (see "Primary Interrupt Status Register (ISR_P)" on page 103). Sufficient Rx descriptors and buffers should be supplied to prevent this condition from occurring.

4.4.3 Descriptor Completion Status Reporting

Upon completion of a Tx or a Rx descriptor (which might or might not be the final descriptor of the frame), the DCU, using the QCU that sourced the frame for transmits, or using the DRU logic for receives, updates the descriptor status fields to reflect that it has been consumed, and to report status information. The host can determine when the descriptor has been consumed by inspecting the done flag. For both Tx and Rx descriptors, the host initializes the done flag to zero, and the descriptor has been consumed when the QCU/DRU sets the value of the Done flag to one. (See Table 4-5 on page 48, and Table 4-4 on page 45.)

When the host has detected that the descriptor has been consumed, it can read words 4 and 5 of the descriptor to retrieve additional status information for the data transfer associated with the descriptor. Table 4-5 on page 48, and Table 4-4 on page 45 list the format of the status information returned by the QCU and DRU upon completion of a Tx descriptor and a Rx descriptor, (respectively). The contents of the bits within words 4 and 5 that are not part of the completion status information assume undefined values. Note that certain completion status fields are valid for all descriptors whereas others are valid only for the final descriptor of a frame, which can be identified by checking the state of the More flag (see Table 4-4 on page 45).

4.4.4 Buffer Description

To facilitate crytographic operations in the MAC, it is required that the length of the data body, including the IV, is a multiple of 4. The MAC header must be padded when the header length is not divisible by 4. For example, when "From_DS" and "TO_DS" fields are clear and the frame type is non-QoS data, the header length is 24 bytes which is divisible by 4, thus no padding is required. However, when both "From_DS" and "To_DS" fields are set and the frame type remains as non-QoS data (no TCID header present), the header length becomes 30 bytes and indivisible by 4. In such cases, it is necessary to pad the header by 2 bytes to bring the length up to 32 bytes. Table 4-6 summarizes all possible header lengths and their required padding action.

Table 4-6. **Header Lengths**

To DS	From DS	QoS	Header Length	Padding
0	0	0	24	No
1	0	0	24	No
0	1	0	24	No
1	1	0	30	2 bytes. 32 bytes total.
0	0	1	26	2 bytes. 28 bytes total.
0	1	1	26	2 bytes. 28 bytes total.
1	0	1	26	2 bytes. 28 bytes total.
1	1	1	32	No

The content of the padded bytes is not important since they will be skipped and will not be transmitted. Therefore, padding does not affect the frame length. However, as the padded bytes are DMA'ed into the transmit buffer, the Buffer Length in the transmit descriptor that points to the header portion of the frame has to be 2 bytes longer. Note that the length reported in the descriptor includes the padded bytes.

Similarly, on receive, the MAC inserts padded bytes after the header in accordance with the cases listed in Table 4-6.

Padding is performed automatically and software should be aware of where the data body of the frame begins.

5. Queue Control Unit (QCU)

The queue control unit performs two tasks:

- Managing the Tx descriptor chain processing for frames pushed to the QCU from the host. This involves traversing the linked list of Tx descriptors and transferring frame data from the host to the targeted DCU.
- Managing the queue transmission policy. This determines when the frame at the head of the queue should be marked as available for transmission.

The MAC contains 10 QCUs. Each QCU contains all the logic and state (registers) needed to manage a single queue (linked list) of Tx descriptors. A QCU is associated with exactly one DCU. When a QCU prepares a new frame it signals ready to the DCU. When the DCU accepts the frame, the QCU responds by fetching the frame data and passing it to the DCU for eventual transmission to the PCU and on to the air.

The host controls how the QCU performs these tasks by writing to various QCU configuration registers, listed in Table 5-1.

Table 5-1. **QCU Registers**

Register	Size (bits)	Description
TxDP	32	Points to next Tx descriptor to be fetched.
TxE	1	Enables descriptor processing.
TxD	1	Disables descriptor processing after any pending frames complete.
SchedPolicy	3	Controls frame scheduling policy; see "Frame Scheduling Policy" on page 54.
CBRInterval	24	Determines the CBR interval when CBR scheduling is enabled, in µs.
CBROvfThresh	8	Determines the value for the CBR expired counter at which the QCU generates a QCBROVF interrupt (see "Host Interface Unit Interrupts" on page 34).
ReadyTimeEn	1	Enables the operation of the ReadyTime parameter.
ReadyTime	24	Determines the maximum continuous duration for which the queue signals that it has valid frames to send, in µs.
RTShutdown	1	Status bit that, if set, indicates the QCU shut down the queue because its ReadyTime expired even though the queue still had additional frames on it.
OneShotEn	1	If set, the queue operates in one-shot mode: when software sets OneShotEn and then sets OneShotArm, the queue triggers when the associated CBR or DBA event occurs, per the SchedPolicy setting. However, upon expiration of ReadyTime or upon reaching the end of the queue, the QCU clears OneShotArm and does not mark the queue as ready until the software again sets OneShotArm, even if the CBR interval expires or if DBA occurs. This mode can be used only with non-ASAP SchedPolicy settings.
OneShotArm	1	Set by software to arm the queue for one-shot operation.
FrPendCnt	2	Count of how many frames the QCU has pending in the PCU.

5.1 Descriptor Chain Processing

The TXDP register points to the head of the linked list of Tx descriptors and the QCU begins fetching descriptors when the host sets the TxE queue enable flag. Descriptors are fetched starting at the head of the queue and continuing until the QCU reaches the end of the descriptor chain, discussed more fully below. Writes to TxE without an intervening write to TXDP cause the QCU to perform a descriptor refresh.

To stop transmission for QCU N (N represents any 1 of 10 QCUs):

- Write a 1 to QCU N's TxD bit.
- Poll the Q_TXE register until QCU N's TxE bit is clear.
- Poll QCU N's Q_STS register until its pending frame count (bits [1:0] of Q_STS) is zero.
- Write a 0 to QCU N's TxD bit.

At this point, QCU N has shut down and has no frames pending in its associated DCU. Software must not write a '1' to a QCU's TxE bit when that QCU's TxD bit is set. Undefined operation will result. (Writing a '0' to TxE when TxD is set has no effect on the QCU.)

5.2 Frame Scheduling Policy

The MAC supports frame scheduling logic, which is controlled by the SchedPolicy and constant bit rate (CBR) Interval registers. The QCU frame scheduling policy determines when the QCU signals to its associated DCU that the frame at the head of the queue is “ready” (available for transmission). The frame scheduling policy setting applies to the QCU as a whole — it is per-QCU and not per-frame or per-time interval — and is assumed to remain static for the duration of the QCU’s use.

In general, the QCU provides three types of frame scheduling:

- Unthrottled — Frames are marked ready as soon as they reach the head of the queue
- Time-throttled — Frames are marked ready only upon the elapse of a certain time interval (that is, they are held at the head of the queue until the time interval elapses)
- Event-throttled — Frames are marked ready only upon the occurrence of a particular event, typically one that is detected outside the QCU. (Time-throttled and event-throttled are basically the same if the expiration of the time interval is viewed as an event.)

The specific QCU frame scheduling policies provided in the MAC are:

- ASAP — A frame is marked ready as soon as it reaches the head of the queue. Frame transmission continues until the end of the queue is reached (refer to “End of Queue/ End of List Detection” on page 55). This is an unthrottled mode.
- CBR — A frame is marked ready only upon expiration of the QCU’s CBR interval timer. Once this timer elapses, frame transmission continues until the end of the queue is reached (refer to “End of Queue/End of List Detection” on page 55). In addition, the CBR interval timer is immediately reset and begins counting down the next CBR interval. This is a time-throttled mode.
  - Each time the CBR interval elapses, the QCU increments a CBR expired counter. Whenever the CBR expired counter is non-zero and a frame is available at the head of the queue, the QCU marks the frame ready.
  - Upon encountering the end of queue condition, the QCU decrements the CBR expired counter. If this decrement of the CBR expired counter brings the counter value to zero, then the QCU does not attempt new frame transmission until the current CBR interval elapses, at which point the CBR expired counter increments to one and frame transmission resumes.
  - If, however, the decrement of the CBR expired counter leaves the counter value still non-zero, then the QCU resumes frame transmission attempts immediately. In this way, the QCU attempts to catch up to the host’s desired frames-per-CBR interval rate, even if network conditions temporarily cause the achieved frame transmission rate to fall below the desired value.

- DBA-gated — A frame is marked ready only upon the occurrence of the DMA beacon alert (DBA), as signalled from the PCU. Once the DBA occurs, frame transmission continues until the end of the queue is reached (refer to "End of Queue/End of List Detection" on page 55). This is an event-throttled mode.
  - The occurrence of DBA is tracked using the same "CBR expired" counter mechanism as was discussed above for the CBR scheduling policy. That is, this counter is incremented each time DBA occurs and decremented upon reaching an end-of-queue condition (refer to "End of Queue/End of List Detection" on page 55). The QCU marks frames ready whenever this counter is non-zero.
- TIM-gated (BSS mode) — The same as DBA-gated except that the trigger event for marking the queue valid is the receipt of a beacon with the local STA's bit set in the partial virtual bitmap within the TIM element. Note that a beacon arriving with the DTIM bit set (bit zero of the bitmap control field within the TIM element) but not the local STA's bit within the partial virtual bitmap does not qualify as a trigger event for this frame scheduling policy.
- TIM-gated (IBSS mode) — The same as DBA-gated except that the trigger event for marking the queue valid is the receipt of an ATIM frame directed to the local STA.
- Beacon-sent-gated — The same as DBA-gated except that the trigger event for marking the queue valid is the successful transmission of a beacon frame from the DCU designated for beacon transmission. Refer to "Beacon-gated Frames Handling" on page 60.

5.3 End of Queue/End of List Detection

A number of QCU functions depend on the detection of the end of the Tx descriptor chain, a so-called end of queue or end of list (EOL) condition. In the MAC, an end of queue occurs whenever the QCU:

- Fetches a descriptor whose LinkPtr field is NULL
- Fetches a descriptor whose virtual end-of-list (VEOL) bit is set. (See Table 4-2 on page 39 for the format of a Tx descriptor.)
- Exceeds the ReadyTime limit. The ReadyTime QCU parameter determines the maximum continuous period of time the queue indicates that it has frames ready for transmission.
  - When the ReadyTime function is enabled by setting the ReadyTimeEn bit, the QCU begins counting down the ReadyTime starting at the same event (that is, the expiration of the CBR interval timer or the occurrence of DBA) that causes the queue to be marked ready. Thereafter, normal frame processing occurs until the ReadyTime duration expires. At this point the QCU ceases marking frames ready even if it has not yet encountered one of the other two end-of-queue conditions.
  - ReadyTime may be used only with non-ASAP frame scheduling policies.

In most cases the three end-of-queue conditions are treated identically, with two exceptions:

- The QCU signals an end-of-list interrupt only if a descriptor's LinkPtr is NULL.
- The QCU by default does not clear the TxE bit on occurrence of VEOL or ReadyTime expire. The QCU clears TxE only when it encounters a NULL LinkPtr. A register bit within each QCU can be set to change this policy so that the QCU clears TxE for VEOL and ReadyTime expire.

6. DCF Control Unit (DCU)

Collectively, the 10 DCUs implement the EDCF channel access arbitration mechanism defined in the Task Group E (TGe) QoS extension to the 802.11 specification. Each DCU is associated with one of the eight EDCF priority levels and arbitrates with the other DCUs on behalf of all QCUs associated with it. A central DCU arbiter monitors the state of all DCUs and grants one the next access to the PCU (that is, access to the channel).

Because the EDCF standard defines eight priority levels, the first eight DCUs (DCUs 0–7) map directly to the eight EDCF priority levels. The two additional DCUs handle beacons and beacon-gated frames for a total of 10 DCUs.

The mapping of physical DCUs to absolute channel access priorities is fixed and cannot be altered by software:

- The highest-priority DCU is DCU 9. Typically, this DCU is the one associated with beacons.
- The next highest priority DCU is DCU 8. Typically, this DCU is the one associated with beacon-gated frames. (Refer to "Beacon-gated Frames Handling" on page 60)
- The remaining eight DCUs priority levels are filled with DCUs 7 through 0. Among these 8 DCUs, DCU 7 has highest priority, DCU 6 the next highest priority, and so on through DCU 0, which has the lowest priority. Typically, these DCUs are associated with EDCF priorities seven through zero, respectively.

6.1 DCU State Information

Each DCU maintains sufficient state information to implement EDCF channel arbitration. Table 6-1 lists the basic DCU state registers.

Table 6-1. **DCU Registers**

Register	Size (bits)	Description
QCUMask	10	Indicates which QCUs are associated with this DCU. Used to mask the QCU ready bits. Bit [i] maps to QCU [i].
CWMin	8	The EDCF CWMin parameter, in slots.
AIFS	8	The EDCF AIFS (arbitration inter-frame spacing) interval, in slots beyond SIFS.
PF	1	The EDCF CW persistence factor (0=CWnew equals CWOld; 1=binary exponential backoff).
FRFL	6	Frame RTS failure limit.
SRFL	6	Station RTS failure limit.
SDFL	6	Station data failure limit.
SeqNum	12	Next frame sequence number.
ChannelTimeEn	1	Enable use of ChannelTime setting.
ChannelTime	20	Maximum burst time, in µsec.
DestMask	128	Filter for each of the 128 destinations, as indexed by the KeyTableIdx field of the Tx descriptor.

6.2 DCU Channel Arbitration Procedure

The DCU begins channel arbitration by determining whether any of the associated QCUs has a frame ready for transmission. The DCU makes this determination by logically ANDing the 10 QCU ready bits with the QCUMask register to arrive at a set of QCUs that are both associated with the DCU and have a frame available.

If at least one QCU is ready, the DCU next performs the EDCF channel access procedure, meaning it waits until the channel has been idle for at least an AIFS (if the channel has not already been idle for this long) and then attempts transmission or, if the channel is found to be busy or becomes busy, it generates a backoff count and CW value and begins counting down the backoff slots. (See the EDCF specification for a more detailed description of the channel access procedure and state machine.)

At some point, the DCU determines that frame transmission is “imminent.” The definition of “imminent” would, in theory, be when the DCU’s backoff count reaches zero, but in practice needs to be somewhat more conservative to allow time to fetch the frame data and forward it to the PCU before the PCU actually needs to put the frame on the air. Thus the DCU might, for instance, determine that frame transmission is imminent when a frame is available and the backoff count is less than or equal to four (the threshold for the “imminent” determination is software-programmable).

Regardless of the actual threshold value, once the DCU determines that frame transmission is imminent, it asserts a ready signal to the central DCU arbiter (see Figure 4-1, “MAC Block Diagram,” on page 37). The DCU arbiter selects the highest-priority DCU.

The selected DCU now proceeds to select the QCU to be the source of the frame. To do so, the DCU again logically ANDs the QCU-ready bits with the QCUMask value and passes the result into a round-robin priority encoder. The encoder’s output is the QCU that is the source of the next frame. Note that the selected QCU might not be the one that caused the DCU to begin arbitrating for the channel. Once the DCU has selected the QCU, it signals the selected QCU to begin transferring the frame data from host memory.

The DCU places the frame data into its prefetch buffer and, simultaneously, drives the data from its prefetch buffer to the PCU. In addition to the frame data itself, the DCU also conveys the following to the PCU:

- The control information from words 2 and 3 of the first Tx descriptor
- A tag that identifies the DCU and QCU from which the frame originated

The DCU now waits (if needed) until the EDCF channel access requirements have been met (backoff count is zero, channel has been idle for at least an AIFS, etc.) and then indicates to the PCU to begin frame transmission on the air.

The PCU transmits the frame (more details in “Protocol Control Unit (PCU)” on page 61) and reports the result to the DCU that sourced the frame. If the frame was sent successfully, the DCU repeats the above process and selects a new frame for transmission, potentially from a different QCU. If, however, the PCU reports that frame transmission failed, then the DCU follows the backoff procedure defined in the EDCF specification and rearbitrates for the PCU on behalf of the same frame until either the PCU reports successful transmission or until the frame’s retry limit is reached, as controlled by the SRL/LRL DCU parameters.

Once a frame is completed, either by successful transmission or by reaching its retry limit, the DCU accepts the status information from the PCU and issues the necessary completion write to update the descriptor status words in host memory.

6.3 Handling of the ChannelTime Parameter (Frame Bursting)

If the ChannelTimeEn bit is set, then the DCU performs a frame burst each time it gains access to the channel. To manage this process, the DCU initializes a timer to the value of the ChannelTime register setting and starts the timer when the DCU arbiter first grants the DCU access to the PCU. The DCU also indicates to the DCU arbiter that it is starting a frame burst. The DCU arbiter responds by continuing to grant the DCU access to the channel, even if higher priority DCUs become ready, until the bursting DCU indicates that its burst is complete. The DCU ends the frame burst either when the ChannelTime duration elapses or when there are no ready QCUs. Note that during a burst the DCU continues to process ready QCUs in round-robin order, and that the DCU terminates ChannelTime bursts only at intra-frame boundaries.

6.4 Tx frame Filtering

Filtering of frames to be sent is designed solely to deal efficiently with the overall IEEE 802.11 restriction that all frames must be delivered to their destination in order. Implementing this requirement conflicts with the desire of the host—especially one controlling an access point—to queue multiple frames for transmit to the same destination. Enabling the host to queue multiple frames, to the same destination or to multiple destinations, is desirable to achieve high throughput on the network, and low utilization of the host.

If the DCU blindly sent all frames that the host had queued for transmission, errors could occur. For example, if the host queued two frames to the same destination and the transmission of the first frame failed, the DCU might end up sending the second frame before the host could be notified that transmission of the first frame failed. If this occurred, and if transmission of the second frame were successful, the destination node would receive the two frames out of order, thereby violating the in-order delivery requirement.

To address this problem, yet still allow the host to queue multiple frames to the same destination, the DCU provides a mechanism for dynamically blocking transmits to a particular set of destination nodes, while continuing to allow transmitting to other nodes to proceed until the host indicates that it is safe to resume sending to the blocked nodes. To do so, the DCU contains an array of 64 destination mask bits. These bits are initialized to zero. (Refer to "Transmit Filter Command Register (D_TXBLK_CMD)" on page 135.)

Each time the DCU wishes to send a frame, it first inspects the EncryptKeyValid field from the Tx descriptor. If this flag is clear, the DCU allows the frame send to proceed. If the EncryptKeyValid bit is set, the DCU uses the EncryptKeyIdx field from the Tx descriptor to select one of the 64 destination mask bits from its on-chip array. If the selected destination mask bit is zero, the DCU allows the frame to proceed. However, if the destination mask bit is a one, the DCU inspects the "ClearDestMask" flag in the Tx descriptor. If the ClearDestMask flag is zero, the DCU discards the frame and leaves the destination mask bit set to one. If the ClearDestMask bit is one, the DCU allows the frame to be sent and updates the destination mask bit to a zero. Each time the DCU fails to transmit a frame, it sets the corresponding destination mask bit to a one.

For access point and ad hoc applications, the DCU provides a mechanism in which a failure to transmit a frame causes the DCU to notify the host, and then halt all further transmission until the host re-enables the DCU. This mechanism allows the host to queue many frames to the same destination (and to different destinations) while adhering to the in-order delivery requirement. Basically, each time a frame transmission fails, the DCU sets the corresponding destination mask bit to a one. This prevents future frames to the same destination address from being sent until the host has had time to re-issue the failed frame. Only on this retry does the host set the ClearDestMask bit to a one. For all other transmits (non retries, or re-issues of frames that were blocked by the destination mask), the host sets the ClearDestMask bit to a zero.

6.5 Beacon-gated Frames Handling

The “beacon-gated frames” are required to be sent immediately following the beacon, as specified by IEEE 802.11a.

There are two situations:

- In a BSS, the AP is required to send a beacon as close to TBTT as possible, and then to follow it, if signalled in the beacon, with buffered multicast/broadcast frames.
- In an IBSS, each STA must compete to send the beacon, and then, upon sending or successfully receiving a beacon, must send ATIMs announcing all buffered data frames. ATIM transmission must cease at the end of the ATIM window, at which point data frames for which an ATIM was transmitted successfully may be sent until the next TBTT, when the process repeats.

Both of these situations are handled by dedicating a separate DCU (DCU 9)to beacons and another separate DCU (DCU 8) to beacon-gated frames. Each of these dedicated DCUs associates with only a single QCU. The QCU associated with each DCU must be configured to have a DMA beacon alert triggered frame scheduling policy. Refer to section “Frame Scheduling Policy” on page 54.

7. Protocol Control Unit (PCU)

7.1 PCU Functional Description

The PCU is responsible for the details of sending a frame to the baseband logic for transmission, for receiving frames from the baseband logic and passing the frame data to the DRU. This includes:

- Buffering Tx and Rx frames
- Encrypting and decrypting
- Generating ACK, RTS, and CTS frames
- Maintaining the timing synchronization function (TSF)
- Inserting and verifying FCS
- Generating virtual clear channel assessment (CCA)
- Updating and parsing beacons

The PCU is primarily responsible for buffering outgoing and incoming frames and conducting medium access compatible with the IEEE 802.11 DCF protocol. This includes maintaining valid inter-frame spacings of short inter-frame space (SIFS), acknowledge (ACK), virtual carrier sense, and support of software governed RTS/CTS frames. In addition, the PCU is the keeper of the timing synchronization function (TSF) and various timer functions, including beacon period, ATIM window, and beacon update.

To lessen the load on the PCI bus and the host, the AR5212 can filter received frames based on a number of criteria, and parse the TIM field of a beacon frame to determine the generation of PS poll. Encryption and decryption are handled in the PCU with an on-chip cipher engine and a key cache that can hold up to 128 shared and unique keys. During receive, the key cache behaves like a content addressable memory (CAM), searching for the key that is associated with the transmitter address (TA).

The PCU ensures that the STA can operate in point coordination function (PCF) basic service set (BSS) by being aware of the contention free period (CFP) and obeying the special rules for setting the network allocation vector (NAV) during a CFP. However, the PCU does not support the IEEE 802.11 optional PCF poll

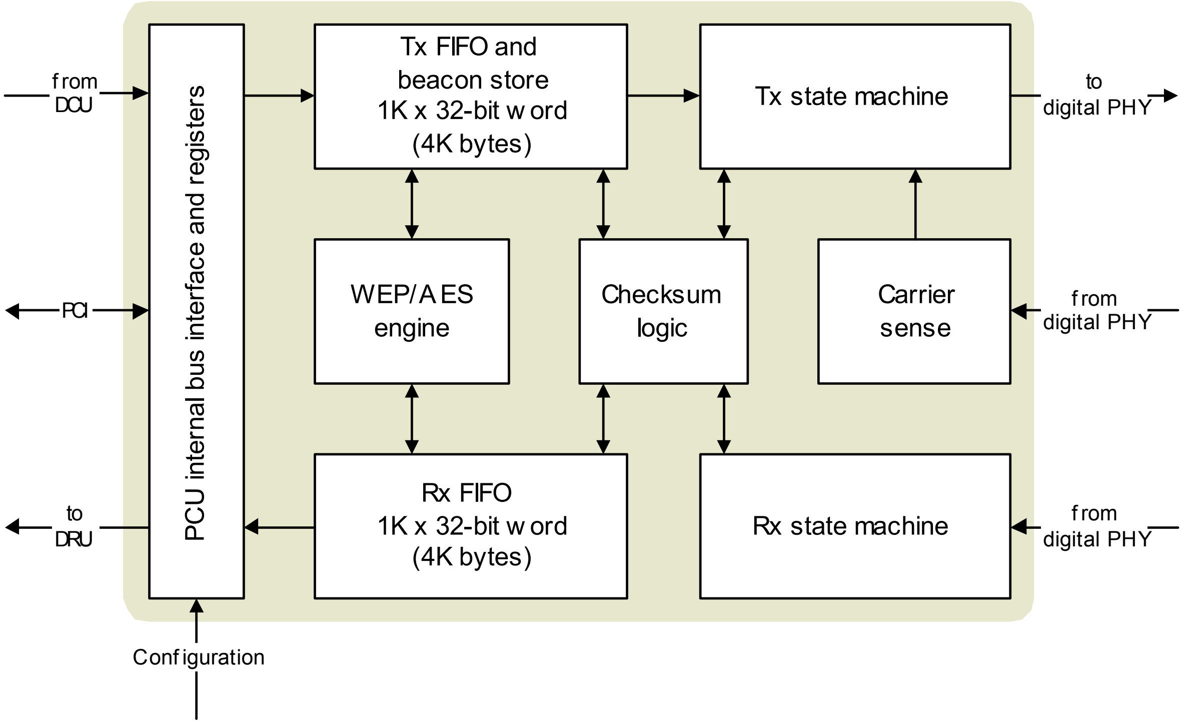


Figure 7-1. **PCU Functional Block Diagram**

7.2 Time Bases

The PCU maintains a number of time bases to support various real-time related functions. For example: TSF, timers, inter-frame spacings, NAV, and ACK/CTS time-out. [There is a 1 µs time base from which most other timings are derived. This time base is generated with a counter clocked at the core clock frequency of 40 MHz (in standard 802.11a mode) or 80 MHz (in turbo mode).] There is a separate 1 µs timebase counter in the 32 MHz clock domain to support TSF and all timer functions which would remain operational even in the doze state. These counters can be programmed through setting the fields MAC_USEC and MAC_USEC_32 fields of register MAC_USEC. The default values are 39 (for a 40 MHz core clock) and 31 (for a 32 MHz non-stop clock).

The TSF is a 64-bit micro-second timer kept by the local STA. (Refer to "Time Synchronization Function L32 (TSF_L32)" on page 163 and "Time Synchronization Function U32 (TSF_U32)" on page 163.) The TSFs of all STAs in the network are kept synchronized (+/- 2 µs) by copying the timestamp carried within the beacon. An AP's TSF is the absolute reference for all other STAs and is not modified when started. For non-AP STAs in a BSS, the TSF would be unconditionally updated with the timestamp of an error-free beacon while ad hoc STAs update the TSF when the received beacon timestamp is later than its TSF.

To make TSF synchronization as accurate as possible, Tx and Rx delays experienced by a beacon have to be compensated. Before inserting into the timestamp field of a transmitting beacon, the TSF is adjusted by the MAC_USEC_TX_LAT field of register MAC_USEC. The default value is 58 (for a 6Mb/s data rate). In the receive direction, the received beacon timestamp is adjusted by the MAC_USEC_RX_LAT field of register MAC_USEC before it is used to update the TSF. The default value is 29 (for a 6 Mbps data rate). The current value of TSF can be accessed by reading the registers MAC_TSF_L32 and MAC_TSF_U32.

Another PCU responsibility is to report CCA information to the DCUs so that the DCUs can properly implement the EDCF channel access state machine. The PCU continuously reports to the DCU when the channel is busy, taking into account both when the channel is physically busy and when the channel is virtually busy, as indicated by the NAV or other 802.11 protocol state. In other words, the PCU implements the 802.11 physical/virtual carrier sense mechanism and reports the results to the DCUs.

The NAV (refer to "NAV Value (NAV)" on page 165), also known as virtual carrier sense, is a method to reserve a length of time for a STA to perform its subsequent transmissions with a minimal chance of being interrupted. Within the MAC header of every frame type except power save (PS) poll, there is a 16-bit field known as duration. When a STA receives an error-free frame not directed to it and its current NAV value is less than the received duration, it sets its NAV to this duration value. The STA decrements its NAV by 1 every µs. Whenever its NAV is non-zero, a STA considers the medium to be busy, regardless of what CCA indicates, and refrains from transmitting. The current value of NAV can be read from the read-only register NAV.

After the transmission of a unicast non-control frame or a RTS frame, the receiver is expected to respond with ACK or CTS frames, respectively. If the expected response is not forthcoming within a time-out period, a reception error is assumed.

The ACK and CTS time-out periods are based on the core clock, and their lengths are controlled by programming the TIME_OUT register (refer to "Time Out (TIME_OUT)" on page 156).

7.3 RTS/CTS

RTS/CTS exchange is supported by the PCU. On Tx, the host can specify a frame or fragment burst is preceded by a RTS/CTS exchange by setting the RTS enable bit in the Tx descriptor. The PCU generates an RTS frame using the duration field copied from the RTS duration field in the descriptor, and waits for the receiver to respond with a CTS frame. When an error-free CTS frame is received, the data frames are transmitted after a SIFS interval. If the expected CTS is not received within CTS-Timeout of the RTS (refer to "Time Out (TIME_OUT)" on page 156), a retransmission is scheduled after performing an appropriate back-off. If the RTS/CTS exchange was successful, but the data frame failed, a fresh RTS/CTS exchange takes place before the frame is retransmitted.

7.4 Beacon

A beacon is a special management frame transmitted at a regular interval by the access point (AP) in a BSS or by all STAs in an IBSS. The beacon advertises the existence of the network and carries all the essential parameters of the network.

In a BSS, the AP schedules a beacon for transmission when target beacon transmission time (TBTT) is reached. TBTTs are separated by a beacon interval, which is a network-wide parameter. The moment of actual transmission must be dictated by DCF rules.

The content of the beacon in a BSS changes every interval, because it contains some fields that count down, and fields that indicate buffered traffic for STAs operating in power saving mode. The host is expected to update the beacon ahead of every TBTT and place it into a queue. A timer (SW beacon alert) generates an interrupt at a programmable time interval before TBTT to prompt the host to prepare the beacon. Another timer (DMA beacon alert) signals the QCU at a later time, but before TBTT, to begin transmitting frames. At TBTT, whose arrival is determined by a third timer, the beacon is scheduled for transmission. (If a valid beacon has not been loaded at TBTT, the PCU does not transmit, but waits until a valid beacon signal is detected.)

Unlike its counterpart in a BSS, the content of beacons in an IBSS remains unchanged (except for the timestamp field which is filled in by the PCU hardware), when the network is established. Because no AP is present in this configuration, all STAs in the network share the responsibility to transmit the beacon. At TBTT, all STAs contend to transmit the beacon using a modified back-off rule. Whenever a beacon is transmitted, all STAs cancel their scheduled beacon transmissions. The STA that transmitted the beacon must stay awake (no power save condition) throughout the beacon interval to reply to probe requests. At the moment of beacon transmission, the PCU inserts the current local TSF into the timestamp field in the beacon frame body.

7.5 Timers

The PCU supports the timers shown in Table 7-1. The timers are used differently according to the role of the STA. When the beacon interval register is written to, these timers are enabled simultaneously. All four timers operate in the 32 MHz clock domain.

Refer to "Next Beacon Time (NEXT_BEACON)" on page 158, "DMA Beacon Alert Time (DBA)" on page 158, "Software Beacon Alert (SBA)" on page 159, and "ATIM Window (ATIM_WIN)" on page 159.

Table 7-1. **Beacon Timers**

Timer	Time Units	Size (bits)	STA Role	Description
MAC_NEXT_BEACON	1024 µs	16 bits	AP and IBSS	Used for generating the TBTT events. The event time is incremented by a beacon interval at TBTT to the next TBTT. For an AP or the STA that initiates an IBSS, the host initializes timer to zero at the moment the network becomes operational. For a STA that joins an IBSS, the host listens for beacons, and uses the knowledge of the beacon interval and the beacon timestamp to compute the next TBTT and initialize the timer with it. Beacon interval is set in field BEACON_PERIOD of register BEACON.
MAC_SW_BEACON_ALERT	128 µs	19 bits	AP and IBSS	Used for generating the SW beacon alert events. The event time is incremented by a beacon interval to the next event time at the current event.
MAC_DMA_BEACON_ALERT	128 µs	25 bits	AP and IBSS	Used for generating the DMA beacon alert events. The event time is incremented by a beacon interval to the next event time at the current event. Immediately after a DMA beacon alert event occurs, the PCU hardware increments the value of this timer (modulo 2^{25}) by the value of the beacon interval, thus indicating the time for the next DMA beacon alert. **NOTE:** The next DMA beacon alert is not delayed even if the next beacon time is delayed (due to contention when the AP wants to access the medium to send the beacon frame).
			STA in BSS	Used for generating the start of contention-free period (CFP) events. The event time is incremented by a contention-free period to the next event time at the current event.
MAC_ATIM_WINDOW	1024 µs	16 bits	IBSS	Used for generating the end of ATIM window events. The event time is incremented by a beacon interval to the next event time at the current event.

7.6 Encryption and Decryption

The cipher unit is shared between encryption of outgoing frames and decryption of incoming ones. It supports all mandatory and optional cipher modes specified in 802.11i, including WEP, TKIP and AES. In addition, the unit contains a 128-entry key cache which holds both shared and unique keys. Each entry in the cache holds a key up to 128 bits long, the address of the STA with which the key is associated, and the type of cipher to be performed. In addition to the 802.11 cipher modes, there is a key type where no HW encryption or decryption is performed.

Keycode entries 0 to 3 are reserved for the standard shared keys.

In the transmit direction, the PCU examines the WEP bit in the MAC header. If set, the appropriate key is read out from the key cache entry using the EncryptKeyIdx field in the Tx descriptor as the index. The WEP unit encrypts the frame on the fly as the frame is transmitted. In the receive direction, if the WEP bit is set in the MAC header the KeyID field in the IV portion of the WEP frame body is used to find the key. If KeyID is nonzero, the shared key is looked up from the key cache using KeyID as the index. Otherwise, the key cache is searched sequentially for the transmitter address contained in the Rx frame. If the address is found and the entry is valid, the key associated with the entry is used to decrypt the frame body. If no match is found, the frame is decrypted using the key at index 0.

NOTE: The search is conducted even if the WEP bit in the MAC header is zero. The results are communicated back to SW using the KeyIdx and KeyIdxValid fields of the Rx descriptor. However, no decryption is performed.

CRPT_MIC_ENABLE mode bit enables the verification and replacement of the Michael in TKIP. If this bit is not set then the expectation is that software will be performing the check on receive and inserting the Michael field on transmit.

The Michael key is stored in the upper 64 location of the key cache. If the TKIP key entry for a particular destination is at i, then the corresponding Michael entry will be at i + 64. TKIP entries should not be entered into the upper 64 entries if Michael is enabled.

The Michael key is stored in the bit location normally used to store key [31:0] and key [79:48] for Michael key [31:0] and Michael key [63:32], respectively.

The valid bit is set to 0 for the entry when the entry is used for Michael.

Hardware does not perform Michael verification or replacement of Michael for fragmented frames and QOS data frames. Ignore Michael check failure for receives in either of these conditions.

7.7 TIM and PS Poll

In a BSS, the AP buffers any frames that are destined for a power-save (PS) capable STA that is in the doze state. The STA wakes every listen interval to listen for a beacon from the AP. Carried in the beacon frame body is a TIM element with which the AP indicates the presence of traffic for all the associated PS STAs. In the event where TIM indicates the presence of buffered frames, the STA is expected to respond by sending a PS poll frame to the AP, prompting it to transmit one frame from its buffer.

To facilitate the parsing of the TIM element by the PCU, the byte offset from the start of the MAC header to the bitmap control sub-field in the TIM is programmed by the host by writing to the TIM_OFFSET bit (refer to "Beacon (BEACON)" on page 157).

7.8 Co-existence with PCF

At the beginning of a contention free period (CFP), (refer to "CFP Interval (CFP_PERIOD)" on page 158), NAV is set to CFP maxduration, a constant which can be found in the contention free (CF) parameter element of a beacon. CFP_PERIOD is programmed by the host to synchronize with the CFP repetition period to signal the TBTT of the start of a CFP. Whenever a CF-End/CF-End-Ack frame is detected from any BSS, NAV is reset to zero. Intermediate updates of NAV are also performed with CFPDurRemaining, when present in any valid beacon that is received.

7.9 Ad Hoc

Before TBTT arrives, a PS STA is wakened by the DMA beacon alert (refer to "Software Beacon Alert (SBA)" on page 159) in preparation for sending and listening for beacons and ATIMs. At TBTT, a beacon transmission is scheduled using a contention window (CW) of 2xCWmin, where CWmin is 15 (in unit of slot) as specified by IEEE 802.11a. Note that each slot is 9 µs. During the back-off count-down, if a beacon is received, the scheduled beacon is cancelled. Note that a fragment burst is not interrupted in order to send a beacon.

During the ATIM window, which lasts for an ATIM window interval from TBTT, no transmission of frames other than beacon, ATIM, ACK, RTS, and CTS are initiated, but a frame started before TBTT is allowed to complete. ATIMs (if any) are loaded into the TxFIFO after a DMA beacon alert is generated by Timer1. After a beacon is either sent or received, ATIM frames are read out from the TxFIFO and sent. When the first non-ATIM frame emerges from the FIFO, transmission is suspended until the ATIM window has ended. Any ATIMs left in the FIFO after the end of the ATIM window are discarded.

As part of the preparation for the up-coming beacon period, the host sets bits of the "Set Transmit Filter Register (D_TXBLK_SET)" on page 138 and "Default Antenna (DEF_ANTENNA)" on page 164 registers that correspond to all PS STAs in the network. Only an acknowledged transmission of ATIM directed at that destination can clear the block, allowing subsequent frames for that destination to be sent. For broadcast or multicast traffic, a key index and hence a Tx block bit, must be allocated to each session. Clearing the bit in this case only requires the transmission of the corresponding ATIM.

When a STA has received a relevant ATIM or transmitted a beacon, it remains awake until the end of the next ATIM window. Otherwise, it enters the power save mode. If there are frames to be transmitted, the host must wake the STA manually.

7.10 Sleep

To reduce power when operating as a station, the AR5212 can be programmed to disable all logic except the TSF timer and the PCI core. The hardware will power up at programmed intervals to receive beacons. If the beacon indicates directed receive traffic (through the TIM element in the beacon frame), the hardware will remain powered up by switching from sleep to wake mode to wait for the receive frame. The hardware will also power up to transmit when directed by the host.

Sleep mode is selected by the SLE register. In force wake mode, the sleep function is disabled. Whenever an interrupt occurs, SLE switches to force wake. In force sleep mode, the hardware remains in a power down mode. This is the default value following cold reset. In normal sleep mode, the hardware enters sleep when idle.

During normal sleep, the hardware will periodically wake to receive beacons, multicast, and broadcast frames, under the control of SLEEP1 (refer to "SLEEP 1 (SLP1)" on page 167), SLEEP2 (refer to "SLEEP 2 (SLP2)" on page 168), and SLEEP3 (refer to "SLEEP 3 (SLP3)" on page 168) registers. The SLP_DTIM_PERIOD and SLP_TIM_PERIOD in the SLEEP2 register each set the period to sleep between beacons. The NEXT_DTIM field in the SLEEP1 and the NEXT_TIM field in the SLEEP2 register each set the TSF count at which the hardware should wake looking for a beacon. Whenever TSF equals either (or both) NEXT_DTIM or NEXT_TIM, the hardware wakes and that NEXT_TIM bit in SLEEP 2 is incremented by its SLP_TIM_PERIOD bit in the SLEEP 3 register. Once the hardware wakes, the next beacon is received and processed.

If the hardware wakes because of NEXT_TIM and the bit in the TIM element indicated by the AID field of the BSS_ID1 register bit is set, an interrupt is generated. Otherwise, the hardware returns to sleep.

If the hardware wakes because of NEXT_DTIM and the DTIM bit in the TIM element of the DTIM beacon is set, then the hardware will stay awake until a multicast frame is received with a clear More bit in the MAC header or if CAB_TIMEOUT in SLEEP1 expires. If the DTIM bit is not set in the DTIM beacon, an interrupt is generated.

The BEACON_TIMEOUT field in the SLEEP2 register controls how long to wait for a beacon after waking. After waking because of NEXT_TIM, if no beacon is received by BEACON_TIMEOUT, the hardware returns to sleep. If the hardware wakes because of NEXT_DTIM and no beacon is received by BEACON_TIMEOUT, the ASSUME_DTIM bit in the SLEEP1 register controls the action. If ASSUME_DTIM is clear, the hardware will immediately return to sleep.

In general, SLP_DTIM_PERIOD and SLP_TIM_PERIOD are initialized to a multiple of the beacon period. NEXT_DTIM and NEXT_TIM are initialized to wake the hardware shortly before the next TBTT (or next DTIM TBTT in the case of NEXT_DTIM), making sure to allow time for the hardware to exit sleep. Once initialized, the registers should automatically increment, tracking TSF.

7.11 Diagnostic and Management Information Base Features

The following features may be used for debugging:

- Disable ACK response
- Disable CTS response
- Disable encryption
- Disable decryption
- Disable transmit
- Disable receive
- Loop back
- Corrupt FCS
- Dump channel info
- Fix scrambler seed
- Freeze sequence number

Refer to "PCU Diagnostic (DIAG_SW)" on page 162.

The following statistical information is accumulated in the PCU and is accessible using the PCU registers:

- RTS success count (refer to "RTS OK (RTS_OK)" on page 165)
- RTS failure count (refer to "RTS Fail Count (RTS_FAIL)" on page 166)
- ACK failure count (refer to "ACK Fail Count (ACK_FAIL)" on page 166)
- FCS check failure count (refer to "FCS Fail Count (FCS_FAIL)" on page 166)
- Received beacon count (refer to "Beacon Count (BEACONCNT)" on page 167)
- Current timestamp (refer to "Time Synchronization Function L32 (TSF_L32)" on page 163 and "Time Synchronization Function U32 (TSF_U32)" on page 163)
- Timestamp in the last received beacon (refer to "Last Timestamp (LAST_TSTP)" on page 165)

7.12 Rx frame Filtering

Many of the received frames are not destined for this STA and may be discarded. The Rx filter provides hardware support to discard the unnecessary frames.

The PCU supports eight different Rx filters as shown in Table 7-2 on page 68. They are enabled separately by setting the appropriate bits of the RX_FILTER register. Both Probe Requests and directed (unicast) PS Poll frames are always passed to the host regardless of the Rx filter settings.

Table 7-2. **Rx Filter Types**

Filter Type	Description
Unicast	When enabled, data and management frames, with an Address 1 matching the STA address, are passed to the host. When disabled, all received unicast frames are ignored and therefore not acknowledged.
Multicast	When enabled, data and management frames that have a multicast address (bit 47 set) and that pass a hashing test are passed to the host. Hashing test: The 48-bit address is divided into eight 6-bit segments. All eight segments are bit-wise XORed to arrive at a 6-bit number. This is used to look up the corresponding bit in the 64-bit multicast filter vector, which is programmable by the host. If the bit is set, the test is considered passed. When disabled, all received frames are considered directed and if Address 1 matches the STA address, the frames are passed to the host.
Broadcast	When enabled, data and management frames (except beacons) with a receiver address of 0xFFFFFFFFFFFF and a basic service set identification (BSSID) matching that held in registers BSSID_0 and BSSID_1 are passed to the host. When disabled, frames with all F's in Address 1 are not passed to the host.
Beacon	When enabled, all beacon frames, regardless of BSSID, are passed to the host. This is used when a STA is scanning to join the network. When disabled, no beacon frames are passed to the host.
Control	When enabled, all control frames detected in the medium are passed to the host. When disabled, control frames are not passed to the host.
Promiscuous	When enabled, all frames detected in the medium are passed to the host. This is the only mode where it is possible to pass incomplete frames. When disabled, received frames are passed to the host based on the setting of the other filters.
Probe Request	When enabled, probe request frames are passed to the host.

The Rx frame filtering logic supports filtering based on destination address and on frame type. (Refer to "Receive Filter (RX_FILTER)" on page 160.)

Filtering options based on destination address include:

- Whether to accept any unicast frames:
  - If the RX_FILTER[UNICAST] bit is clear, the PCU does not pass any received unicast frames to the host. If the RX_FILTER[UNICAST] bit is set, the PCU passes to the host only those received unicast frames that have a destination address equal to the 48-bit value specified in the STA_ID0 and STA_ID1 registers.
  - Clearing RX_FILTER (Unicast) also disables the PCU from responding to RTS and data receive frames.
  - Setting RX_FILTER (Unicast) also enables automatic CTS and ACK transmit frames in response to RTS and data receive frames.
- Whether to accept any multicast frames:
  - When multicast frames are accepted (the RX_FILTER [MULTICAST] bit is set), the PCU looks for a BSSID match by providing a hashing function that operates on the 48-bit destination address of the incoming multicast frame. Using the hashing function, the destination address is collapsed to a 6-bit index. This index is used to select a single bit from a 64-bit register (refer to "Multicast Filter 1 (MCAST_FIL1)" on page 161, and "Multicast Filter 0 (MCAST_FIL0)" on page 161). If the selected bit is a one, the frame is accepted and passed to the host. If the bit is a zero, the frame is ignored. The index is created from a bitwise XOR of the eight groups of six bits that forms the 48-bit destination address (DA) in the MAC header, as shown here:

    ```
    index = DA[47:42] ⊕ DA[41:36] ⊕ DA
    [35:30] ⊕ DA[29:24] ⊕ DA[23:18] ⊕
    DA[17:12] ⊕ DA[11:6] ⊕ DA[5:0]
    ```
- Whether to accept all nonbeacon broadcast frames (BSSID does not match).
- Whether to operate in promiscuous mode.

  When enabled, the PCU accepts all frames, regardless of destination address or error content, and passes them to the host. A STA in promiscuous mode does not send ACKs except in a situation when it receives a frame that is really intended for it (the UNICAST bit is set).

Filtering options based on frame type include:

- Whether to accept any type of control frame.

  If a frame fails to pass one or more of the filtering tests, the PCU does not pass the frame to the host; however, this does not mean that the PCU does not process the frame and, if required, respond appropriately. For example, if the host has configured the PCU not to pass ACK control frames to the host (the expected situation in normal system operation), the PCU still responds appropriately to incoming ACKs.

7.13 Frame Transmission Procedure

"DCU Channel Arbitration Procedure" on page 58 discusses the information the DCU passes to the PCU when requesting that the PCU attempt to transmit a frame. Once the PCU has completed the frame transmission attempt, it must report the results to the DCU that sourced the frame.

The transmission attempt results include:

- Transmit result
- The remaining status indications as specified in the Tx descriptor completion status

Possible transmit results include:

- Sent successfully (that is, sent on the air and received a valid ACK if one was expected)
- Sent on the air, but no ACK was received
- Never sent on the air because the Tx descriptor RTSEn bit was set. An RTS was sent on the air, but no CTS was received

7.14 PHY Errors

The 32-bit PHY Error Mask provides the ability to choose which PHY errors from the baseband will be filtered. The error number is used an offset into this mask. If the mask value at the offset is 0, then this error will be filtered and not show up on the receive queue. For more information on PHY errors, refer to "PHY Error Mask Register (PHYERR)" on page 173.

8. Digital PHY Block

The digital physical layer (PHY) block is described in 802.11a, 802.11b, and 802.11g modes. Transmit and receive paths are provided and shown as block diagrams for each mode.

8.1 802.11a Mode

The digital physical layer (PHY) block is a half-duplex, OFDM baseband processor compatible with IEEE 802.11a. All data rates defined by the IEEE 802.11a standard are supported (6 to 54 Mbps), including BPSK, QPSK, 16 QAM and 64 QAM modulation schemes, and forward error correction coding with rates of 1/2, 2/3, and 3/4.

In addition, enhanced turbo modes provide data rates up to 108 Mbps, which are higher than the rates specified by IEEE 802.11a.

Frames begin with training symbols used for signal detection, automatic gain control, frequency offset estimation, symbol timing, and channel estimation. The first data symbol is transmitted at the most robust rate (BPSK, 1/2 rate), and contains length and rate information for the remainder of the frame. This process uses 52 sub-carriers, 48 for data transmission and 4 for pilots.

8.1.1 Transmitter

The transmit path is shown in Figure 8-1.

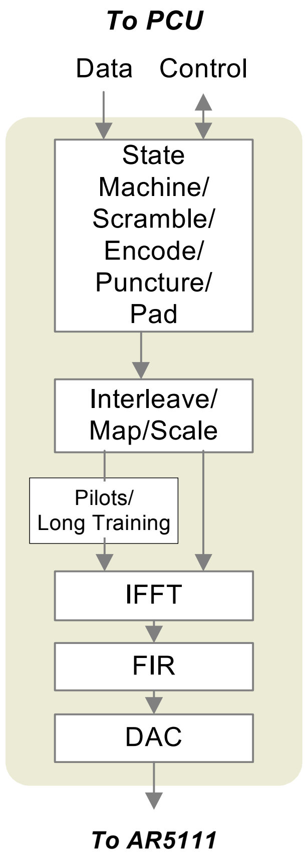


Figure 8-1. **Digital PHY 802.11a Transmit Block Diagram**

Transmission is initiated by the PCU block. The digital PHY powers on both the digital to analog converter (DAC) and transmit portions of the AR5111. The training symbols are a fixed waveform and are generated within the digital PHY in parallel with the PCU sending the Tx header (initial scrambler state, frame length, and data rate). The PCU must send the transmitted data quickly enough to prevent buffers in the digital PHY from becoming empty. The PCU is prevented from sending data too quickly by pauses generated within the digital PHY. The pace is determined by the number of data bytes consumed during each OFDM symbol.

Data bytes are processed as required by the 802.11a specification:

- Pad–Extend the number of data bits so that an integer number of OFDM symbols is used.
- Scramble–Randomize the data to avoid certain types of data-dependent transmission and reception problems.
- Encode–Provide redundancy so that bit errors can be corrected.
- Puncture–Optionally discard selected encoder outputs to increase throughput.
- Interleave–Re-order so that adjacent coded bits are mapped to non-adjacent sub-carriers, and not mapped to LSBs of constellations. The re-ordering distributes less robust bits, thereby improving error correction.
- Map and scale–Map the bits to gray-coded constellations, and scale constellation amplitude so that average power remains unchanged across constellation types.
- Pilot insertion–Insert pilots into fixed sub-carriers to aid reception.
- Perform an inverse fast fourier transform (IFFT)–All operations so far are considered to be in the frequency domain. Convert the frequency domain to a time domain waveform to be transmitted.
- Filter–The waveform must be windowed and filtered when being sent to the DAC.

This process continues for the number of symbols required to transmit the number of bytes at the specified rate. At the end of transmission, the DAC and Tx portions of the AR5111 are disabled, the ADC is enabled, and the signal detection logic in the receiver begins searching for the next incoming frame.

8.1.2 Receiver

The receive path is shown in Figure 8-2.

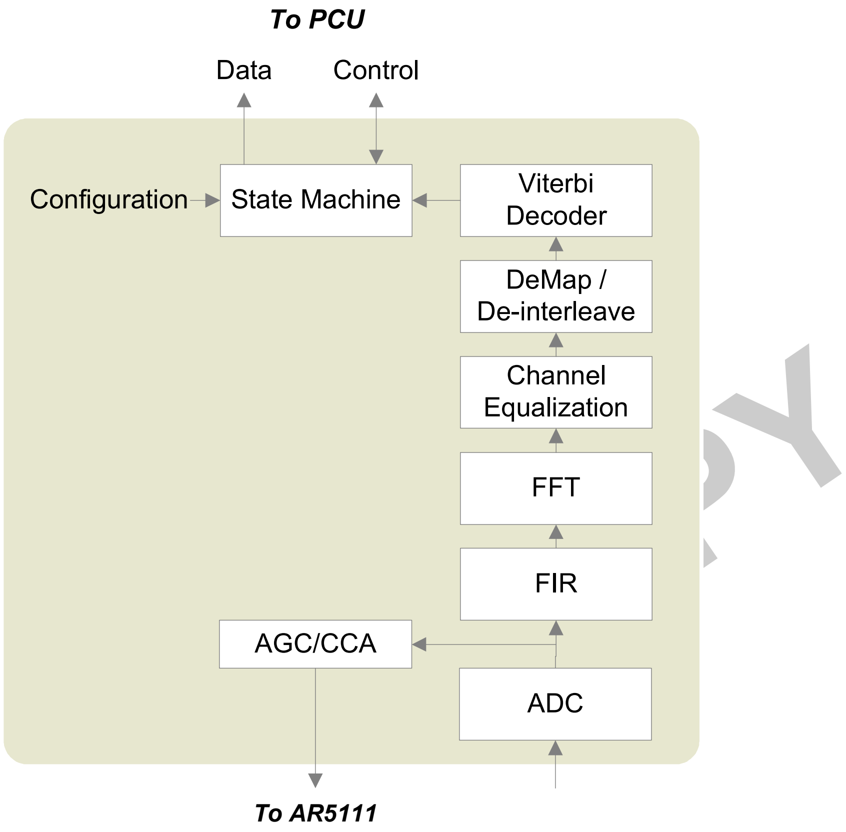


Figure 8-2. **Digital PHY 802.11a Receive Block Diagram**

- The receiver inverts the steps implemented by the transmitter: perform a fast fourier transform (FFT), extract bits from the received constellations, de-interleave, account for puncturing, viterbi decode, and descramble. Channel compensation is also required. The training symbols are used for:
- Signal detection
- Automatic gain control
- Diversity selection
- Frequency offset estimation
- Channel estimation
- Timing synchronization

8.2 802.11b Mode

The digital PHY incorporates a Direct Sequence Spread Spectrum (DSSS) transceiver, supporting all data rates defined by IEEE 802.11b (1, 2, 5.5, and 11 Mbps). Differential binary phase shift keying (DBPSK) is required for 1 Mbps. Differential quadrature phase shift keying (DQPSK) is required for 2 Mbps. Complementary code keying (CCK) is required for both 5.5 Mbps and 11 Mbps. A long preamble format only is supported for 1 Mbps. Both short and long preamble formats are supported for all other data rates.

Frames begin with a SYNC field (see 802.11b specification) used for signal detection, antenna diversity, automatic gain control, frequency offset estimation, timing and channel estimation. A start frame delimiter (SFD) sequence provides coarse symbol alignment preceding the PLCP header, which contains rate and length of the packet, as well as a SERVICE byte (see 802.11b specification) and 16-bit header CRC. Following the header is the payload of the packet.

8.2.1 Transmitter

The transmit path is shown in Figure 8-3.

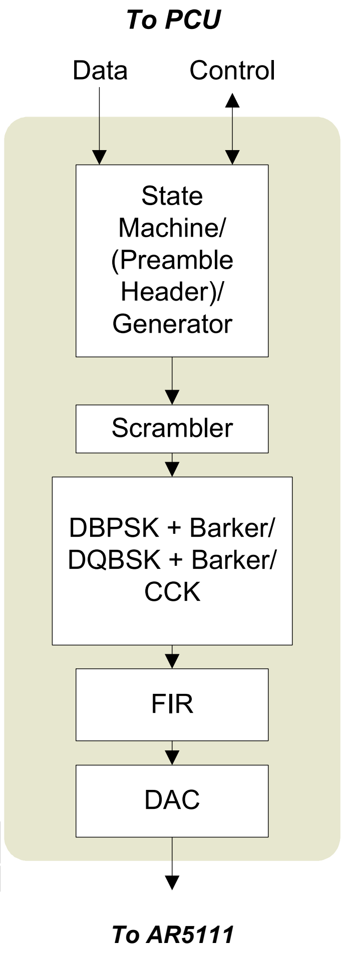

To PCU
Data
Control
State Machine/ (Preamble Header)/ Generator
Scrambler
DBPSK + Barker/ DQBSK + Barker/ CCK
FIR
DAC
To AR5111

Figure 8-3. **Digital PHY 802.11b Transmit Block Diagram**

Transmission is initiated by the PCU block. The preamble, header and payload are scrambled, encoded, and spread with an 11-chip Barker sequence for 1 and 2 Mbps, and an 8-chip sequence for both 5.5 and 11 Mbps. The transmitted waveform is filtered prior to the DAC.

8.2.2 Receiver

The receive path is shown in Figure 8-4.

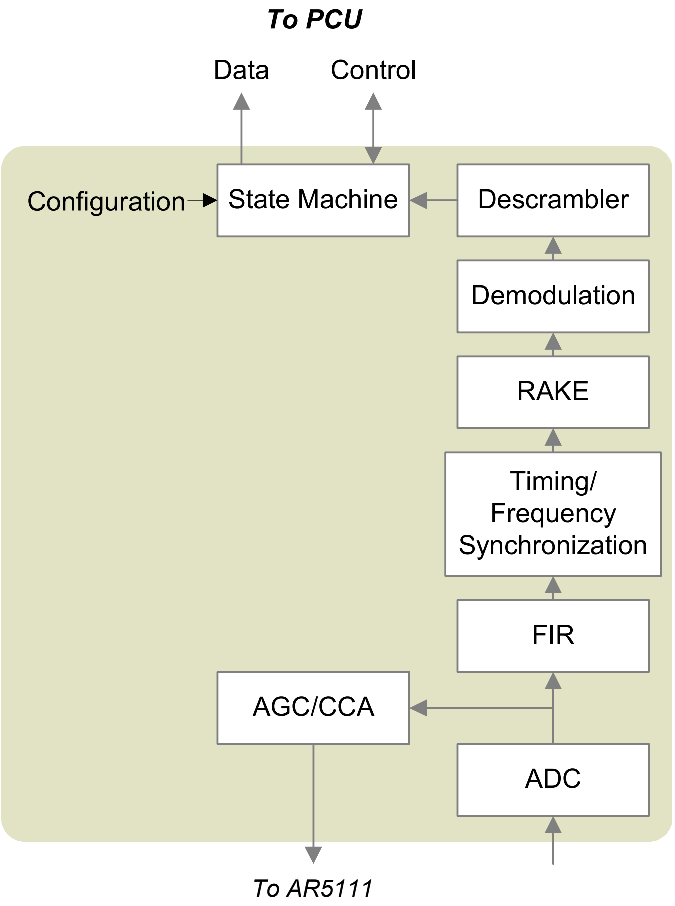


Figure 8-4. **Digital PHY 802.11b Receive Block Diagram**

The receiver performs analog to digital conversion of differential I and Q signals from the AR5111. Signal detection is performed, and if fast receive antenna diversity is enabled, both antennas are scanned for the highest quality signal.

The preamble is descrambled, and a determination of short or long preamble is made. The data rate and frame length are extracted from the physical layer convergence protocol (PLCP) header, qualified by a 16-bit cyclic redundancy code (CRC). If the CRC passes, and the rate and SERVICE fields are legal, decoding of the packet proceeds.

8.3 802.11g Mode

In 802.11g mode, the digital PHY is a superset of that required for 802.11a and 802.11b.

8.3.1 Transmitter

On a per packet basis, the transmitter can dynamically switch between generating OFDM and DSSS signals. The OFDM and DSSS signals are muxed just before the DAC. The transmit path is shown in Figure 8-5.

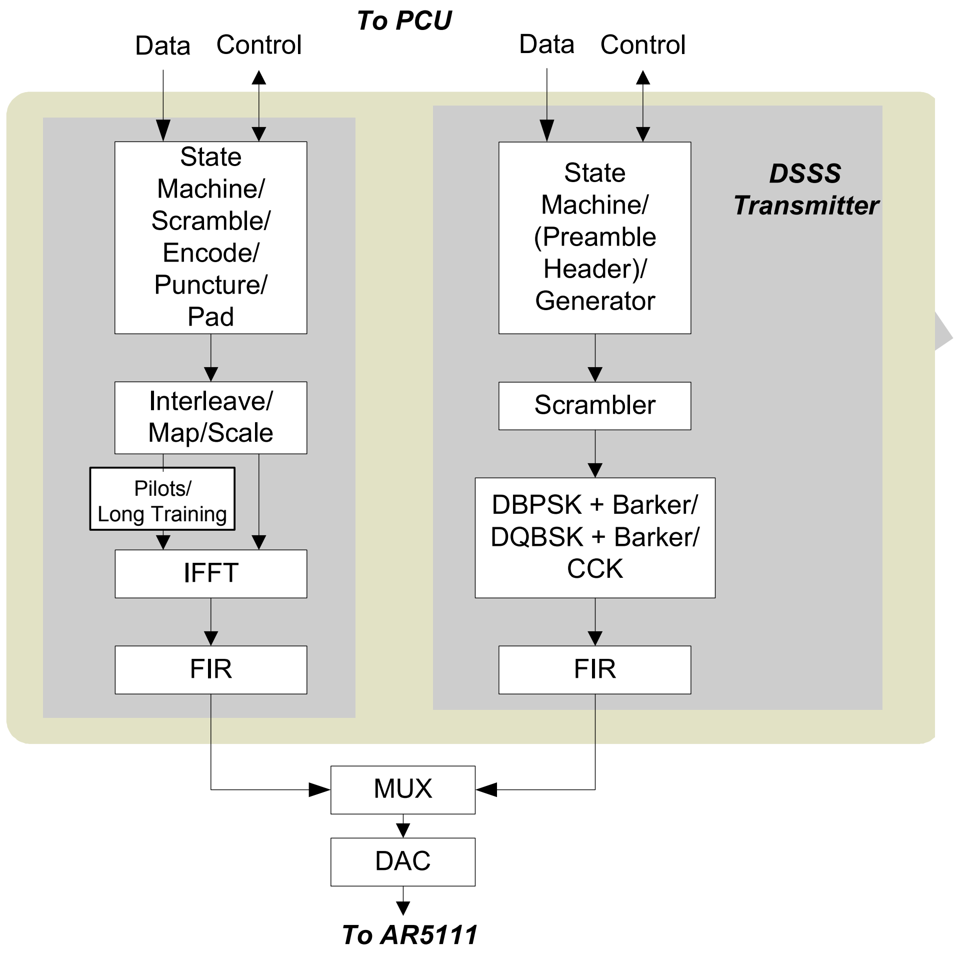

To PCU
Data
Control
Data
Control
State Machine/ Scramble/ Encode/ Puncture/ Pad
Interleave/ Map/Scale
Pilots/ Long Training
IFFT
FIR
State Machine/ (Preamble Header)/ Generator
DSSS Transmitter
Scrambler
DBPSK + Barker/ DQBSK + Barker/ CCK
FIR
MUX
DAC
To AR5111

Figure 8-5. **Digital PHY 802.11g Transmit Block Diagram**

8.3.2 Receiver

While listening for incoming packets, the receiver simultaneously searches for OFDM and DSSS signals. If an OFDM signal is found, the rest of the OFDM receiver is enabled, and the DSSS receiver is disabled. If a DSSS signal is found, the rest of the DSSS receiver is enabled, and the OFDM receiver is disabled. Once reception is complete, simultaneous searching for OFDM and DSSS signals resumes. The receive path is shown in Figure 8-6.

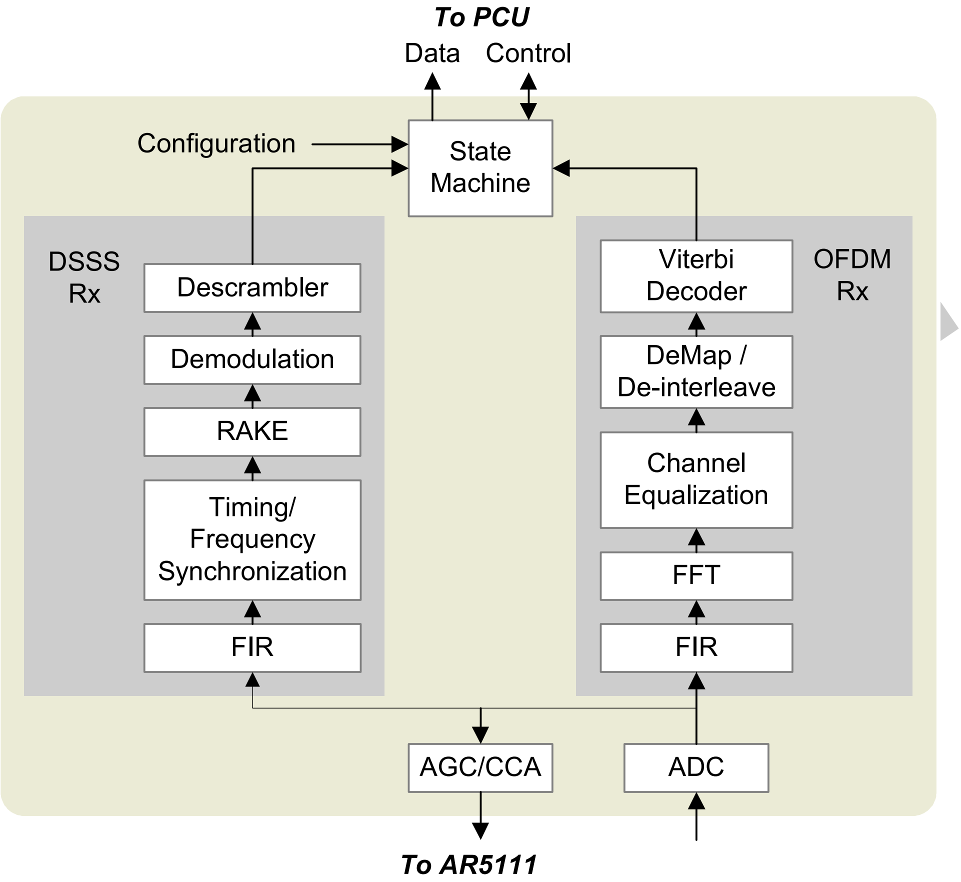


Figure 8-6. **Digital PHY 802.11g Receive Block Diagram**

9. Register Descriptions

The following sections describe the internal registers for the various blocks of the AR5212. The offset column refers to the offset from the base address configured by the host in the PCI base address configuration register (refer to "Base Address" on page 85). For CardBus applications, the AR5212 internal registers are mapped to the memory space of the host by specifying the required information using the card information structure (CIS). A pointer to the CIS is provided in the CIS register (refer to "CIS Pointer" on page 85).

The internal registers are divided into the following groups:

- "PCI Configuration Space Registers" on page 79
- "Host Interface and Receive Registers" on page 91
- "QCU Registers" on page 118
- "DCU Registers" on page 126
- "PCI Clock Domain Registers" on page 139
- "EEPROM Interface Registers" on page 149
- "PCU Registers" on page 152

9.1 PCI Configuration Space Registers

Table 9-1 summarizes the PCI Configuration Space registers for the AR5212. These registers are accessed by the host using PCI configure operations and are used at boot time by the host to detect the type of card present and to perform low-level configuration, such as assigning base addresses to the card. At reset, some of these registers are initialized from an off-chip serial EEPROM, while others must be programmed by the host or are initialized by the AR5212 hardware. Refer to "EEPROM PCI/CardBus Partition" on page 24 for register values that are loaded from the EEPROM upon reset. Registers that are loaded by the host or initialized by the AR5212 are identified in "PCI Configuration Space Register Summary" on page 79.

Refer to version 2.3 of the PCI bus standard for detailed information on these registers.

Table 9-1. **PCI Configuration Space Register Summary**

Offset	Name	Description	Initialized by	Page
0x00	Vendor ID	Identification of the manufacturer.	EEPROM	page 81
0x02	Device ID	Identification of the device type.	EEPROM	page 81
0x04	Command	Controls accessibility of the device.	Host	page 81
0x06	Status	Provides status of the device's functionality.	AR5212	page 82
0x08	Revision ID	Identification of the device's revision.	EEPROM	page 83
0x09	Class Code	Identification of the device's basic function.	EEPROM	page 84
0x0C	Cache Line Size	Specifies system cache line size.	Host	page 84
0x0D	Latency Timer	Defines the minimum time (in PCI bus cycles) that the bus master can retain ownership of the PCI bus.	Host	page 84
0x0E	Header Type	Defines device's configuration header format.	EEPROM	page 85
0x0F	Reserved	Not used.		
0x10	Base Address	Base address for accessing the memory mapped registers.	Host	page 85
0x14 to 0x27	Reserved	Not used.		
0x28	CIS Pointer	Pointer to the device's Card Information Structure.	EEPROM	page 85
0x2C	Subsystem Vendor ID	Identification of the subsystem manufacturer.	EEPROM	page 86
0x2E	Subsystem ID	Identification of the subsystem device type.	EEPROM	page 86

Table 9-1. **PCI Configuration Space Register Summary (continued)**

Offset	Name	Description	Initialized by	Page
0x34	Capabilities Pointer	Pointer to the device's list of capabilities.	AR5212	page 86
0x38 to 0x3B	Reserved	Not used.		
0x3C	Interrupt Line	Defines if the device's interrupts are generated using the PCI interrupt pins, or using message signaled interrupts (MSI) capability.	Host	page 87
0x3D	Interrupt Pin	Defines specific PCI interrupt pins that are associated with particular functions of the device.	EEPROM	page 87
0x3E	MinGnt	Indicates how long the device retains the PCI bus.	EEPROM	page 87
0x3F	MaxLat	Indicates how often the device accesses the PCI bus.	EEPROM	page 88
0x40	CFG_TIMER	PCI retry limit and TRDY timeout counters.	AR5212	page 88
0x42 to 0x43	Reserved	Not used.		
0x44	CFG_PMCAP_ID	PCI configuration power management capability ID.	AR5212	page 88
0x45	CFG_PMCAP_PTR	PCI configuration power management capability pointer.	AR5212	page 89
0x46	CFG_PMCAP	PCI configuration power management capabilities register.	EEPROM	page 89
0x48	CFG_PMCSR	PCI configuration power management control and status.	EEPROM	page 89
0x4A	CFG_PMCSR_ESE	PCI configuration power management bridge support extensions.	AR5212	page 90
0x4B	CFG_PMDATA	PCI configuration power management data.	EEPROM	page 90
0x4C to 0xFF	Reserved	Not used.		

9.1.1 Vendor ID

This register contains the vendor identification number. Default value of this register is loaded from the EEPROM.

Address/offset: 0x00
Access: Read Only
Size: 16 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
15–0	VENDOR_ID	Vendor identification.

9.1.2 Device ID

This register identifies the device type. The default value of this register is loaded from the EEPROM.

Address/offset: 0x02
Access: Read Only
Size: 16 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
15–0	DEVICE_ID	Device identification.

9.1.3 Command Register

This register provides access control of the AR5212 PCI interface. The register is controlled by the host.

Address/offset: 0x04
Access: Read/Write
Size: 16 bits
Reset Value: Undefined

Bit	Bit Name	Description
0	IO_SPACE	I/O Space. 0 = Disable. 1 = Enable.
1	MEM_SPACE	Memory Space. 0 = Disable. 1 = Enable.
2	BUS_MSTR	Bus Master. 0 = Disable. 1 = Enable.

Bit	Bit Name	Description
3	SPEC_CYCLES	Special Cycles. 0 = Disable. 1 = Enable.
4	MEM_WR_INV	Memory Write and Invalidate Enable. 0 = Disable. 1 = Enable.
5	VGA_SNOOP	VGA Palette Snoop. 0 = Disable. 1 = Enable.
6	PAR_ERR_RESP	Parity Error Response. 0 = Disable. 1 = Enable.
7	STEP_CNTL	Stepping Control. 0 = Disable. 1 = Enable.
8	SERR_EN	System Error Enable. 0 = Disable. 1 = Enable.
9	FAST_BB_EN	Fast Back-to-Back Enable. 0 = Disable. 1 = Enable.
15–10	RES	Reserved. Must be written with zero. On read, can contain any value.

9.1.4 Status

This register provides status of the functionality provided by the AR5212 PCI interface. This register is mostly controlled by the AR5212.

Address/offset: 0x06
Access: Read/Write, except as noted
Size: 16 bits
Reset Value: 0x0290

Bit	Bit Name	Description
3–0	RES	Reserved. Must be written with zero. On read, can contain any value.
4	CAP_LIST	Capabilities list. Read only.
5	66MHZ_EN	66 MHz capable. Read only.
6	RES	Reserved. Must be written with zero. On read, can contain any value.
7	FAST_BB	Fast back-to-back capable. Read only. 0 = Disabled. 1 = Enabled.
8	MD_PAR_ERR	Master Data Parity Error. *On Read:* *On Write:* 0 = No error. 0 = Do not clear bit. 1 = Error. 1 = Clear error bit.

Bit	Bit Name	Description
10–9	DEVSEL_TIMING	Device Select Timing. Read only. 01 = Medium
11	SIG_TARG_ABORT	Signaled Target Abort. *On Read:* *On Write:* 0 = No abort. 0 = Do not clear bit. 1 = Abort. 1 = Clear abort bit.
12	RX_TARG_ABORT	Received Target Abort. *On Read:* *On Write:* 0 = No abort. 0 = Do not clear bit. 1 = Abort. 1 = Clear abort bit.
13	RX_MAS_ABORT	Received Master Abort. *On Read:* *On Write:* 0 = No abort. 0 = Do not clear bit. 1 = Abort. 1 = Clear abort bit.
14	SIG_SYS_ERR	Signaled System Error. *On Read:* *On Write:* 0 = No error. 0 = Do not clear bit. 1 = Error. 1 = Clear error bit.
15	DETECT_PAR_ERR	Detected Parity Error. *On Read:* *On Write:* 0 = No error 0 = Do not clear bit 1 = Error 1 = Clear error bit

9.1.5 Revision ID

This register contains the device revision identification number. Default value is loaded from EEPROM.

Address/offset: 0x08
Access: Read/Write
Size: 8 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
7–0	REVISION_ID	Revision identification.

9.1.6 Class Code

This register contains the class code identification number that identifies the basic function of the device. Default value is loaded from the EEPROM.

Address/offset: 0x09
Access: Read only
Size: 24 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
23–0	CLASS_CODE	Class code identification value.

9.1.7 Cache Line Size

This register contains the size of the system cache line. This register is controlled by the host.

Address/offset: 0x0C
Access: Read/Write
Size: 8 bits
Reset Value: 0x00

Bit	Bit Name	Description
7–0	CACHE_SZ	Cache line size, in units of 32-bit words (4 bytes).

9.1.8 Latency Timer

This register provides the minimum amount of time, in PCI clock cycles, that the bus master can retain ownership of the bus whenever it initiates a new transaction. This register is controlled by the host.

Address/offset: 0x0D
Access: Read/Write
Size: 8 bits
Reset Value: 0x00

Bit	Bit Name	Description
7–0	LATENCY_TMR	Latency timer.

9.1.9 Header Type

This register contains the header type information. Default value is loaded from the EEPROM.

Address/offset: 0x0E
Access: Read Only
Size: 8 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
7–0	HDR_TYPE	Header type. 0 = Nonbridge PCI device.

9.1.10 Base Address

This register contains the base address for accessing the AR5212 registers. This register is controlled by the host.

Address/offset: 0x10
Access: Bits [15:0] are Read Only (always return 0)
Bits [31:16} are Read/Write
Size: 32 bits
Reset Value: Undefined

Bit	Bit Name	Description
31–0	BASE_ADDR	Base address.

9.1.11 CIS Pointer

This register contains the value of the CIS pointer. Default value is loaded from the EEPROM.

Address/offset: 0x28
Access: Read Only
Size: 32 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
31–0	CIS_PTR	CIS pointer value.

9.1.12 Subsystem Vendor ID

This register contains the subsystem vendor identification number. Default value is loaded from the EEPROM.

Address/offset: 0x2C
Access: Read Only
Size: 16 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
15–0	SSYS_VEND_ID	Subsystem vendor ID.

9.1.13 Subsystem ID

This register contains the subsystem device identification number. Default value is loaded from the EEPROM.

Address/offset: 0x2E
Access: Read Only
Size: 16 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
15–0	SSYS_ID	Subsystem ID.

9.1.14 Capabilities Pointer (CAP_PTR)

This register contains the value of the capabilities pointer. Default value is provided by the AR5212.

Address/offset: 0x34
Access: Read Only
Size: 8 bits
Reset Value: 0x44

Bit	Bit Name	Description
7–0	CAP_PTR	Capabilities pointer value.

9.1.15 Interrupt Line (INT_LINE)

This register contains the host interrupt controller's interrupt line value that the device's interrupt pin is connected to. This register is controlled by the host.

Address/offset: 0x3C
Access: Read/Write
Size: 8 bits
Reset Value: 0x00

Bit	Bit Name	Description
7–0	INT_LINE	Interrupt line value.

9.1.16 Interrupt Pin (INT_PIN)

This register defines which of the four PCI interrupt request pins, a PCI function is connected to. Default is loaded from the EEPROM.

Address/offset: 0x3D
Access: Read only
Size: 8 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
7–0	INT_PIN	Interrupt pin value.

9.1.17 MinGnt

This register contains a value that indicates how long the device (bus-master) retains PCI bus ownership. Default is loaded from the EEPROM.

Address/offset: 0x3E
Access: Read Only
Size: 8 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
7–0	MIN_GNT	Minimum grant value.

9.1.18 MaxLat

This register contains the maximum latency value. Default is loaded from the EEPROM.

Address/offset: 0x3F
Access: Read Only
Size: 8 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
7–0	MAX_LAT	Maximum latency value.

9.1.19 Configuration Timer (CFG_TIMER)

This register contains the retry limit and TRDY timeout counters. Default value is provided by the AR5212.

Address/offset: 0x40
Access: Read/Write
Size: 32 bits
Reset Value: 0x0000_FF80

Bit	Bit Name	Description
15–0	CFG_TIMER	Retry limit and TRDY timeout counters.
31–16	RES	Reserved. Must be written with zero. Can contain any value on read.

9.1.20 Power Management Capability ID (CFG_PMCAP_ID)

The register contains the power management capability identification. This register is read only and default value is provided by the AR5212.

Address/offset: 0x44
Access: Read Only
Size: 8 bits
Reset Value: 0x01

Bit	Bit Name	Description
7–0	PMCAP_ID	Power management capability ID, always return 0x01 when read.

9.1.21 Next Capability Pointer (CFG_PMCAP_PTR)

This register contains the pointer to the next capability. This register is read only and default value is provided by the AR5212.

Address/offset: 0x45
Access: Read Only
Size: 8 bits
Reset Value: 0x00

Bit	Bit Name	Description
7–0	PMCAP_PTR	Pointer to the next power management capability, always return 0x00 when read.

9.1.22 Power Management Capabilities (CFG_PMCAP)

This register contains the details of the power management functions supported by the device. Default value provided by the EEPROM.

Address/offset: 0x46
Access: Read Only
Size: 16 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
15-0	PM_CAP	Describes power management capabilities.

9.1.23 Power Management Control/Status (CFG_PMCSR)

This register provides power management control and status information of the device. Default value provided by the EEPROM.

Address/offset: 0x48
Access: Read/Write
Size: 16 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
1–0	PWR_STATE	Power state. Controls power states supported by the device. 00 D0 01 D1 (not supported) 10 D2 (not supported) 11 $D3_{hot}$
7-2	RES	Reserved. Must be written with zero. Can contain any value on read.
8	PME_EN	Not supported.

Bit	Bit Name	Description
12-9	DATA_SEL	Selects data item reported through CFG_PMData register.
14-13	PM_DATA_SCALE	Data scale. Read only. Data read from CFG_PMData register must be multiplied by this factor.
15	PME_STATUS	Not supported.

9.1.24 Power Management Bridge Support Extensions (CFG_PMCSR_ESE)

This register contains the PCI to PCI bridge support extension. This register is read only. Default value is provided by the AR5212.

Address/offset: 0x4A
Access: Read only
Size: 8 bits
Reset Value: 0x00

Bit	Bit Name	Description
7–0	CFG_PMCSR_ESE	PCI to PCI bridge support extension, always return 0x00 when read.

9.1.25 Power Management Data Register (CFG_PMDATA)

This register contains the power consumed, dissipated and other device-specific operational information. Default value is provided by the EEPROM.

Address/offset: 0x4B
Access: Read only
Size: 8 bits
Reset Value: Refer to "EEPROM PCI/CardBus Partition" on page 24.

Bit	Bit Name	Description
7–0	PM_DATA	Provides power consumption, dissipation or other device-specific operational information in selected PM state. PM states are selected by the DATA_SEL bits in the CFG_PMCSR register.

9.2 Host Interface and Receive Registers

Table 9-2 summarizes the host Rx registers and descriptions of each of these registers, provided by the AR5212. Host interface and Rx registers are located at offset addresses 0x0008 to 0x00F8.

Table 9-2. **Host Interface and Receive Register Summary**

Offset	Name	Description	Page
0x0008	CR	Command register. Controls the receive FIFO (RxFIFO) and generates software interrupts. RxE enables RxFIFO. RxD is used to disable RxFIFO and gracefully stop writing Rx frames.	page 93
0x000C	RXDP	Receive queue descriptor pointer. Specifies the address of the current Rx descriptor.	page 94
0x0014	CFG	Configuration and status register.	page 94
0x0024	IER	Global interrupt enable register. Controls whether the internal interrupts are allowed to generate an external interrupt on the PCI hardware interrupt line.	page 95
0x0030	TXCFG	Transmit configuration register.	page 96
0x0034	RXCFG	Receive configuration register.	page 97
0x0038	RXJLA	Receive jumbo descriptor last address register. Specifies address of last word written in receive jumbo mode.	page 98
0x0040	MIBC	MIB control register. Control the initialization and update of the MIB counters.	page 98
0x0044	TOPS	Timeout prescale register. Used to prescale the system clock (40 MHz in normal mode, 80 MHz in turbo mode) which is then used as a clock for RXNF, TXNF, and RFGTO.	page 99
0x0048	RXNOFR	RXNOFR timeout register. Generates an interrupt if no Rx frame (successful or error) is received within a programmable timeout.	page 99
0x004C	TXNOFR	TXNOFR timeout register. Generates an interrupt if no Tx frame was sent within a programmable timeout.	page 100
0x0050	RFGTO	Receive frame gap timeout.	page 100
0x0054	RFCNT	Rx frame count register. Sets the number of frames to be received before generating an RXDESC interrupt.	page 101
0x0058	MACMISC	MAC-specific miscellaneous status/control register.	page 101
0x005C	SPC_0	Sleep performance counter 0.	page 102
0x0060	SPC_1	Sleep performance counter 1.	page 102
0x0080	ISR_P	Primary interrupt status register. Summary of all pending interrupts.	page 103
0x0084	ISR_S0	Secondary interrupt status register 0. Tracks the TXOK and TXDESC interrupts from the individual Tx queues.	page 105
0x0088	ISR_S1	Secondary interrupt status register 1. Tracks the TXERR and TXEOL interrupts from the individual Tx queues.	page 106
0x008C	ISR_S2	Secondary interrupt status register 2. Tracks the TXURN and PCI error interrupts from the individual Tx queues.	page 106
0x0090	ISR_S3	Secondary interrupt status register 3. Tracks the QCBROVF and QCBRURN interrupts from the individual Tx queues.	page 107

Table 9-2. **Host Interface and Receive Register Summary (continued)**

Offset	Name	Description	Page
0x0094	ISR_S4	Secondary interrupt status register 4. Tracks the QTRIG interrupts from the individual Tx queues.	page 107
0x00A0	IMR_P	Primary interrupt mask register. Used to mask individual pending interrupts from the PCI hardware interrupt.	page 108
0x00A4	IMR_S0	Secondary interrupt mask register 0. Used to keep pending TXOK_SI and TXDESC_SI interrupts in ISR_S0 from appearing in TXOK_PI and TXDESC_PI of ISR_P.	page 110
0x00A8	IMR_S1	Secondary interrupt mask register 1. Used to keep pending TXERR_SI and TXEOL_SI interrupts in ISR_S1 from appearing in TXERR_PI and TXEOL_PI of ISR_P.	page 110
0x00AC	IMR_S2	Secondary interrupt mask register 2. Used to keep pending TXURN_SI, MCABT_SI, SERR_SI, and DPERR_SI interrupts in ISR_S2 from appearing in TXURN_PI and HIUERR_PI of ISR_P.	page 111
0x00B0	IMR_S3	Secondary interrupt mask register 3. Used to keep pending QCBROVF_SI, and QCBRURN_SI interrupts in ISR_S3 from appearing in QCBROVF_PI and QCBRURN_PI of ISR_P.	page 112
0x00B4	IMR_S4	Secondary interrupt mask register 4. Used to keep pending QTRIG_SI interrupts in ISR_S4 from appearing in QTRIG_PI of ISR_P.	page 112
0x00C0	ISR_P_RAC	Primary RAC interrupt status register. Summary of all pending interrupts.	page 113
0x00C4	ISR_S0_S	Secondary shadow interrupt status register 0. Summary of TXOK_SI and TXDESC_SI interrupts from the individual Tx queues.	page 115
0x00C8	ISR_S1_S	Secondary shadow interrupt status register 1. Summary of TXERR_SI and TXEOL_SI interrupts from the individual Tx queues.	page 115
0x00CC	ISR_S2_S	Secondary shadow interrupt status register 2. Summary of TXURN_SI interrupts from the individual Tx queues and MCABT_SI, SERR_SI, and DPERR_SI interrupts.	page 116
0x00D0	ISR_S3_S	Secondary shadow interrupt status register 3. Summary of QCBROVF_SI and QCBRURN_SI interrupts from the individual Tx queues.	page 116
0x00D4	ISR_S4_S	Secondary shadow interrupt status register 4. Summary of QTRIG_SI interrupts from the individual Tx queues.	page 117

9.2.1 Command Register (CR)

This register enables and disables Rx and Tx queues. It also provides a software interrupt.

Address offset: 0x0008
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
1–0	RES	Reserved. Must be written with zero. Can contain any value on read.
2	RxE	Receive Enable. 0 = Ignore. 1 = Enable Rx FIFO.
4–3	RES	Reserved. Must be written with zero. Can contain any value on read.
5	RxD	Receive disable. 0 = Allow Rx FIFO to be enabled. 1 = Disable Rx FIFO.
6	SWI	Software interrupt. (one-shot/automatically cleared by hardware so always reads as 0). 0 = Ignore. 1 = Trigger software interrupt.
31–7	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.2 Receive Queue Descriptor Pointer (RXDP)

This register contains the value of the Rx queue descriptor pointer.

Address offset: 0x000C
Access: Read/Write
Cold reset: Undefined
Warm reset: Unaffected

Bit	Bit Name	Description
1–0	RES	Reserved. Must be written with zero. Can contain any value on read.
31–2	RXDP	Rx descriptor pointer.

9.2.3 Configuration and Status Register (CFG)

This register configures the Tx and Rx descriptor operations, selects the enhanced turbo mode (108 Mbps operation), and provides status regarding EEPROM and host memory operation.

Address offset: 0x0014
Access: Read/Write
Cold reset: 0x0000_0100
Warm reset: 0x0000_0100

Bit	Bit Name	Description
0	SWTD	0 = Disable. 1 = Byteswap Tx descriptor words.
1	SWTB	0 = Disable. 1 = Byteswap Tx data buffer words.
2	SWRD	0 = Disable. 1 = Byteswap Rx descriptor words.
3	SWRB	0 = Disable. 1 = Byteswap Rx data buffer words.
4	SWRG	0 = Disable. 1 = Byteswap register access words.
5	ADHOC	AP/AdHoc indication. 0 = AP mode: MAC is operating either as an AP or as a STA in a BSS. 1 = AdHoc mode: MAC is operating as a STA in an IBSS.
7–6	RES	Reserved. Must be written with zero. Can contain any value on read.
8	PHY_OK	Currently hardwired to ‘1’.
9	EEBS	EEPROM busy. Indicates whether the PCI/CardBus Interface is accessing off-chip serial EEPROM. Resets to 0x1, but will clear when PCI/CardBus Interface has completed loading the EEPROM contents after the negation of PCI_RST_L. 0 = EEPROM is idle. 1 = EEPROM is busy.

Bit	Bit Name	Description
10	CLKGATEDIS	Clock gating disable. 0 = Allow clock gating in the DRU, DCU, DCU Arb, and QCU blocks to operate normally. 1 = Disable clock gating in the DRU, DCU, DCU Arb, and QCU blocks. For debug.
16–11	RES	Reserved. Must be written with zero. Can contain any value on read.
18–17	PCITHR	Controls how many requests the DRU and QCUs can have pending in the PCI core. 0 = Allow up to 4 pending requests. 1 = Allow up to 1 pending requests. 2 = Allow up to 2 pending requests. 3 = Allow up to 3 pending requests.
31–19	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.4 Global Interrupt Enable Register (IER)

This register globally enables or disables hardware interrupts.

Address offset: 0x0024
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
0	IER	Enable PCI interrupt. 0 = Disable hardware interrupt. 1 = Enable hardware interrupt.
31–1	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.5 Transmit Configuration Register (TXCFG)

This register configures the Tx operation.

Address offset: 0x0030
Access: Read/Write
Cold reset: 0x0000_0015
Warm reset: 0x0000_0015

Bit	Bit Name	Description
2–0	SDMAMR	Maximum Burst Size for Master Reads. 000 = 4 bytes 001 = 8 bytes 010 = 16 bytes 011 = 32 bytes 100 = 64 bytes 101 = 128 bytes 110 = 256 bytes 111 = 512 bytes
3	RES	Reserved. Must be written with zero. Can contain any value on read.
9–4	TXFULL	Frame trigger level. Specifies the minimum number of bytes, in units of 64 bytes, that must be written into the PCU TxFIFO before the PCU will initiate sending the frame on the air.
10	JUMBOTXD	Jumbo descriptor mode enable. 0 = QCUs treat the BufLen field normally 1 = QCUs scale the Tx descriptor BufLen field by 4096. For example, if the Tx descriptor has a BufLen value of 100, then the QCUs will act as if the BufLen were 100*4096, or 409600, bytes). This is a debug mode only.
11		Adhoc beacon ATIM window transmission policy. 0 = If the ATIM window ends before the station can send its beacon, the station cancels its beacon transmission. 1 = Station continues to attempt to send its beacon until it is able to do so, regardless of the status of the ATIM window.
12		Fragment burst versus ATIM window defer disable. 0 = In Adhoc mode only, if the ATIM window begins in the middle of a fragment burst, halt the burst and allow frames from other DCUs (for example, DCUs generating the beacon and CAB traffic) to proceed. Resume the fragment burst after the ATIM window ends and after following the normal DCF channel access procedure. 1 = Pause the fragment burst for the duration of the ATIM window, but do not allow frames from other DCUs to appear on the air. Meant as a debugging mode or if a problem is suspected with the fragment burst deferral logic.
13	RES	Reserved. Must be written with zero. Can contain any value on read.
14	RDYDIS	ReadyTime/CBR disable for QCUs 8–9. When the MAC is running at a clock rate of 32 MHz or slower, this bit must be set and only the ASAP frame scheduling policy may be selected for QCUs 8-9. QCUs 0-7 may continue to use any frame scheduling policy. This is meant as a debugging mode only, because the MAC clock rate is at least 40 MHz in normal operation. 0 = MAC clock rate is at least 33 MHz. Enable all frame scheduling policies for all QCUs. 1 = MAC clock rate is 32 MHz or slower. Disable non-ASAP FSP for QCUs 8-9 so that CBR and ReadyTime logic will continue to operate correctly for QCUs 0-7.

Bit	Bit Name	Description
15		DCU double-buffering disable. 0 = Allow the DCUs to use both of the PCU transmit FIFOs so that while one frame is being transmitted, the next frame can be DMA'ed (frame double buffering). 1 = Force the DCUs to use only one of the PCU's transmit FIFOs. Debugging mode only, or if a problem is suspected with the DCU/PCU interaction with frame double-buffering.
16		DCU intraframe caching disable. Debugging purposes only.
31–17	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.6 Receive Configuration Register (RXCFG)

This register configures the receive operation.

Address offset: 0x0034
Access: Read/Write
Cold reset: 0x0000_0005
Warm reset: 0x0000_0005

Bit	Bit Name	Description
2–0	SDMAMW	Maximum Burst Size for Master Writes. 000 = 4 bytes 001 = 8 bytes 010 = 16 bytes 011 = 32 bytes 100 = 64 bytes 101 = 128 bytes 110 = 256 bytes 111 = 512 bytes
4–3	RXZERO	Zero Length Frame DMA Enable. 0 = Disable DMA of all zero-length frames. DMA logic will suppress all zero-length frames. Reception of zero-length frames will be invisible to the host (they will neither appear in host memory nor consume a receive descriptor). 1 = Enable chirp/double-chirp DMA only. Only chirps and double-chirps are DMA'ed into host memory like normal (non-zero-length) frames; all other zero-length frames are suppressed. A chirp or double-chirp zero-length frame can be identified by its receive descriptor, which will have a DataLen field of zero, the More bit clear, the Done and PHYErr bits set, and the PHYErr code set to the value for chirp or double-chirp. The PHYErr code is 0x08 for chirp and 0x03 for double-chirp. 2 = Enable DMA of all zero-length frames. All zero-length frames will be DMA'ed into host memory just like normal (non-zero-length) frames. 3 = Reserved.
5	JUMBORXD	Jumbo descriptor mode enable. 0 = DRU treats the BufLen field normally. 1 = DRU scales the Rx descriptor BufLen field by 4096. For example, if the Rx descriptor has a BufLen value of 100, then the DRU will act as if the BufLen were 100*4096, or 409,600, bytes. This is a debug mode only.
6	JUMBOWRP	0 = After reaching end of jumbo descriptor's data buffer, proceed to next descriptor. 1 = After reaching end of jumbo descriptor's data buffer, re-transfer the same descriptor's data buffer again. This means the descriptor's data buffer will be overwritten with data from the PCU repeatedly in an infinite loop.

Bit	Bit Name	Description
7		Sleep entry policy when frames are pending in the PCU RX FIFO. 0 = The DMA receive logic will require all frames to be drained from the PCU's RX FIFO before allowing the chip to sleep (default). 1 = The DMA receive logic will allow the chip to sleep even when frames are pending in the PCU's RX FIFO. Debugging mode only, or if there is suspected operation in the DMA's tracking of the PCU's RX FIFO frame count.
31–8	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.7 Receive Jumbo Descriptor Last Address Register (RXJLA)

This register contains the value of the receive queue descriptor pointer.

Address offset: 0x0038
Access: Read Only
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
31–0	RXJLA	Address of last word written in receive jumbo mode. See JUMBORXD, bit 5, of "Receive Configuration Register (RXCFG)" on page 97. Valid only after jumbo mode has been entered and then exited by setting and then clearing the JUMBORXD bit of RXCFG.

9.2.8 Management Information Base Control Register (MIBC)

This register controls the MIB counters.

Address offset: 0x0040
Access: Read/Write
Cold reset: 0x0000_0006
Warm reset: 0x0000_0006

Bit	Bit Name	Description
0	COW	Counter Overflow Warning. 0 = All counters are below the warning threshold. 1 = At least one counter has reached the warning threshold.
1	FMC	Freeze MIB Counters. 0 = Allow MIB counters to update. 1 = MIB counters frozen at current value.
2	CMC	Clear MIB Counters. 0 = Allow MIB counters to update. 1 = MIB counters forced to zero.
3	MCS	MIB Counter Strobe. This bit is a one-shot and always reads as zero. 0 = No effect. 1 = Increment all MIB counters by one.
31–4	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.9 Timeout Prescale Register (TOPS)

This register sets the prescale count for interrupt-related timeouts.

Address offset: 0x0044
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
15–0	TOPS	Timeout prescale count. Value for interrupt-related timeouts in core clock cycles. A value of zero disables the prescale counter.
31–16	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.10 RXNOFR Timeout Register (RXNF)

This register sets the time to wait before generating an interrupt to indicate that no frame has been received.

Address offset: 0x0048
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	RXNOFRM	No frame received timeout. The number of TOPS clock cycles to wait before generating an interrupt if no frame is received. The associated interrupt is disabled if this field is zero.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.11 TXNOFR Timeout Register (TXNF)

This register sets the time to wait before generating an interrupt to indicate that no frame has been transmitted (that is, no attempts for transmission, both Tx queues remained empty).

Address offset: 0x004C
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TXNOFRM	No frame transmitted timeout. The number of TOPS clock cycles to wait before generating an interrupt if no frame is transmitted. The associated interrupt is disabled if this field is zero.
19–10	TXNFM	QCU mask. Specifies the set of QCUs for which frame completions will cause a reset of the TXNOFRM timeout. For each bit position corresponding to a QCU: 0 = Ignore frames transmitted by this QCU. 1 = Watch this QCU for transmitted frames.
31–20	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.12 Rx Frame Gap Timeout (RFGTO)

This register sets the delay between received frames after which a RxDESC interrupt will be generated.

Address offset: 0x0050
Access: Read/Write
Cold reset: 0x0000_001F
Warm reset: 0x0000_001F

Bit	Bit Name	Description
9–0		Rx timeout count limit.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.13 Rx Frame Count Limit (RFCNT)

This register sets the limit on the number of received frames before generating a RXDESC. Refer to "Primary Interrupt Status Register (ISR_P)" on page 103.

Address offset: 0x0054
Access: Read/Write
Cold reset: 0x0000_001F
Warm reset: 0x0000_001F

Bit	Bit Name	Description
4–0	RFCNT	Rx frame count limit. The AR5212 maintains an internal counter and increments it each time that it receives a new frame. When the internal counter's value is equal to RFCNT plus 1, the AR5212 sets the CR[RXDESC_INT] bit and signals an interrupt. The associated interrupt is disabled if the value of these bits equals 0x1F (decimal 31).
31–5	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.14 MAC-specific Miscellaneous Status/ Control Register (MACMISC)

This register control the internal debugging features that are not needed under normal operation.

Address offset: 0x0058
Access: Read/Write
Cold reset:
Warm reset:

Bit	Bit Name	Description
4–0	RES	Reserved. Must be written with zero. Can contain any value on read.
8–5		DMA observation bus mux select.
11–9		MISC observation bus mux select.
14–12		MAC observation bus bits (8–0) mux select: 0 = Source DMA observation bus (8:0). 1 = Source DMA observation bus (17:9). 2 = Source DMA observation bus (8:0). 3 = Source DMA observation bus (17:9). 4 = Source DMA observation bus (8:0). 5 = Source DMA observation bus (17:9). 6 = Source DMA observation bus (8:0). 7 = Source DMA observation bus (17:9).

Bit	Bit Name	Description
17–15		MAC observation bus bits (17–9) mux select: 0 = Source DMA observation bus (8:0). 1 = Source DMA observation bus (17:9). 2 = Source DMA observation bus (8:0). 3 = Source DMA observation bus (17:9). 4 = Source DMA observation bus (8:0). 5 = Source DMA observation bus (17:9). 6 = Source DMA observation bus (8:0). 7 = Source DMA observation bus (17:9).
31–18	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.15 Sleep Performance Counter Register (SPC_0)

This register sets the total number of cycles, in units of 256 sleep clocks, that the chip's core clock was running.

Address offset: 0x005C
Access: Read and clear
Cold reset: 0x0000_001F
Warm reset: 0x0000_001F

Bit	Bit Name	Description
23–0		The total number of cycles, in units of 256 sleep clocks, that the chip's core clock was running.
31–24	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.16 Sleep Performance Counter Register (SPC_1)

This register sets the total number of cycles, in units of 256 sleep clocks, that the chip's core clock was stopped.

Address offset: 0x0060
Access: Read and clear
Cold reset: 0x0000_001F
Warm reset: 0x0000_001F

Bit	Bit Name	Description
23–0		The total number of cycles, in units of 256 sleep clocks, that the chip's core clock was stopped.
31–24	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.17 Primary Interrupt Status Register (ISR_P)

Refer to "Host Interface Unit Interrupts" on page 34 for details on accessing primary ISR bits.

Address offset: 0x0080
Access: Read/Write (one-to-clear)
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
0	RXOK_PI	Frame was received with no errors. 0 = No interrupt. 1 = Interrupt pending.
1	RXDESC_PI	Frame was received and InterReq in the descriptor was set. 0 = No interrupt. 1 = Interrupt pending.
2	RXERR_PI	Frame was received with errors. 0 = No interrupt. 1 = Interrupt pending.
3	RXNOFRM_PI	No frame received for RXNOFRM timeout clocks. 0 = No interrupt. 1 = Interrupt pending.
4	RXEOL_PI	Rx descriptor fetch logic has no more Rx descriptors available. 0 = No interrupt. 1 = Interrupt pending.
5	RXORN_PI	RxFIFO overrun. 0 = No interrupt. 1 = Interrupt pending.
6	TXOK_PI	Logical OR of all TXOK bits in secondary ISR_0. Indicates that at least one frame was completed with no errors. 0 = No interrupt. 1 = Interrupt pending.
7	TXDESC_PI	Logical OR of all TXDESC bits in secondary ISR_0. Indicate a Tx frame completed with the InterReq bit set in the Tx descriptor. 0 = No interrupt. 1 = Interrupt pending.
8	TXERR_PI	Logical OR of all TXERR bits in secondary ISR_1. Indicates that at least one frame was completed with an error. 0 = No interrupt. 1 = Interrupt pending.
9	TXNOFRM_PI	No frames transmitted for TXNOFRM timeout clocks. There is just one TXNOFRM bit for all QCUs. Refer to "TXNOFR Timeout Register (TXNF)" on page 100 for details. 0 = No interrupt.t 1 = Interrupt pending.

Bit	Bit Name	Description
10	TXEOL_PI	Logical OR of all TXEOL bits in secondary ISR_1. Indicates that at least one Tx descriptor fetch state machine has no more Tx descriptors available. 0 = No interrupt. 1 = Interrupt pending.
11	TXURN_PI	Logical OR of all TXURN bits in secondary ISR_2. Indicates that the PCU reported a TxFIFO underrun for at least one QCU's frame. 0 = No interrupt. 1 = Interrupt pending.
12	MIB_PI	One of the MIB registers has reached its threshold. 0 = No interrupt. 1 = Interrupt pending.
13	SWI_PI	Software interrupt signalled. Refer to "Command Register (CR)" on page 93. 0 = No interrupt. 1 = Interrupt pending.
14	RXPHY_PI	The PHY signalled an error on a received frame. 0 = No interrupt. 1 = Interrupt pending.
15	RXKCM_PI	Key cache miss. A frame was received with the key cache miss receive status bit set. 0 = No interrupt. 1 = Interrupt pending.
16	SWBA_PI	PCU has signalled a software beacon alert. 0 = No interrupt. 1 = Interrupt pending.
17	BRSSI_PI	The RSSI of a received beacon has fallen below a programmable threshold. 0 = No interrupt. 1 = Interrupt pending.
18	BMISS_PI	A beacon has not been received during a programmable threshold. 0 = No interrupt. 1 = Interrupt pending.
19	HIUERR_PI	HIU block has encountered an error. The HIU is the logical OR of all SSERR, DPERR, and MCABT bits in secondary ISR_2. 0 = No interrupt. 1 = Interrupt pending.
20	BNR_PI	Beacon not ready. Indicates that the QCU marked as being used for beacons (refer to "Miscellaneous QCU Control Registers (Q_MISC)" on page 123) received a DMA beacon alert when the queue contained no frames. 0 = No interrupt. 1 = Interrupt pending.
21	RXCHIRP	Indicates that the PHY reported a chirp was received.
22	RES	Reserved. Must be written with zero. Can contain any value on read.
23	BCNMISC_PI	Miscellaneous beacon-related interrupts. This bit is the logical OR of the TIM, CABEND, DTIMSYNC, SCNTO, CABTO, and DTIM bits in ISR_2 register.

Bit	Bit Name	Description
24	GPIO_PI	A programmable GPIO pin was asserted. Interrupt generated by GPIO logic (refer to "GPIO Control Register (GPIOCR)" on page 144) 0 = No interrupt. 1 = Interrupt pending.
25	QCBROVF_PI	Logical OR of all QCBROVF bits in secondary ISR_3. Indicates that at least one QCU's CBR expired counter has reached the value of the QCU's CBROVFL parameter. 0 = No interrupt. 1 = Interrupt pending.
26	QCBRURN_PI	Logical OR of all QCBRURN bits in secondary ISR_3. Indicates that at least one QCU's frame scheduling trigger event occurred when no frames were present on the queue. 0 = No interrupt. 1 = Interrupt pending.
27	QTRIG_PI	Logical OR of all QTRIG bits in secondary ISR_4. Indicates that at least one QCU's frame scheduling trigger event has occurred. 0 = No interrupt. 1 = Interrupt pending.
31-28	RES	Reserved. Must be written with zero. On read, can contain any value.

9.2.18 Secondary Interrupt Status Register 0 (ISR_S0)

Address offset: 0x0084
Access: Read/Write (one-to-clear)
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TXOK_SI	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = A frame was transmitted from this QCU with errors.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
25–16	TXDESC_SI	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = A frame was transmitted from this QCU and InterReq in the descriptor was set.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.19 Secondary Interrupt Status Register 1 (ISR_S1)

Address offset: 0x0088
Access: Read/Write (one-to-clear)
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TXERR_SI	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = A frame was transmitted from this QCU with errors.
15–10	RES	Reserved. Must be written with zero. On read, can contain any value.
25–16	TXEOL_SI	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = A frame was transmitted from this QCU with a null LinkPtr in the Tx descriptor.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.20 Secondary Interrupt Status Register 2 (ISR_S2)

Address offset: 0x008C
Access: Read/Write (one-to-clear)
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TXURN_SI	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = This QCU underflowed while transmitting.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
16	MCABT_SI	Set if the PCI bus signals a master cycle abort. During a MAC-initiated master read or write cycle, the PCI/CardBus Interface received either a Target Abort on the PCI bus because the target explicitly aborted the transaction, or the PCI/CardBus Interface received a Master Abort on the PCI bus because no target ever responded to the transaction. 0 = No interrupt. 1 = PCI master cycle abort.
17	SSERR_SI	Signalled system error. Set if a parity error is detected on a PCI address cycle. 0 = No interrupt. 1 = PCI system error.
18	DPERR_SI	Detected parity error. Set if a parity error is detected on a PCI data cycle. 0 = No interrupt. 1 = PCI data parity error.
23–19	RES	Reserved. Must be written with zero. Can contain any value on read.
24	TIM	A beacon was received with the local station’s bit set in the TIM element.
25	CABEND	End of CAB traffic. A CAB frame was received with the ‘more data’ bit clear in the frame control field.

Bit	Bit Name	Description
26	DTIMSYNC	DTIM synchronization lost. A beacon was received that was expected to be a DTIM but was not, or a beacon was received that was not expected to be a DTIM but was.
27	BCNTO	Beacon timeout. TBTT occurred and the station began waiting to receive a beacon, but no beacon was received before the PCU's beacon timeout expired.
28	CABTO	CAB timeout. A beacon was received that indicated that the station should expect to receive CAB traffic. However, the PCU's CAB timeout expired either because the station received no CAB traffic, or because the station received some CAB traffic but never received a CAB frame with the 'more data' bit in the frame control field (which would indicate the final CAB frame).
29	DTIM	A beacon was received with the DTIM bit set and a DTIM count value of zero.
31–30	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.21 Secondary Interrupt Status Register 3 (ISR_S3)

Address offset: 0x0090
Access: Read/Write (one-to-clear)
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	QCBROVF_SI	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = CBR expired counter for this QCU reached threshold.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
25–16	QCBRURN_SI	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = This QCU received a trigger but there were no frames.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.22 Secondary Interrupt Status Register 4 (ISR_S4)

Address offset: 0x0094
Access: Read/Write (one-to-clear)
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	QTRIG_SI	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = This QCU received an enabled trigger event.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.23 Primary Interrupt Mask Register (IMR_P)

Address offset: 0x00A0
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
0	RXOK_PM	Interrupt mask for RXOK_PI. 0 = Disable interrupt. 1 = Enable interrupt.
1	RXDESC_PM	Interrupt mask for RXDESC_PI. 0 = Disable interrupt. 1 = Enable interrupt.
2	RXERR_PM	Interrupt mask for RXERR_PI. 0 = Disable interrupt. 1 = Enable interrupt.
3	RXNOFR_PM	Interrupt mask for RXNOFR_PI. 0 = Disable interrupt. 1 = Enable interrupt.
4	RXEOL_PM	Interrupt mask for RXEOL_PI. 0 = Disable interrupt. 1 = Enable interrupt.
5	RXORN_PM	Interrupt mask for RXORN_PI. 0 = Disable interrupt. 1 = Enable interrupt.
6	TXOK_PI	Interrupt mask for TXOK_PI. 0 = Disable interrupt. 1 = Enable interrupt.
7	TXDESC_PI	Interrupt mask for TXDESC_PI. 0 = Disable interrupt. 1 = Enable interrupt.
8	TXERR_PI	Interrupt mask for TXERR_PI. 0 = Disable interrupt. 1 = Enable interrupt.
9	TXNOFR_PI	Interrupt mask for TXNOFR_PI. 0 = Disable interrupt. 1 = Enable interrupt.
10	TXEOL_PI	Interrupt mask for TXEOL_PI. 0 = Disable interrupt. 1 = Enable interrupt.
11	TXURN_PI	Interrupt mask for TXURN_PI. 0 = Disable interrupt. 1 = Enable interrupt.
12	MIB_PI	Interrupt mask for MIB_PI. 0 = Disable interrupt. 1 = Enable interrupt.

Bit	Bit Name	Description
13	SWI_PI	Interrupt mask for SWI_PI. 0 = Disable interrupt. 1 = Enable interrupt.
14	RXPHY_PI	Interrupt mask for RXPHY_PI. 0 = No interrupt. 1 = Interrupt pending.
15	RXKCM_PI	Interrupt mask for RXKCM_PI. 0 = Disable interrupt. 1 = Enable interrupt.
16	SWBA_PI	Interrupt mask for SWBA_PI. 0 = Disable interrupt. 1 = Enable interrupt.
17	BRSSI_PI	Interrupt mask for BRSSI_PI. 0 = Disable interrupt. 1 = Enable interrupt.
18	BMISS_PI	Interrupt mask for BMISS_PI. 0 = Disable interrupt. 1 = Enable interrupt.
19	HIUERR_PI	Interrupt mask for HIUERR_PI. 0 = Disable interrupt. 1 = Enable interrupt.
20	BNR_PI	Interrupt mask for BNR_PI 0 = Disable interrupt. 1 = Enable interrupt.
21	RXCHIRP_PI	RXCHIRP interrupt enable.
22	RES	Reserved. Must be written with zero. Can contain any value on read.
23	BCNMISC_PI	Beacon miscellaneous.
24	GPIO_PI	Interrupt mask for GPIO_PI. 0 = Disable interrupt. 1 = Enable interrupt.
25	QCBROVF_PI	Interrupt mask for QCBROVF_PI. 0 = Disable interrupt. 1 = Enable interrupt.
26	QCBRURN_PI	Interrupt mask for QCBRURN_PI. 0 = Disable interrupt. 1 = Enable interrupt.
27	QTRIG_PI	Interrupt mask for QTRIG_PI. 0 = Disable interrupt. 1 = Enable interrupt.
31-28	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.24 Secondary Interrupt Mask Register 0 (IMR_S0)

Address offset: 0x00A4
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TXOK_SM	For each bit position corresponding to a QCU: 0 = Disable TXOK interrupt for this QCU. 1 = Enable TXOK interrupt for this QCU.
15–10	RES	Reserved. Must be written with zero. On read, can contain any value.
25–16	TXDESC_SM	For each bit position corresponding to a QCU: 0 = Disable TXDESC interrupt for this QCU. 1 = Enable TXDESC interrupt for this QCU.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.25 Secondary Interrupt Mask Register 1 (IMR_S1)

Address offset: 0x00A8
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TXERR_SM	For each bit position corresponding to a QCU: 0 = Disable TXERR interrupt for this QCU. 1 = Enable TXERR interrupt for this QCU.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
25–16	TXEOL_SM	For each bit position corresponding to a QCU: 0 = Disable TXEOL interrupt for this QCU. 1 = Enable TXEOL interrupt for this QCU.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.26 Secondary Interrupt Mask Register 2 (IMR_S2)

Address offset: 0x00AC
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TXURN_SM	For each bit position corresponding to a QCU: 0 = Disable TXURN interrupt for this QCU. 1 = Enable TXURN interrupt for this QCU.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
16	MCABT_SM	Set if the PCI bus signals a master cycle abort. 0 = Disable MCABT interrupt for this QCU. 1 = Enable MCABT interrupt for this QCU.
17	SERR_SM	Set if a parity error is detected on a PCI address cycle. 0 = Disable SERR interrupt for this QCU. 1 = Enable SERR interrupt for this QCU.
18	DPERR_SM	Set if a parity error is detected on a PCI data cycle. 0 = Disable DPERR interrupt for this QCU. 1 = Enable DPERR interrupt for this QCU.
23–19	RES	Reserved. Must be written with zero. Can contain any value on read.
24	TIM_SM	TIM interrupt enable.
25	CABEND_SM	CABEND interrupt enable.
26	DTIMSYNC_SM	DTIMSYNC interrupt enable.
27	BCNTO_SM	BCNT interrupt enable.
28	CABTO_SM	CABTO interrupt enable.
31–29	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.27 Secondary Interrupt Mask Register 3 (IMR_S3)

Address offset: 0x00B0
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	QCBROVF_SM	For each bit position corresponding to a QCU: 0 = Disable QCBROVF interrupt for this QCU. 1 = Enable QCBROVF interrupt for this QCU.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
25–16	QCBRURN_SM	For each bit position corresponding to a QCU: 0 = Disable QCBRURN interrupt for this QCU. 1 = Enable QCBRURN interrupt for this QCU.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.28 Secondary Interrupt Mask Register 4 (IMR_S4)

Address offset: 0x00B4
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	QTRIG_SM	For each bit position corresponding to a QCU: 0 = Disable QTRIG interrupt for this QCU. 1 = Enable QTRIG interrupt for this QCU.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.29 Primary RAC Interrupt Status Register (ISR_P_RAC)

This register contains the (RAC) interrupt status.

A read from this location atomically performs the following:

- Copies all secondary ISRs into the corresponding secondary ISR shadow registers ("Secondary Interrupt Status Register 0 (ISR_S0)" on page 105 is copied to "Secondary Shadow Interrupt Status Register 0 (ISR_S0_S)" on page 115, and so forth).
- Clears all bits of the "Primary Interrupt Status Register (ISR_P)" on page 103, as well as all bits of all secondary ISRs ("Secondary Interrupt Status Register 0 (ISR_S0)" on page 105 through "Secondary Interrupt Status Register 4 (ISR_S4)" on page 107.
- Returns the contents of the ISR_P.

Address offset: 0x00C0
Access: Read-and-clear/No Write access
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
0	RXOK_RAC	Frame was received with no errors. 0 = No interrupt. 1 = Interrupt pending.
1	RXDESC_RAC	Frame was received and InterReq in the descriptor was set. 0 = No interrupt. 1 = Interrupt pending.
2	RXERR_RAC	Frame was received with errors. 0 = No interrupt. 1 = Interrupt pending.
3	RXNOFR_RAC	No frame received for RXNOFR timeout clocks. 0 = No interrupt. 1 = Interrupt pending.
4	RXEOL_RAC	Rx descriptor fetch logic has no more Rx descriptors available. 0 = No interrupt 1 = Interrupt pending.
5	RXORN_RAC	RxFIFO overrun. 0 = No interrupt. 1 = Interrupt pending.
6	TXOK_RAC	Logical OR of all TXOK bits in ISR_S0. 0 = No interrupt. 1 = Interrupt pending.
7	TXDESC_RAC	Logical OR of all TXDESC bits in ISR_S0. 0 = No interrupt. 1 = Interrupt pending.
8	TXERR_PI	Logical OR of all TXERR bits in ISR_S1. 0 = No interrupt. 1 = Interrupt pending.
9	TXNOFR_RAC	No frames transmitted for TXNOFR timeout clocks. 0 = No interrupt. 1 = Interrupt pending.
10	TXEOL_RAC	Logical OR of all TXEOL bits in ISR_S1. 0 = No interrupt. 1 = Interrupt pending.

Bit	Bit Name	Description
11	TXURN_RAC	Logical OR of all TXURN bits in ISR_S2. 0 = No interrupt. 1 = Interrupt pending.
12	MIB_RAC	One of the MIB registers has reached its threshold. 0 = No interrupt. 1 = Interrupt pending.
13	SWI_RAC	Software interrupt signalled. 0 = No interrupt. 1 = Interrupt pending.
14	RXPHY_RAC	PHY signaled an error on a received frame. 0 = No interrupt. 1 = Interrupt pending.
15	RXKCM_RAC	A frame was received that did not match in the key cache. 0 = No interrupt 1 = Interrupt pending
16	SWBA_RAC	PCU has signaled a software beacon alert. 0 = No interrupt. 1 = Interrupt pending.
17	BRSSI_RAC	The RSSI of a received beacon has fallen below a programmable threshold. 0 = No interrupt. 1 = Interrupt pending.
18	BMISS_RAC	A beacon has not been received during a programmable threshold. 0 = Disable interrupt. 1 = Enable interrupt.
19	HIUERR_RAC	Logical OR of the SSERR, DPERR, and MCABT bits in ISR_S2. 0 = No interrupt. 1 = Interrupt pending.
20	BNR_RAC	Beacon not ready. 0 = No interrupt. 1 = Interrupt pending.
21	RXCHIRP_RAC	Indicates that the PHY reported a chirp was received.
22	RES	Reserved. Must be written with zero. On read, can contain any value.
23	TIM_RAC	A beacon was received with this STA’s TIM bit set. 0 = No interrupt. 1 = Interrupt pending.
24	GPIO_RAC	A programmable GPIO pin was asserted. 0 = No interrupt. 1 = Interrupt pending.
25	QCBROVF_RAC	Logical OR of all QCBROVF bits in ISR_S3. 0 = No interrupt. 1 = Interrupt pending.
26	QCBRURN_RAC	Logical OR of all QCBRURN bits in ISR_S3. 0 = No interrupt. 1 = Interrupt pending.

Bit	Bit Name	Description
27	QTRIG_RAC	Logical OR of all QTRIG bits in ISR_S4. 0 = No interrupt. 1 = Interrupt pending.
31-28	RES	Reserved. Must be written with zero. On read, can contain any value.

9.2.30 Secondary Shadow Interrupt Status Register 0 (ISR_S0_S)

Address offset: 0x00C4
Access: Read Only
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
9–0	TXOK_SS	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = A frame was transmitted from this QCU with no errors.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
25–16	TXDESC_SS	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = A frame was transmitted from this QCU and InterReq in the Tx descriptor was set.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.31 Secondary Shadow Interrupt Status Register 1 (ISR_S1_S)

Address offset: 0x00C8
Access: Read Only
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
9–0	TXERR_SS	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = A frame was transmitted from this QCU with no errors.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
25–16	TXEOL_SS	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = A frame was transmitted from this QCU with a null LinkPtr in the Tx descriptor.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.32 Secondary Shadow Interrupt Status Register 2 (ISR_S2_S)

Address offset: 0x00CC
Access: Read Only
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
9–0	TXURN_SM	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = This QCU underflowed while transmitting.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
16	MCABT_SM	Signal PCI master cycle abort. 0 = No interrupt. 1 = PCI master cycle abort.
17	SSERR_SM	Signal PCI system error. 0 = No interrupt. 1 = PCI system error.
18	DPERR_SM	Signal PCI data parity error. 0 = No interrupt. 1 = PCI data parity error.
31–19	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.33 Secondary Shadow Interrupt Status Register 3 (ISR_S3_S)

Address offset: 0x00D0
Access: Read Only
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
9–0	QCBROVF_SS	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = CBR expired counter for this QCU reached threshold.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
25–16	QCBRURN_SS	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = This QCU received a trigger but there were no frames.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.2.34 Secondary Shadow Interrupt Status Register 4 (ISR_S4_S)

Address offset: 0x00D4
Access: Read Only
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
9–0	QTRIG_SS	For each bit position corresponding to a QCU: 0 = No interrupt. 1 = This QCU received an enabled trigger event.
31–10	RES	Reserved. Must be written with zero. On read, can contain any value.

9.3 QCU Registers

Table 9-3 summarizes the QCU registers and their descriptions provided by the AR5212. These registers control the QCU operation. Some registers are global and affect all QCUs, while others affect individual QCUs. The QCUs are numbered 0 to 9 and are identical. QCU registers are located at offset addresses 0x0800 to 0x0A40.

Table 9-3. **QCU Register Summary**

Offset	Name	Description	Page
0x0800	Q_TXDP_0	Tx descriptor pointer for QCU 0.	page 120
0x0804	Q_TXDP_1	Tx descriptor pointer for QCU 1.	page 120
0x0808	Q_TXDP_2	Tx descriptor pointer for QCU 2.	page 120
0x080C	Q_TXDP_3	Tx descriptor pointer for QCU 3.	page 120
0x0810	Q_TXDP_4	Tx descriptor pointer for QCU 4.	page 120
0x0814	Q_TXDP_5	Tx descriptor pointer for QCU 5.	page 120
0x0818	Q_TXDP_6	Tx descriptor pointer for QCU 6.	page 120
0x081C	Q_TXDP_7	Tx descriptor pointer for QCU 7.	page 120
0x0820	Q_TXDP_8	Tx descriptor pointer for QCU 8.	page 120
0x0824	Q_TXDP_9	Tx descriptor pointer for QCU 9.	page 120
0x0840	Q_TXE	QCU Tx queue enable. Enables the TxFIFO to process descriptors and send frames.	page 121
0x0880	Q_TXD	QCU Tx queue disable. Enables the TxFIFO to gracefully stop sending Tx frames.	page 121
0x08C0	Q_CBRCFG_0	CBR configuration for QCU 0. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121
0x08C4	Q_CBRCFG_1	CBR configuration for QCU 1. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121
0x08C8	Q_CBRCFG_2	CBR configuration for QCU 2. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121
0x08CC	Q_CBRCFG_3	CBR configuration for QCU 3. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121
0x08D0	Q_CBRCFG_4	CBR configuration for QCU 4. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121
0x08D4	Q_CBRCFG_5	CBR configuration for QCU 5. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121
0x08D8	Q_CBRCFG_6	CBR configuration for QCU 6. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121

Table 9-3. **QCU Register Summary (continued)**

Offset	Name	Description	Page
0x08DC	Q_CBRCFG_7	CBR configuration for QCU 7. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121
0x08E0	Q_CBRCFG_8	CBR configuration for QCU 8. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121
0x08E4	Q_CBRCFG_9	CBR configuration for QCU 9. Sets the period between QCU CBR (constant bit rate) triggers and controls when a CBR overflow interrupt is generated.	page 121
0x0900	Q_RDYTIMECFG_0	Readytime configuration for QCU 0. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x0904	Q_RDYTIMECFG_1	Readytime configuration for QCU 1. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x0908	Q_RDYTIMECFG_2	Readytime configuration for QCU 2. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x090C	Q_RDYTIMECFG_3	Readytime configuration for QCU 3. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x0910	Q_RDYTIMECFG_4	Readytime configuration for QCU 4. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x0914	Q_RDYTIMECFG_5	Readytime configuration for QCU 5. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x0918	Q_RDYTIMECFG_6	Readytime configuration for QCU 6. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x091C	Q_RDYTIMECFG_7	Readytime configuration for QCU 7. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x0920	Q_RDYTIMECFG_8	Readytime configuration for QCU 8. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x0924	Q_RDYTIMECFG_9	Readytime configuration for QCU 9. Controls how long a QCU is allowed to schedule Tx frames.	page 122
0x0940	Q_ONESHOTARM_SC	Set oneshot arm. Allows a single frame to be sent when a trigger event occurs. ONESHOTSET arms the oneshot function.	page 122
0x0980	Q_ONESHOTARM_CC	Clear oneshot arm. Allows a single frame to be sent when a trigger event occurs. ONESHOTCLR disarms the oneshot function.	page 122
0x09C0	Q_MISC_0	Miscellaneous control for QCU 0.	page 123
0x09C4	Q_MISC_1	Miscellaneous control for QCU 1.	page 123
0x09C8	Q_MISC_2	Miscellaneous control for QCU 2.	page 123
0x09CC	Q_MISC_3	Miscellaneous control for QCU 3.	page 123
0x09D0	Q_MISC_4	Miscellaneous control for QCU 4.	page 123
0x09D4	Q_MISC_5	Miscellaneous control for QCU 5.	page 123
0x09D8	Q_MISC_6	Miscellaneous control for QCU 6.	page 123
0x09DC	Q_MISC_7	Miscellaneous control for QCU 7.	page 123
0x09E0	Q_MISC_8	Miscellaneous control for QCU 8.	page 123
0x09E4	Q_MISC_9	Miscellaneous control for QCU 9.	page 123

Table 9-3. **QCU Register Summary (continued)**

Offset	Name	Description	Page
0x0A00	Q_STS_0	Miscellaneous status for QCU 0.	page 125
0x0A04	Q_STS_1	Miscellaneous status for QCU 1.	page 125
0x0A08	Q_STS_2	Miscellaneous status for QCU 2.	page 125
0x0A0C	Q_STS_3	Miscellaneous status for QCU 3.	page 125
0x0A10	Q_STS_4	Miscellaneous status for QCU 4.	page 125
0x0A14	Q_STS_5	Miscellaneous status for QCU 5.	page 125
0x0A18	Q_STS_6	Miscellaneous status for QCU 6.	page 125
0x0A1C	Q_STS_7	Miscellaneous status for QCU 7.	page 125
0x0A20	Q_STS_8	Miscellaneous status for QCU 8.	page 125
0x0A24	Q_STS_9	Miscellaneous status for QCU 9.	page 125
0x0A40	Q_RDYTIMESHDN	ReadyTimeShutdown. Set whenever readytime expires while frames are pending in the QCU. This indicates whether all the available frames in a QCU were transmitted before the readytime expired.	page 125

9.3.1 QCU Tx Descriptor Pointer Registers (Q_TXDP)

Address offset: (0x0800 + (QCU*4))
QCU Range: 0–9 (see page 118 for range)
Access: Read/Write
Cold reset: Undefined
Warm reset: Unaffected

Bit	Bit Name	Description
1–0	RES	Reserved. Must be written with zero. Can contain any value on read.
31–2	TXDP	Transmit FIFO pointer. Contains the address of the current Tx descriptor for this QCU. The effect of writing TXDP depends on the state of the TxFIFO: ■ If TxFIFO is busy (TxE is one), writes are ignored. ■ If TxFIFO is idle (after reset, after TxD is cleared, or at the end of a descriptor chain), writing TXDP updates the location where the next Tx descriptor will be read when TxE is set.

9.3.2 QCU Transmit Queue Enable Register (Q_TXE)

Address offset: 0x0840
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TxE	For each bit position corresponding to a QCU: 0 = Ignore. 1 = Enable QCU.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.3.3 QCU Transmit Queue Disable Register (Q_TXD)

Address offset: 0x0880
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TxD	For each bit position corresponding to a QCU: 0 = Ignore. 1 = Disable QCU.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.3.4 QCU CBR Registers (Q_CBRCFG)

Address offset: (0x08C0 + (QCU*4))
QCU Range: 0–9 (see page 118 for range)
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
23–0	CBRINT	CBR period in µs.
31–24	CBROVFL	CBR overflow threshold read/write. Determines the value of the CBR expired counter at which a CBROVF interrupt will be generated.

9.3.5 QCU Readytime Configuration Registers (Q_RDYTIMECFG)

Address offset: (0x0900 + (QCU*4))
QCU Range: 0–9 (see page 119 for range)
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
23–0	RDYTIMEINT	Readytime interval in µs. Controls how long a QCU is allowed to be ready, that is, schedule Tx frames.
24	RDYTIMEEN	0 = Disable readytime. 1 = Enable readytime.
31–25	RES	Reserved. Must be written with zero. Can contain any value on read.

9.3.6 Set Oneshot Arm Register (Q_ONESHOTARM_SC)

Address offset: 0x0940
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	ONESHOTSET	The oneshot function allows a single frame to be sent when a trigger event occurs. For each bit position corresponding to a QCU: 0 = Ignore. 1 = Set arm.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.3.7 Clear Oneshot Arm Register (Q_ONESHOTARM_CC)

Address offset: 0x0980
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	ONESHOTCLR	The oneshot function allows a single frame to be sent when a trigger event occurs. For each bit position corresponding to a QCU: 0 = Ignore. 1 = Clear arm.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.3.8 Miscellaneous QCU Control Registers (Q_MISC)

Address offset: (0x09C0 + (QCU*4))
QCU Range: 0–9 (see page 119 for range)
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
3–0	FRSHED	Selects the frame scheduling policy: 0 = ASAP. The QCU is enabled continuously. 1 = CBR. The QCU will be enabled under control of the settings in "QCU CBR Registers (Q_CBRCFG)" (whenever the CBREXPCNT field of "Miscellaneous QCU Status Registers (Q_STS)" is non-zero). 2 = DBA–gated. The QCU will be enabled at each occurrence of a DMA beacon alert, which is controlled by the DBA register. 3 = TIM–gated. The QCU will be enabled in STA mode and AdHoc mode when: ■ In STA mode (ADHOC in CFG is clear), the PCU indicates that a beacon frame has been received with the local STA's bit set in the TIM element. ■ In AdHoc mode (ad hoc in CFG is set), the PCU indicates that an ATIM frame has been received. 4 = Beacon–sent–gated. The QCU will be enabled when the DCU that is marked as being used for BEACON (bit 16) set in "Miscellaneous Registers (D_MISC)" indicates that it has sent the beacon frame on the air.
4	ONESHOTEN	OneShot enable. Enables a mode where a QCU is made ready only if oneshot is armed when the trigger occurs. 0 = Disable OneShot function. 1 = Enable OneShot function. **NOTE:** OneShot must not be enabled when the QCU is set to an ASAP frame scheduling policy.
5	CBREXPCTL	Enables a mode where CBREXPCNT increment (of "Miscellaneous QCU Status Registers (Q_STS)" on page 125) is disabled when the queue is disabled (TxE is clear). 0 = Enable the CBR expired counter increment each time the frame scheduling trigger occurs, regardless of whether the queue contains frames 1 = Disable the CBR expired counter increment only when both the frame scheduling trigger occurs and the queue is valid (the queue is valid whenever TxE is asserted)
6	CBREXPBCTL	Enables a mode where CBREXPCNT increment (of "Miscellaneous QCU Status Registers (Q_STS)" on page 125) is disabled when the QCU which has BEACONEN set is disabled (TxE is clear). 0 = Enable the CBR expired counter increment each time the frame scheduling trigger occurs, regardless of whether the beacon queue contains frames 1 = Disable the CBR expired counter increment only when both the frame scheduling trigger occurs and the beacon queue is valid (the beacon queue is valid whenever its TxE is asserted)

Bit	Bit Name	Description
7	BEACONEN	Beacon use indication. Indicates whether the QCU is being used for beacons. 0 = QCU is being used for non-beacon frames only. 1 = QCU is being used for beacon frames (and possibly for non-beacon frames).
8	CBRTHREN	CBR expired counter limit enable. Enables a mode where the maximum CBREXPCNT is set by CBROVFL in Q_CBRCFG. 0 = Maximum CBR expired counter value is 255, but a CBROVFL interrupt will be generated when the counter reaches the value set in the CBR overflow threshold field of the Q_CBRCFG register. 1 = Maximum CBR expired counter is limited to the value of the CBR overflow threshold field of the Q_CBRCFG register. Note that in addition to limiting the maximum CBR expired counter to this value, a CBROVFL interrupt also will be generated when the CBR expired counter reaches the CBR overflow threshold.
9	TXECTL	ReadyTime expiration and VEOL handling policy. Enables a mode where QCU is disabled (TxE is clear) when VEOL is set in the Tx descriptor or readytime expires (controlled by Q_RDYTIMECFG). 0 = On expiration of ReadyTime or on setting VEOL, the TxE bit is not cleared. Only reaching the physical end-of-queue, that is, a NULL LinkPtr, will clear TxE. 1 = The TxE bit is cleared on expiration of ReadyTime, on VEOL, and on reaching the physical end-of-queue.
10	CBRCTL	CBR expired counter force-clear control. Write-only (always reads as zero). Used to clear CBREXPCNT. 0 = No effect. 1 = Resets the CBR expired counter to zero.
11	DCUEARLYCTL	DCU frame early termination request control. Used to quickly flush the QCU. 0 = Never request early frame termination. Once a frame enters the DCU, it will remain active until its normal retry count has been reached or the frame succeeds. 1 = Allow this QCU to request early frame termination. When requested, the DCU will try to complete processing the frame more quickly than it normally would.
31–12	RES	Reserved. Must be written with zero. Can contain any value on read.

9.3.9 Miscellaneous QCU Status Registers (Q_STS)

Address offset: (0x0A00 + (QCU*4)
QCU Range: 0–9 (see page 120 for range)
Access: Read Only
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
1–0	FRPENDCNT	Pending frame count. Indicates the number of outstanding frames this QCU presently has pending in its associated DCU.
7–2	RES	Reserved. Must be written with zero. Can contain any value on read.
15–8	CBREXPCNT	Current value of CBR expired counter. Expired count. Indicates the number of outstanding CBR triggers.
31–16	RES	Reserved. Must be written with zero. Can contain any value on read.

9.3.10 Readytime Shutdown Register (Q_RDYTIMESHDN)

Address offset: 0x0A40
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description	
9–0	RDYTIMESHDN	Set whenever readytime expires while frames are pending in the QCU. For each bit position corresponding to a QCU: On Read: 0 = No frames pending when readytime expired. 1 = Frames pending when readytime expired.	 On Write: 0 = Ignore (no effect). 1 = Clear readytime shutdown status.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.	

9.4 DCU Registers

Table 9-4 summarizes the DCU registers and their descriptions provided by the AR5212. These registers configure and control the individual DCUs. The DCUs are numbered 0 to 9 and are identical except that when arbitrating for DCUs, DCU 0 has the lowest priority and DCU 9 has the highest priority. Each register has a base address that is then offset based on the DCU to be accessed. DCU registers are located at offset addresses 0x1000 to 0x147C.

Table 9-4. **DCU Register Summary**

Offset	Name	Description	Page
0x1000	D_QCUMASK_0	QCU mask for DCU 0. Selects which QCUs feeds this DCU.	page 128
0x1004	D_QCUMASK_1	QCU mask for DCU 1. Selects which QCUs feeds this DCU.	page 128
0x1008	D_QCUMASK_2	QCU mask for DCU 2. Selects which QCUs feeds this DCU.	page 128
0x100C	D_QCUMASK_3	QCU mask for DCU 3. Selects which QCUs feeds this DCU.	page 128
0x1010	D_QCUMASK_4	QCU mask for DCU 4. Selects which QCUs feeds this DCU.	page 128
0x1014	D_QCUMASK_5	QCU mask for DCU 5. Selects which QCUs feeds this DCU.	page 128
0x1018	D_QCUMASK_6	QCU mask for DCU 6. Selects which QCUs feeds this DCU.	page 128
0x101C	D_QCUMASK_7	QCU mask for DCU 7. Selects which QCUs feeds this DCU.	page 128
0x1020	D_QCUMASK_8	QCU mask for DCU 8. Selects which QCUs feeds this DCU.	page 128
0x1024	D_QCUMASK_9	QCU mask for DCU 9. Selects which QCUs feeds this DCU.	page 128
0x1040	D_LCL_IFS_0	Local IFS setting for DCU 0. Controls the various per-DCU interframe spacing parameters.	page 129
0x1044	D_LCL_IFS_1	Local IFS setting for DCU 1. Controls the various per-DCU interframe spacing parameters.	page 129
0x1048	D_LCL_IFS_2	Local IFS setting for DCU 2. Controls the various per-DCU interframe spacing parameters.	page 129
0x104C	D_LCL_IFS_3	Local IFS setting for DCU 3. Controls the various per-DCU interframe spacing parameters.	page 129
0x1050	D_LCL_IFS_4	Local IFS setting for DCU 4. Controls the various per-DCU interframe spacing parameters.	page 129
0x1054	D_LCL_IFS_5	Local IFS setting for DCU 5. Controls the various per-DCU interframe spacing parameters.	page 129
0x1058	D_LCL_IFS_6	Local IFS setting for DCU 6. Controls the various per-DCU interframe spacing parameters.	page 129
0x105C	D_LCL_IFS_7	Local IFS setting for DCU 7. Controls the various per-DCU interframe spacing parameters.	page 129
0x1060	D_LCL_IFS_8	Local IFS setting for DCU 8. Controls the various per-DCU interframe spacing parameters.	page 129
0x1064	D_LCL_IFS_9	Local IFS setting for DCU 9. Controls the various per-DCU interframe spacing parameters.	page 129

Table 9-4. **DCU Register Summary (continued)**

Offset	Name	Description	Page
0x1080	D_RETRY_LIMIT_0	Retry limit for DCU 0. Specifies the short and long retry limits for both frame and STA.	page 129
0x1084	D_RETRY_LIMIT_1	Retry limit for DCU 1. Specifies the short and long retry limits for both frame and STA.	page 129
0x1088	D_RETRY_LIMIT_2	Retry limit for DCU 2. Specifies the short and long retry limits for both frame and STA.	page 129
0x108C	D_RETRY_LIMIT_3	Retry limit for DCU 3. Specifies the short and long retry limits for both frame and STA.	page 129
0x1090	D_RETRY_LIMIT_4	Retry limit for DCU 4. Specifies the short and long retry limits for both frame and STA.	page 129
0x1094	D_RETRY_LIMIT_5	Retry limit for DCU 5. Specifies the short and long retry limits for both frame and STA.	page 129
0x1098	D_RETRY_LIMIT_6	Retry limit for DCU 6. Specifies the short and long retry limits for both frame and STA.	page 129
0x109C	D_RETRY_LIMIT_7	Retry limit for DCU 7. Specifies the short and long retry limits for both frame and STA.	page 129
0x10A0	D_RETRY_LIMIT_8	Retry limit for DCU 8. Specifies the short and long retry limits for both frame and STA.	page 129
0x10A4	D_RETRY_LIMIT_9	Retry limit for DCU 9. Specifies the short and long retry limits for both frame and STA.	page 129
0x10C0	D_CHNTIME_0	Channel time setting for DCU 0. Specifies channeltime duration in µs for the specified DCU.	page 129
0x10C4	D_CHNTIME_1	Channel time setting for DCU 1. Specifies channeltime duration in µs for the specified DCU.	page 129
0x10C8	D_CHNTIME_2	Channel time setting for DCU 2. Specifies channeltime duration in µs for the specified DCU.	page 129
0x10CC	D_CHNTIME_3	Channel time setting for DCU 3. Specifies channeltime duration in µs for the specified DCU.	page 129
0x10D0	D_CHNTIME_4	Channel time setting for DCU 4. Specifies channeltime duration in µs for the specified DCU.	page 129
0x10D4	D_CHNTIME_5	Channel time setting for DCU 5. Specifies channeltime duration in µs for the specified DCU.	page 129
0x10D8	D_CHNTIME_6	Channel time setting for DCU 6. Specifies channeltime duration in µs for the specified DCU.	page 129
0x10DC	D_CHNTIME_7	Channel time setting for DCU 7. Specifies channeltime duration in µs for the specified DCU.	page 129
0x10E0	D_CHNTIME_8	Channel time setting for DCU 8. Specifies channeltime duration in µs for the specified DCU.	page 129
0x10E4	D_CHNTIME_9	Channel time setting for DCU 9. Specifies channeltime duration in µs for the specified DCU.	page 129
0x1100	D_MISC_0	Various arbitration controls for DCU 0.	page 130
0x1104	D_MISC_1	Various arbitration controls for DCU 1.	page 130
0x1108	D_MISC_2	Various arbitration controls for DCU 2.	page 130
0x110C	D_MISC_3	Various arbitration controls for DCU 3.	page 130
0x1110	D_MISC_4	Various arbitration controls for DCU 4.	page 130

Table 9-4. **DCU Register Summary (continued)**

Offset	Name	Description	Page
0x1114	D_MISC_5	Various arbitration controls for DCU 5.	page 130
0x1118	D_MISC_6	Various arbitration controls for DCU 6.	page 130
0x111C	D_MISC_7	Various arbitration controls for DCU 7.	page 130
0x1120	D_MISC_8	Various arbitration controls for DCU 8.	page 130
0x1124	D_MISC_9	Various arbitration controls for DCU 9.	page 130
0x1140	D_SEQNUM	Frame sequence number for all DCUs. Specifies local sequence number.	page 132
0x1030	D_GBL_IFS_SIFS	SIFS settings. Fixed interval backoff following each valid packet. For 802.11a, SIFS is 16 µs.	page 132
0x1070	D_GBL_IFS_SLOT	DCU global slot interval. Unit of measure for defining interframe spacing. For 802.11a, a slot is 9 µs.	page 133
0x10B0	D_GBL_IFS_EIFS	EIFS setting. Fixed interval backoff following each error packet. For 802.11a, EIFS is 87 µs.	page 133
0x10F0	D_GBL_IFS_MISC	Miscellaneous IFS settings.	page 133
0x1230	D_FPCTL	Frame prefetch settings.	page 134
0x1270	D_TXPSE	Transmit pause control/status.	page 135
0x1038	D_TXBLK_CMD	Transmit filter command. Updates individual Tx filter bits.	page 135
0x1038	D_TXBLK_DATA	Transmit filter data. Specifies a 32-bit slice of the Tx filter.	page 136
0x143C	D_TXBLK_CLR	Clear Tx filter. Clears all 128 bits of the Tx filter for that DCU.	page 138
0x147C	D_TXBLK_SET	Set Tx filter. Sets all 128 bits of the Tx filter for that DCU.	page 138

9.4.1 Queue Mask Registers (D_QCUMASK)

To achieve lowest power consumption, software should set this register to 0x0 for all DCUs that are not in use. (The hardware detects that the QCU mask is all zero and shuts down certain logic in response, which helps save power.)

Address offset: (0x1000 + (DCU*4))
DCU Range: 0–9 (see page 126 for range)
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	QCUMASK	Selects which QCU(s) feed this DCU. For each bit position corresponding to a QCU: 0 = Disable QCU. 1 = Enable QCU.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.2 Local IFS Settings Registers (D_LCL_IFS)

Address offset: (0x1040 + (DCU*4))
DCU Range: 0–9 (see page 126 for range)
Access: Read/Write
Cold reset: 0x002F_FC04
Warm reset: 0x002F_FC04

Bit	Bit Name	Description
9–0	CW_MIN	Minimum contention window in slots. Must be equal to a power of 2, minus 1.
19–10	CW_MAX	Maximum contention window in slots. Must be equal to a power of 2, minus 1.
27–20	AIFS	AIFS value, in slots beyond SIFS. For example, a setting of 2 means AIFS is equal to DIFS.
31–28	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.3 Retry Limits Registers (D_RETRY_LIMIT)

Address offset: (0x1080 + (DCU*4))
DCU Range: 0–9 (see page 127 for range)
Access: Read/Write
Cold reset: 0x0002_0844
Warm reset: 0x0002_0844

Bit	Bit Name	Description
3–0	FSR	Frame short retry limit.
7–4	FLR	Frame long retry limit.
13–8	SSR	STA short retry limit.
19–14	SLR	STA long retry limit.
31–20	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.4 ChannelTime Settings Registers (D_CHNTIME)

Address offset: (0x10C0 + (DCU*4))
DCU Range: 0–9 (see page 127 for range)
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
19–0	CHNTIME	Channel time duration in µs.
20	CHNTIME_EN	Channel time enable. 0 = Disable ChannelTime function. 1 = Enable ChannelTime function.
31–21	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.5 Miscellaneous Registers (D_MISC)

Address offset: (0x1100 + (DCU*4))
DCU Range: 0–9 (see page 127 for range)
Access: Read/Write
Cold reset: 0x0000_1002
Warm reset: 0x0000_1002

Bit	Bit Name	Description
5–0	BO_THR	Backoff threshold setting. Determines the backoff count at which the DCU will initiate arbitration for access to the PCU and commit to sending the frame.
6		End of transmission series station RTS/data failure count reset policy. Note that this bit controls only whether the two station failure counts are reset when transitioning from one transmission series to the next *within* a single frame. The counts are reset per the 802.11 spec when the entire frame attempt terminates (either because the frame was sent successfully or because all transmission series failed). Resets to 0x0. 0 = Do not reset the station RTS failure count or the station data failure count at the end of each transmission series. 1 = Reset both the station RTS failure count and the station data failure count at the end of each transmission series.
7		End of transmission series CW reset policy. Note that this bit controls only whether the contention window is reset when transitioning from one transmission series to the next *within* a single frame. The CW is reset per the 802.11 spec when the entire frame attempt terminates (either because the frame was sent successfully or because all transmission series failed). Resets to 0x0. 0 = Reset the CW to CW_MIN at the end of each intraframe transmission series. 1 = Do not reset the CW at the end of each intraframe transmission series.
8		Fragment burst frame starvation handling policy. This bit controls the DCU operation when the DCU is in the middle of a fragment burst and finds that the QCU sourcing the fragments does not have the next fragment available. Resets to 0x0. 0 = The DCU terminates the fragment burst. Note that when this occurs, the remaining fragments (when the QCU eventually has them available) will be sent as a separate fragment burst with a different sequence number. 1 = The DCU waits for the QCU to have the next fragment available. While doing so, all other DCUs will be unable to transmit frames.
9		Fragment burst backoff policy. This bit controls whether the DCU performs a backoff after each transmission of a fragment (that is, a frame with the MoreFrag bit set in the frame control field). Resets to 0x0. 0 = The DCU handles fragment bursts normally -- no backoff is performed after a successful transmission, and the next fragment is sent at SIFS. 1 = Modified handling. The DCU performs a backoff after all fragments, even those transmitted successfully. In addition, after the backoff count reaches zero, the DCU then follows the normal channel access procedure and sends at AIFS rather than at SIFS. This setting is intended to ease the use of fragment bursts in XR mode; see bug 4454 for more details.
10	RES	Reserved. Must be written with zero. Can contain any value on read.
11		HCF poll enable.
12	BO_PF	Backoff persistence factor setting. 0 = Contention window remains constant. 1 = Contention window grows using a binary-exponential.

Bit	Bit Name	Description
13	RES	Reserved. Must be written with zero. Can contain any value on read.
15–14	VIRTCOL	Virtual collision handling policy. 0 = Default handling. A virtual collision is processed like a collision on the air except that the retry count for the frame is not incremented (perform only backoff). 1 = Ignore. Virtual collisions are ignored (DCU immediately re-arbitrates for access to the PCU without performing a backoff and without incrementing the retry count). 2 = Reserved. 3 = Reserved.
16	BEACON	Beacon use indication. Indicates whether the DCU is being used for beacons. 0 = This DCU is only used for non-beacon frames. 1 = This DCU is only used for beacons.
18–17	ARBLOCKCTL	DCU Arbiter lockout control. 0 = No lockout. Allow lower priority DCUs to arbitrate for access to the PCU concurrently with this DCU. 1 = Intra-frame lockout only. Prevent lower priority DCUs from arbitrating for access to the PCU (until the active frame completes) while the current DCU is either arbitrating for access to the PCU or performing an intra-frame backoff. 2 = Global lockout. Prevent lower priority DCUs from arbitrating for access to the PCU when: ■ At least one of the QCUs that feed into the current DCU has a frame ready. ■ The current DCU is actively processing a frame (i.e., is not idle). This includes arbitrating for access to the PCU, performing an intra-frame or post-frame backoff, passing frame data to the PCU, or waiting for the PCU to complete the frame.
19	ARBLOCKPRI	DCU Arbiter lockout ignore control. 0 = Obey ARBLOCKCTL lockouts from higher priority DCUs. 1 = Ignore ARBLOCKCTL lockouts from higher priority DCUs, that is, allow the current DCU to arbitrate for access to the PCU even if one or more higher-priority DCUs is asserting a DCU arbiter lockout.
20	SEQNUMINCDIS	Sequence number increment disable. 0 = Increment sequence number for each new frame. 1 = Do not increment sequence number for each new frame.
21	BACKOFFDIS	Post frame backoff disable. 0 = Perform a post-Tx backoff following each frame. 1 = Do not perform a post-Tx backoff following each frame.
22	VC_NOACK	Virtual collision contention window increment policy. 0 = Virtual collisions do not increment (advance) the frame’s contention window. 1 = Virtual collisions increment the frame’s contention window.

Bit	Bit Name	Description
23	FAILEDIFSCTL	Blown IFS handling policy. This setting controls how the DCU handles the case in which the reading of a frame takes so long that the IFS spacing is met before the frame trigger level is reached. 0 = Send a frame on the air, which will cause a frame to be sent even if it is in violation of the IFS specification. 1 = Do not send the frame on the air. Instead, act as if the frame had been sent on the air but failed and initiate the retry procedure. A retry will be charged against the frame. If more retries are permitted, the frame will be retried. If the retry limit has been reached, the frame will fail.
31–24	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.6 Frame Sequence Number Register (D_SEQNUM)

Address offset: 0x1140
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
11–0	SEQNUM	Local sequence number.
31–12	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.7 SIFS Setting Register (D_GBL_IFS_SIFS)

Address offset: 0x1030
Access: Read/Write
Cold reset: 0x0000_0280
Warm reset: 0x0000_0280

Bit	Bit Name	Description
15–0	IFS_SIFS	SIFS interval in core clocks. Fixed interval backoff following each valid packet.
31–16	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.8 DCU Global Slot Interval Register (D_GBL_IFS_SLOT)

Address offset: 0x1070
Access: Read/Write
Cold reset: 0x0000_0168
Warm reset: 0x0000_0168

Bit	Bit Name	Description
15–0	IFS_SLOT	Slot interval in core clocks.
31–16	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.9 EIFS Setting Register (D_GBL_IFS_EIFS)

Address offset: 0x10B0
Access: Read/Write
Cold reset: 0x0000_0d98
Warm reset: 0x0000_0d98

Bit	Bit Name	Description
15–0	IFS_EIFS	EIFS interval in core clocks.
31–16	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.10 Miscellaneous IFS Settings Register (D_GBL_IFS_MISC)

Address offset: 0x10F0
Access: Read/Write
Cold reset: 0x0000_a100
Warm reset: 0x0000_a100

Bit	Bit Name	Description
2–0	LFSR_SEL	LFSR select for randomizing backoff. If RNDMLFSRDIS (bit 24 of this register) is set to 1, then LFSR chooses one of eight random number sequences to be used for generation backoff counts.
3	TURBO	Turbo mode indication. Software is required to keep this register consistent with the turbo/non-turbo state of the overall system. This is not a status bit generated by the MAC, but rather a control bit that must be maintained by software, so that certain parts of the MAC that are sensitive to whether the system is in turbo mode will operate correctly. 0 = Disable turbo mode. 1 = Enable turbo mode.
9–4	RES	Reserved. Must be written with zero. Can contain any value on read.
19–10	MICROSEC	µs interval in core clocks.

Bit	Bit Name	Description
21–20	ARB_DLY	DCU arbiter delay. Controls the delay between arbiter unlock and the next arbitration decision. This is for debugging use only. Leave at the default (reset) setting for normal use. 0 = DCU arbitration occurs 64 cycles after unlock. 1 = DCU arbitration occurs 128 cycles after unlock. 2 = DCU arbitration occurs 256 cycles after unlock. 3 = DCU arbitration occurs 32 cycles after unlock.
23–220	RES	Reserved. Must be written with zero. Can contain any value on read.
24	RNDMLFSRDIS	Random LFSR selection disable. 0 = Select random LSFR (bits [2:0] of this register). The random selection method is meant to ensure independence of the LFSR output values for nodes on different PCI buses but on the same network, as well as for multiple nodes connected to the same physical PCI bus. 1 =Disable random LFSR selection. Use LFSR_SEL to select LFSR.
31–25	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.11 Frame Prefetch Register (D_FPCTL)

Address offset: 0x1230
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
3–0	FPDCUSEL	Selects which DCU for non-burst frame prefetch. 0x0 = DCU 0 0x1 = DCU 1 0x2 = DCU 2 0x3 = DCU 3 0x4 = DCU 4 0x5 = DCU 5 0x6 = DCU 6 0x7 = DCU 7 0x8 = DCU 8 0x9 = DCU 9 0xA to 0xF = Reserved
4	FPNORMALEN	Normal (non-burst) frame prefetch enable. 0 = Disable non-burst frame prefetch. 1 = Enable non-burst frame prefetch. (Only for DCU specified in bits [3:0] of this register.)
14–5	FPBURSTEN	Burst frame prefetch enable. 0 = Disable burst frame prefetch. 1 = Enable burst frame prefetch.
31–15	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.12 Transmit Pause Control/Status (D_TXPSE)

Address offset: 0x1270
Access: Read/Write
Cold reset: 0x0001_0000
Warm reset: 0x0001_0000

Bit	Bit Name	Description
9–0	TXPSEEN	Request that the DCUs pause transmission. For each bit position corresponding to a DCU: 0 = Allow DCU to continue to transmit normally. 1 = Request that DCU pause transmission as soon as it is able to do so.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
16	TXPSE	Transmit pause status. 0 = Transmit pause request has not yet taken effect. This means that some of the DCUs for which a transmission pause request has been issued via bits [9–0] of this register still are transmitting and have not yet paused. 1 = All DCUs for which a transmission pause request has been issued via bits [9–0] of this register, if any, have in fact paused their transmissions. Note that if no transmission pause request is pending (that is, bits [9–0] of this register are all set to 0), then this transmit pause status bit will be set to one.
31–17	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.13 Transmit Filter Command Register (D_TXBLK_CMD)

Address offset: 0x1038
Access: Write Only
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
15–0	TXBLKMASK	Bitmask. For each bit corresponding to a bit in the Tx filter slice: 0 = Disable bit update. 1 = Enable bit update.
19–16	TXBLKSLICE	Slice number. Specifies which 16 bit slice to update. 0 = Enable filter bits[15:0] 1 = Enable filter bits[31:16] 2 = Enable filter bits[47:32] 3 = Enable filter bits[63:48] 4 = Enable filter bits[79:64] 5 = Enable filter bits[95:80] 6 = Enable filter bits[111:96] 7 = Enable filter bits[127:112]

Bit	Bit Name	Description
23–20	TXBLKDCU	DCU number. Specifies which DCU to enable. 0x0 = DCU 0 0x1 = DCU 1 0x2 = DCU 2 0x3 = DCU 3 0x4 = DCU 4 0x5 = DCU 5 0x6 = DCU 6 0x7 = DCU 7 0x8 = DCU 8 0x9 = DCU 9 0xA to 0xF = Reserved
27–24	TXBLKCMD	Command. Determines what operation will be performed on the selected slice. 0 = Clear enabled Tx filter bits. 1 = Set enabled Tx filter bits.
31–28	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.14 Transmit Filter Data Register (D_TXBLK_DATA)

TXBLKDATA returns a 32 bit slice of the Tx filter. Each DCU maintains a 128 bit Tx filter, for a total of 1280 bits (10 DCU * 128 bits/DCU). The 32 bit slice to be returned is selected by ADDRESS_OFFSET shown in Table 9-5.

Address offset: (0x1038 + ADDRESS_OFFSET)
Access: Read Only
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
31–0	TXBLKDATA	Transmit filter bits (read only).

Table 9-5. **TXBLKDATA Address Offset**

DCU	Transmit Filter Bits	Address Offset
0	31:0	0x000
0	63:32	0x040
0	95:64	0x080
0	127:96	0x0c0
1	31:0	0x100
1	63:32	0x140
1	95:64	0x180
1	127:96	0x1c0
2	31:0	0x200
2	63:32	0x240
2	95:64	0x280
2	127:96	0x2c0

Table 9-5. **TXBLKDATA Address Offset (continued)**

DCU	Transmit Filter Bits	Address Offset
3	31:0	0x300
3	63:32	0x340
3	95:64	0x380
3	127:96	0x3c0
4	31:0	0x400
4	63:32	0x440
4	95:64	0x480
4	127:96	0x4c0
5	31:0	0x500
5	63:32	0x540
5	95:64	0x580
5	127:96	0x5c0
6	31:0	0x600
6	63:32	0x640
6	95:64	0x680
6	127:96	0x6c0
7	31:0	0x700
7	63:32	0x740
7	95:64	0x780
7	127:96	0x7c0
8	31:0	0x004
8	63:32	0x044
8	95:64	0x084
8	127:96	0x0c4
9	31:0	0x104
9	63:32	0x144
9	95:64	0x184
9	127:96	0x1c4

9.4.15 Clear Transmit Filter Register (D_TXBLK_CLR)

Address offset: 0x143C
Access: Write Only
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TXBLK_CLR	TXBLK_CLR clears all 128 bits of the Tx filter for that DCU. For each bit corresponding to a DCU: 0 = Ignore. 1 = Clear this DCU’s Tx filter.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.4.16 Set Transmit Filter Register (D_TXBLK_SET)

Address offset: 0x147C
Access: Write Only
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
9–0	TXBLK_SET	TXBLK_SET sets all 128 bits of the Tx filter for that DCU. For each bit corresponding to a DCU: 0 = Ignore. 1 = Set this DCU’s Tx filter.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5 PCI Clock Domain Registers

Table 9-6 summarizes the PCI clock domain registers and their descriptions provided by the AR5212. These registers remain accessible using the PCI/CardBus interface when the AR5212 is in sleep mode. These domain registers are located at offset addresses 0x4000 to 0x5000.

Table 9-6. **PCI Clock Domain Register Summary**

Offset	Name	Description	Page
0x4000	RC	Reset control register. Controls reset in various modes.	page 140
0x4004	SCR	Sleep control register. Sleep mode control and duration.	page 140
0x4008	INTPEND	Interrupt pending, indicates when a sleep mode interrupt is pending.	page 141
0x400C	SFR	Force sleep register. Force sleep mode immediately.	page 141
0x4010	PCICFG	PCI configuration, control access of EEPROM, and CLKRUN enable.	page 142
0x4014	GPIOCR	GPIO configuration, provides individual control of the GPIOs that are configured as input or output pins, and interrupt ability.	page 144
0x4018	GPIODO	GPIO data output, provides data output values to the GPIOs that are configured as outputs.	page 145
0x401C	GPIODI	GPIO data input, provides data values input to the GPIOs that are configured as inputs.	page 146
0x4020	SREV	Silicon revision, provides revision identification of the AR5212.	page 146
0x4024	SLE	Sleep enable alias.	page 147
0x4028	TXEPOST	TXE write posting register.	page 147
0x402C	QSM	QCU sleep mask.	page 148
0x5000– 0x50FC	CIST	Card Information Structure (CIS) tuple registers.	page 148

9.5.1 Reset Control Register (RC)

This register controls the device's response under various reset conditions, and returns the current value on a read. This register remains accessible when the AR5212 is in sleep mode.

Address/offset: 0x4000
Access: Read/Write
Cold reset: 0x0000_000F
Warm reset: Unaffected

Bit	Bit Name	Description
0	RMAC	MAC Warm Reset. 0 = Allow MAC to run normally. 1 = Hold MAC in warm reset.
1	RBB	0 = Allow baseband to run normally. 1 = Hold baseband in warm reset.
3–2	RES	Reserved. Must be written with zero. On read, can contain any value.
4	RPCI	0 = Allow PCI to run normally. 1 = Hold PCI in warm reset (auto clear after 64 PCI clocks).
31–5	RES	Reserved. Must be written with zero. On read, can contain any value.

9.5.2 Sleep Control Register (SCR)

This register controls when the device is in sleep mode and for how long, and returns the current value on a read. This register remains accessible when the AR5212 is in sleep mode.

Address offset: 0x4004
Access: Read/Write
Cold reset: 0x0001_XXXX
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	SLDUR	Sleep duration. This is the length of time the device is to remain in sleep mode, in units of 128µs (1/8th of an 802.11 time unit [TU]).
17–16	SLE	Sleep enable. 00 = Force wake. 01 = Force sleep. 10 = Allow sleep logic to control sleep/wake state (normal operation). 11 = Reserved. Must not write these values.
18	SLRSTCTL	Sleep duration timing policy. 0 = Sleep counter is reset at each occurrence of TBTT when the chip is awake. 1 = Sleep counter is reset at each occurrence of TBTT when the chip is awake and the sleep duration has already expired (sleep count is zero).
19	SLSETCTL	Sleep duration write handling policy. 0 = Set sleep counter to SLDUR, each time the SLDUR is written. 1 = Clear sleep counter, each time the SLDUR is written.

Bit	Bit Name	Description
20		Sleep policy mode. 0 = Use STA in a BSS sleep policy mode. 1 = Use Adhoc sleep policy mode.
21		Sleep performance counter MIB interrupt enable. 0 = The sleep performance counter logic will never generate a MIB interrupt, regardless of the values of the various sleep performance counter registers. 1 = When the most-significant bit of any of the sleep performance counter registers (SPC_x) becomes set, the logic will signal. a MIB interrupt.
31–22	RES	Reserved. Must be written with zero. On read, can contain any value.

9.5.3 PCI Clock Domain Interrupt Pending Register (INTPEND)

This register indicates when an interrupt is pending. This register remains accessible when the AR5212 is in sleep mode.

Address offset: 0x4008
Access: Read Only
Cold reset: 0x0000_0000
Warm reset: Unaffected

Bit	Bit Name	Description
0	SMIP	Interrupt pending. Indicates whether AR5212 is asserting the PCI_INT_L signal. Note that if this bit is asserted, the "Primary Interrupt Status Register (ISR_P)" on page 103 must be read to determine what interrupts are pending and to clear the PCI_INT_L assertion. 0 = Host interrupt is not asserted. 1 = Host interrupt is asserted.
31–1	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.4 Sleep Force Register (SFR)

This register can be programmed to immediately force the device into sleep mode. On read, it returns the current value. This register remains accessible when the AR5212 is in sleep mode.

Address offset: 0x400C
Access: Write Only (reads always return 0)
Cold reset: 0x0000_0000
Warm reset: Unaffected

Bit	Bit Name	Description
0	SF	Force sleep immediately. 0 = Ignore. 1 = Enter sleep mode immediately.
1	WF	0 = Ignore 1 = Exit sleep mode immediately.
31–2	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.5 PCI Clock Domain Configuration/Status Register (PCICFG)

This register permits the host to access the EEPROM, and enables or disables the PCI CLKRUN functionality. On read, this register returns the current value. This register remains accessible when the AR5212 is in sleep mode.

Address offset: 0x4010
Access: Read/Write
Cold reset: 0x0001_0000
Warm reset: Unaffected

Bit	Bit Name	Description
0	RES	Reserved. Must be written with zero. Can contain any value on read.
1		Sleep clock select. Resets to 0, but then is overwritten with the value of the SLEEP_CLK_SEL bit in the EEPROM, if EEPROM loading is enabled.
2	CLKRUNEN	CLKRUNEN Enable. Resets to 0, but then is overwritten with the value of the CLKRUNEN bit in the EEPROM, if EEPROM loading is enabled. 0 = Force host to keep PCI clock running continuously. 1 = Permit host to halt PCI clock while idle.
4–3	EEPROMSIZE	Automatically-determined EEPROM size (read-only). 0 = EEPROM is 4 KB. 1 = EEPROM is 8 KB. 2 = EEPROM is 16 KB. 3 = EEPROM size could not be determined. Typical causes for this result: ■ EEPROM is missing. ■ EEPROM is unprogrammed. ■ EEPROM is programmed with an incorrect EEPROM_MAGIC value. ■ Loading from the EEPROM is disabled because the EPRM_EN_L pin is negated. Disabling the EEPROM load also causes the automatic size determination process to be skipped.
6–5	LEDCTL	LED control based on association status provided by software writing to these bits. See Table 2-8, "LED Functionality," on page 29. 0 = STA is not associated and is not presently attempting to associate. 1 = STA is not associated but is presently attempting to associate. 2 = STA is associated.
9–7		PCI observation bus mux select.
10		Disable fix for bad PCI CBE# generation. 0 = Enable the fix. 1 = Disable the fix and potentially allow a bad CBE value to appear on the PCI bus.
11	INTSLEN	Enable interrupt line assertion when asleep. 0 = Do not assert host interrupt when asleep or preparing to sleep. 1 = Assert host interrupt even when sleep or preparing to sleep.
12	RES	Reserved. Must be written with zero. Can contain any value on read.
13	INTWKDIS	Disable logic to force chip awake when an interrupt is pending. 0 = Enable logic to force chip awake while an interrupt is pending. 1 = Disable logic to permit chip to sleep even if an interrupt is pending.

Bit	Bit Name	Description
14		LED hysteresis on sleep exit disable. 0 = When exiting from sleep, the LEDs will continue to display sleep mode for 20 milliseconds. 1 = When exiting from sleep, the LEDs will immediately transition to normal (bytes/sec.) operation.
15	RES	Reserved. Must be written with zero. Can contain any value on read.
16	ASLEEP	Sleep/Power-down indication. 0 = Chip is awake. 1 = Chip is asleep.
19–17	LEDMODE	LED mode select. If the STA is associated (LEDCTL=2): 0 = Blink LED proportional to the count of all Tx bytes and those Rx bytes that pass the Rx filter. 1 = Blink LED proportional to all Tx and Rx bytes. 2 = Blink the power LED for each Tx byte and the network LED for each Rx byte. This is meant mainly as a debugging mode. 3 = Blink the LEDs randomly.
22–20	LEDBLINK	LED blink threshold select. Control LED blink rate if the STA is associated. Six LED blink rates exist (slowest, four intermediate rates, and fastest) as shown below in the LED Blink Rate table. When the LED mode select is set to one of the two bytes/second modes, this field determines the mapping from bytes/sec of data transfer to each of the six LED blink rates. This field has no effect if the LED mode select field is not set to one of the bytes/sec modes. The LED Blink Rate table lists the range of bytes/sec values to achieve each LED blink rate for each setting of the LED blink rate threshold field:

LED Blink Rate

LED Blink Threshold	Slowest	1	2	3	4	Fastest
0	<8K	8K-512K	512K-1M	1M-2M	2M-4M	>4M
1	<4K	4K-256K	256K-512K	512K-1M	1M-2M	>2M
2	<2K	2K-128K	128K-256K	256K-512K	512K-1M	>1M
3	<2K	2K-32K	32K-64K	64K-256K	256K-1M	>1M
4	<4K	4K-64K	64K-128K	128K-512K	512K-2M	>2M
5	<8K	8K-64K	64K-128K	128K-512K	512K-2M	>2M
6	<8K	8K-128K	128K-256K	256K-1M	1M-2M	>2M
7	<4K	4K-64K	64K-128K	128K-256K	256K-1M	>1M

23	LEDSLOW	LED slowest blink rate mode. If the STA is associated, and there is no activity (Tx or Rx): 0 = Blink the LED at the slowest data rate. 1 = Turn off the LED at the slowest data rate.
25–24		Sleep clock rate indication. 0 = Sleep clock is approximately 32 MHz. 1 = Sleep clock is approximately 4 MHz. 2 = Sleep clock is approximately 1 MHz. 3 = Sleep clock is approximately 32 KHz.
31–26	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.6 GPIO Control Register (GPIOCR)

This register controls operations of the GPIO pins, and remains accessible when the AR5212 is in sleep mode.

Address offset: 0x4014
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: Based on bit 4 of Reset Control Register (refer to "Reset Control Register (RC)" on page 140). Value can be 0x0000_0000 or Unaffected.

Bit	Bit Name	Description
1–0	GPCR0	Drive Control for GPIO_0. 0 = Never drive pin GPIO_0. 1 = Drive pin GPIO_0 if GPIODO[0] is clear. 2 = Drive pin GPIO_0 if GPIODO[0] is set. 3 = Always drive pin GPIO_0.
3–2	GPCR1	Drive control for GPIO_1. 0 = Never drive pin GPIO_1. 1 = Drive pin GPIO_1 if GPIODO[1] is clear. 2 = Drive pin GPIO_1 if GPIODO[1] is set. 3 = Always drive pin GPIO_1.
5–4	GPCR2	Drive control for GPIO_2. 0 = Never drive pin GPIO_2. 1 = Drive pin GPIO_2 if GPIODO[2] is clear. 2 = Drive pin GPIO_2 if GPIODO[2] is set. 3 = Always drive pin GPIO_2.
7–6	GPCR3	Drive control for GPIO_3. 0 = Never drive pin GPIO_3. 1 = Drive pin GPIO_3 if GPIODO[3] is clear. 2 = Drive pin GPIO_3 if GPIODO[3] is set. 3 = Always drive pin GPIO_3.
9–8	GPCR4	Drive control for GPIO_4. 0 = Never drive pin GPIO_4. 1 = Drive pin GPIO_4 if GPIODO[4] is clear. 2 = Drive pin GPIO4 if GPIODO[4] is set. 3 = Always drive pin GPIO_4.
11–10	GPCR5	Drive control for GPIO_5. 0 = Never drive pin GPIO_5. 1 = Drive pin GPIO_5 if GPIODO[5] is clear. 2 = Drive pin GPIO_5 if GPIODO[5] is set. 3 = Always drive pin GPIO_5.

Bit	Bit Name	Description
14–12	GPINTSEL	Interrupt pin select. These bits select 1 of the 6 GPIOs to support interrupt generation. 0 = Use GPIODI[0] to generate an interrupt. 1 = Use GPIODI[1] to generate an interrupt. 2 = Use GPIODI[2] to generate an interrupt. 3 = Use GPIODI[3] to generate an interrupt. 4 = Use GPIODI[4] to generate an interrupt. 5 = Use GPIODI[5] to generate an interrupt.
15	GPINTEN	Enable GPIO interrupt. 0 = Disable GPIO interrupt. 1 = Enable GPIO interrupt using the selected GPIO pin and pin level.
16	GPINTLVL	Interrupt level select. 0 = Generate interrupt if selected pin is low. 1 = Generate interrupt if selected pin is high.
31–17	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.7 GPIO Data Output Register (GPIODO)

This register provides output to the GPIO pins that are configured as outputs, and remains accessible when the AR5212 is in sleep mode.

Address offset: 0x4018
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: Based on bit 4 of Reset Control Register (refer to "Reset Control Register (RC)" on page 140). Value can be 0x0000_0000 or Unaffected.

Bit	Bit Name	Description
0	GPIODO_0	GPIO_0 data output (GPIODO[0]).
1	GPIODO_1	GPIO_1 data output (GPIODO[1]).
2	GPIODO_2	GPIO_2 data output (GPIODO[2]).
3	GPIODO_3	GPIO_3 data output (GPIODO[3]).
4	GPIODO_4	GPIO_4 data output (GPIODO[4]).
5	GPIODO_5	GPIO_5 data output (GPIODO[5]).
31–6	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.8 GPIO Data Input Register (GPIODI)

This register provides access to the GPIOs that are configured as inputs, and remains accessible when the AR5212 is in sleep mode.

Address offset: 0x401C
Access: Read Only
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
0	GPIODI_0	GPIO_0 data input (GPIODI[0]).
1	GPIODI_1	GPIO_1 data input (GPIODI[1]).
2	GPIODI_2	GPIO_2 data input (GPIODI[2]).
3	GPIODI_3	GPIO_3 data input (GPIODI[3]).
4	GPIODI_4	GPIO_4 data input (GPIODI[4]).
5	GPIODI_5	GPIO_5 data input (GPIODI[5]).
31–6	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.9 Silicon Revision Register (SREV)

This register indicates the revision of this device, and remains accessible when the AR5212 is in sleep mode.

Address offset: 0x4020
Access: Read Only
Cold reset: 0x53
Warm reset: 0x53

Bit	Bit Name	Description
3–0	REVISION	Silicon revision identification.
7–4	VERSION	Silicon version identification. Current version is 0x5.
31–8	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.10 Sleep Enable Alias Register (SLE)

This register allows the sleep enable field of the "Sleep Control Register (SCR)" on page 140 to be altered without causing a reload of the sleep duration counter in the MAC sleep logic. Writes to this register change the same physical register bits as a write to SCR.

Address offset: 0x4024
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	RES	Reserved. Must be written with zero. Can contain any value on read.
17–16	SLE_ALIAS	Sleep enable. 00 = Force wake. 01 = Force sleep. 10 = Allow sleep logic to control sleep/wake state (normal operation). 11 = Reserved. Must not write these values.
31–18	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.11 TXE Write Posting Register (TXEPOST)

This register

Address offset: 0x4028
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: Unaffected

Bit	Bit Name	Description
9–0	RES	Values to write to the TXE bits. Each bit (9–0) corresponds to a QCU Q's TXE bit.
15–10	RES	Reserved. Must be written with zero. Can contain any value on read.
16		Valid indication. When set, indicates that a posted TXE write has not yet completed. When clear, indicates that no posted TXE write is pending.
31–17	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.12 QCU Sleep Mask Register (QSM)

This register

Address offset: 0x402C
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: Unaffected

Bit	Bit Name	Description
9–0		Selects which QCUs control when the MAC can sleep. The MAC will not sleep until all selected QCUs have their TXE bits clear and no pending frames.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.5.13 CIS Tuples Register (CIST)

Address offset: (0x5000 + TUPLE_OFFSET)
Access: Read Only
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
31–0	TUPLES	Card Information Structure data as read from the EEPROM.

9.6 EEPROM Interface Registers

Table 9-7 summarizes the EEPROM interface registers and their descriptions provided by the AR5212. These registers are located at offset address 0x6000 to 0x6010.

Table 9-7. **EEPROM Interface Register Summary**

Offset	Name	Description	Page
0x6000	E_ADDR	EEPROM address. Enables reading and writing of the EEPROM address.	page 149
0x6004	E_DATA	EEPROM data. Enables reading and writing of the EEPROM data.	page 149
0x6008	E_CMD	EEPROM command. Initiates an EEPROM read and write.	page 150
0x600C	E_STS	EEPROM status. Specifies read and write errors and the status of read and write completion.	page 150
0x6010	E_CFG	EEPROM configuration. Automatically determines EEPROM size.	page 151

9.6.1 EEPROM Address Register (E_ADDR)

Address offset: 0x6000
Access: Read/Write
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
9–0	E_ADDR	EEPROM entry address.
31–10	RES	Reserved. Must be written with zero. Can contain any value on read.

9.6.2 EEPROM Data Register (E_DATA)

Address offset: 0x6004
Access: Read/Write
Cold reset: Undefined
Warm reset: Undefined

Bit	Bit Name	Description
15–0	E_DATA	EEPROM read/write data.
31–16	RES	Reserved. Must be written with zero. Can contain any value on read.

9.6.3 EEPROM Command Register (E_CMD)

Address offset: 0x6008
Access: Write Only (reads always return 0)
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
0	E_READ	0 = Ignore. 1 = Initiate an EEPROM read.
1	E_WRITE	0 = Ignore. 1 = Initiate an EEPROM write.
2	E_RESET	0 = Ignore. 1 = Initiate an EEPROM reset.
31–3	RES	Reserved. Must be written with zero. Can contain any value on read.

9.6.4 EEPROM Status Register (E_STS)

Address offset: 0x600C
Access: Read Only
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
0	E_READ_ERR	0 = No read error. 1 = Read error.
1	E_READ_DONE	0 = Read data not ready. 1 = Read data ready.
2	E_WRITE_ERR	0 = No write error. 1 = Write error.
3	E_WRITE_DONE	0 = Write data not ready. 1 = Write data ready.
31–4	RES	Reserved. Must be written with zero. Can contain any value on read.

9.6.5 EEPROM Configuration Register (E_CFG)

Address offset: 0x6010
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
1–0	E_SIZE	EEPROM size determination override. Allows software to override the results of the automatic EEPROM size determination (see EEPROMSIZE bits [4:3] in "PCI Clock Domain Configuration/Status Register (PCICFG)" on page 142). 0 = Automatically determine EEPROM size. 1 = Force EEPROM size to be 4 Kb. 2 = Force EEPROM size to be 8 Kb. 3 = Force EEPROM size to be 16 Kb.
2	E_FASTWRITE	Disable wait for write completion. 0 = Disable fast write. 1 = Enable fast write.
4–3	E_RATE	EEPROM clock rate control. Listed values are generated regardless of whether the chip is in turbo mode or not. 0 = Set EEPROM clock rate to 156 KHz. 1 = Set EEPROM clock rate to 312 KHz. 2 = Set EEPROM clock rate to 625 KHz.
7–5	RES	Reserved. Must be written with zero. Can contain any value on read.
23–8	E_PROTECT	EEPROM write protect key. This is the value that was loaded from the EEPROM_PROTECT entry in the EEPROM. This field is Read Only. Resets to 0x0, but then is loaded from the EEPROM. Refer to "EEPROM Read/Write Protection Mechanism" on page 26.
24	E_EPRM_EN_L	Correct EPRM_EN_L pin value. This field is Read Only. 0 = Enable loading from the EEPROM on exit from cold reset 1 = Disable loading from the EEPROM on exit from cold reset. Default initial values for all PCI configuration registers are being used.
31–25	RES	Reserved. Must be written with zero. Can contain any value on read.

9.7 PCU Registers

Table 9-8 summarizes the PCU registers and their descriptions provided by the AR5212. PCU registers are located at offset address 0x8000 to 0x881C.

Table 9-8. **PCU Register Summary**

Offset	Name	Description	Page
0x8000	STA_ID0	Lower 32 bits of the STA MAC address.	page 154
0x8004	STA_ID1	Upper 16 bits of the STA MAC address.	page 154
0x8008	BSS_ID0	Lower 32 bits of the BSSID.	page 155
0x800C	BSS_ID1	Upper 16 bits of the BSSID.	page 155
0x8014	TIME_OUT	Time-out to wait for ACK and CTS in clock cycles.	page 156
0x8018	RSSI_THR	Beacon RSSI warning threshold.	page 156
0x801C	USEC	Specifies µs duration for core and reference clocks. Specifies Tx and Rx latencies in µs.	page 157
0x8020	BEACON	Specifies the beacon period in time units (TUs)/1024 µs.	page 157
0x8024	CFP_PERIOD	Specifies the CFP interval in TUs/1024 µs.	page 158
0x8028	NEXT_BEACON	Specifies the next beacon time in TUs/1024 µs.	page 158
0x802C	DBA	Specifies the next DMA beacon alert time in 1/8 TUs/1024 µs.	page 158
0x8030	SBA	Specifies the next software beacon alert time in 1/8 TUs/1024 µs.	page 159
0x8034	ATIM_WIN	Specifies the end of the next ATIM window in TUs/1024 µs.	page 159
0x8038	MAX_CFP_DUR	Specifies maximum CFP interval in TUs/1024 µs.	page 159
0x803C	RX_FILTER	Rx frame filter control.	page 160
0x8040	MCAST_FIL0	Specifies lower 32 bits of the multicast filter mask.	page 161
0x8044	MCAST_FIL1	Specifies upper 32 bits of the multicast filter mask.	page 161
0x8048	DIAG_SW	Enables and disables auto-generated ACK when a valid key is not found for a Rx frame. Various PCU functions.	page 162
0x804C	TSF_L32	Specifies lower 32 bits of the local clock (TSF).	page 163
0x8050	TSF_U32	Specifies upper 32 bits of the local clock (TSF).	page 163
0x8054	RES	Reserved.	
0x8058	DEF_ANTENNA	Specifies default antenna.	page 164
0x8080	LAST_TSTP	Specifies lower 32 bits of the last beacon time-stamp.	page 165
0x8084	NAV	Specifies the current NAV value in µs.	page 165
0x8088	RTS_OK	Specifies successful RTS/CTS exchange counter.	page 165
0x808C	RTS_FAIL	Specifies failed RTS/CTS exchange counter.	page 166
0x8090	ACK_FAIL	Specifies failed DATA/ACK exchange counter.	page 166
0x8094	FCS_FAIL	Specifies failed FCS counter.	page 166
0x8098	BEACON_CNT	Specifies beacon counter.	page 167
0x80D4	SLP1	Sleep 1.	page 167
0x80D8	SLP2	Sleep 2.	page 168

Table 9-8. **PCU Register Summary (continued)**

Offset	Name	Description	Page
0x80DC	SLP3	Sleep 3.	page 168
0x80E0	BSSMSKL	BSSID mask lower 32-bits.	page 168
0x80E4	BSSMSKU	BSSID mask upper 32-bits.	page 169
0x80E8	TPC	Transmit power control.	page 169
0x80EC	TFC	Transmit frame counter.	page 169
0x80F0	RFC	Receive frame counter.	page 170
0x80F4	RRC	Receive clear counter.	page 170
0x80F8	CC	Cycle counter.	page 170
0x80FC	QT1	Quiet time 1.	page 171
0x8100	QT2	Quiet time 2.	page 171
0x8104	TSF	TSF parameters.	page 171
0x8108	NOACK	QOS no ACK locator.	page 172
0x810C	PHYERR	PHY error mask.	page 173
0x8114	ACKSIFS	ACKSIFS table changes.	page 174
0x8700— 0x877C	DURS	ACKSIFS table changes.	page 174

9.7.1 STA Address 0 (STA_ID0)

This register contains the lower 32 bits of the STA's MAC address. The upper 16 bits are contained in the STA_ID1 register.

Address offset: 0x8000
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
31–0	STA_ADDR_L32	STA MAC address lower 32 bits [31:0].

9.7.2 STA Address 1 (STA_ID1)

This register contains the upper 16 bits of the STA's MAC address. The lower 32 bits are contained in the STA_ID0 register. This register also provides PCU configuration capabilities.

Address offset: 0x8004
Access: Read/Write
Cold reset: 0x0000_XXXX
Warm reset: 0x0000_XXXX

Bit	Bit Name	Description
15–0	STA_ADDR_U16	STA MAC address upper 16 bits [48:32].
16	AP	Access Point Operation Enable. 0 = Disable. 1 = Enable.
17	ADHOC	Ad hoc Operation Enable. 0 = Disable. 1 = Enable.
18	PWR_SV	Power Save Reporting (in self-generated frames) Enable. 0 = Disable. 1 = Enable.
19	NO_KEYSRCH	Key Table Search Disable. 0 = Enable. 1 = Disable.
20	PCF	PCF Protocol Observation Enable. 0 = Ignore PCF protocol. 1 = Observe PCF protocol (no frames transmitted during CFP).
21	USE_DEFANT	When AntModeXmit in the Tx descriptor is 0: 0 = Use LAST_TX_ANT in the key cache as Tx antenna. 1 = Use the LSB of the Default Antenna register as Tx antenna.
22	DEFANT_UPDATE	Default Antenna. 0 = Do not update the Default Antenna register after each Tx frame. 1 = Update the Default Antenna register after each Tx frame.
23	RTS_USE_DEF	0 = Use AntModeXmit in the Tx descriptor for RTS. 1 = Use the Default Antenna register for RTS.

Bit	Bit Name	Description
24	ACKCTS_6MB	Rate for ACK and CTS. 0 = Use highest PHY mandatory rate that is less than or equal to the Rx rate. 1 = Use 6 Mbps rate in 802.11a mode. Use 1 Mbps rate in 802.11b mode.
25	BASE_RATE_11B	802.11b base rate. 0 = 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps. 1 = 1 Mbps and 2 Mbps.
26	SECTOR_SELF_GEN	1 = Use default antenna for self-generated frames. 0 = Use current antenna for self-generated frames.
27	CRPT_MIC_ENABLE	1 = Enable TKIP Michael insertion and check. 0 = Disable TKIP Michael insertion and check.
28	KSRCH_MODE	1 = Search keycache first; if no match then use offset. 0 = Use offset (don't send).
29	Preserve_SEQNUM	1 = Preserve sequence number generated by software. 0 = Overwrite sequence number generated by software.
30	CBCIV_ENDIAN	
31	ADHOC_MCAST_SEARCH	1 = Enable the keycache search for adhoc multicast packets. 0 = Disable the keycache search for adhoc multicast packets.

9.7.3 BSS Address 0 (BSS_ID0)

This register contains the lower 32 bits of the BSS identification information.

Address offset: 0x8008
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
31–0	BSSID_L32	BSSID lower 32 bits [31:0].

9.7.4 BSS Address 1 (BSS_ID1)

This register contains the upper 16 bits of BSSID and an association identification.

Address offset: 0x800C
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	BSSID_U16	BSSID upper 16 bits [48:32].
31–16	AID	Association ID. Used by the PCU to calculate which TIM bit to check in a beacon when operating in the power-save mode.

9.7.5 Time Out (TIME_OUT)

This register contains the amount of time to wait for an acknowledgement or clear-to-send signal before issuing a timeout.

Address offset: 0x8014
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
13–0	ACK_TIME_OUT	Acknowledgement timeout. Time-out to wait for ACK in core clock cycles -1.
15–14	RES	Reserved. Must be written with zero. Contains zeros when read.
29–16	CTS_TIME_OUT	Clear-to-send timeout. Time-out to wait for CTS in core clock cycles -1.
31–30	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.6 Signal Strength Threshold (RSSI_THR)

This register contains the beacon RSSI warning threshold value.

Address offset: 0x8018
Access: Read/Write
Cold reset: bits 10–8 are 0, rest Unaffected
Warm reset: bits 10–8 are 0, rest Unaffected

Bit	Bit Name	Description
7–0	RSSI_THR	RSSI threshold. Beacon RSSI warning threshold value.
15–8	BEACON_MISS	Missed beacon threshold. These bits specify the number of beacons missed before the host is alerted.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.7 Timers (USEC)

This register defines the short and long frame retry limits, and the short and long STA retry limits, as specified in the IEEE 802.11a standard.

Address offset: 0x801C
Access: Read/Write
Cold reset: bit 29–20 are 0, rest are Unaffected
Warm reset: bit 29–20 are 0, rest are Unaffected

Bit	Bit Name	Description
6–0	USEC	µs duration in core clocks -1.
13–7	USEC32	µs duration in reference clocks -1.
22–14	TX_LATENCY	Transmit latency in µs -1.
28–23	RX_LATENCY	Receive latency in µs -1.
31–26	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.8 Beacon (BEACON)

This register controls beacon operation by the PCU.

Address offset: 0x8020
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	BEACON_PERIOD	Beacon period. Defined in TU.
22–16	TIM_OFFSET	Byte offset. The byte offset from the start of the MAC header to the bitmap control sub-field in the TIM is programmed by writing to these bits.
23	RES	Reserved. Must be written with zero. Contains zeros when read.
24	RESET_TSF	TSF reset (one shot, write only). 0 = No effect. 1 = Clear TSF to zero.
31–25	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.9 CFP Interval (CFP_PERIOD)

This register contains the CFP period.

Address offset: 0x8024
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
21–0	CFP_PERIOD	Contention-free repetition interval. STA in BSS. Defined in TU.
31–22	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.10 Next Beacon Time (NEXT_BEACON)

This timer register contains the time when the next beacon is expected in the TU (timer unit).

Address offset: 0x8028
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	NEXT_BEACON	Next beacon time. A STA must initialize the time that the next beacon is expected in the TU. For an AP, 0 must be written to this register to start TSF.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.11 DMA Beacon Alert Time (DBA)

This timer register contains the time to send the next DMA beacon alert.

Address offset: 0x802C (AP or ad hoc Mode)
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
18–0	DBA	Next DMA beacon time. Time for next DMA beacon alert (AP/ad hoc) or next wake-up time (STA in BSS) (1/8 TU).
31–19	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.12 Software Beacon Alert (SBA)

This timer register contains the time to send the next software beacon alert or the start time of the next CFP.

Address offset: 0x8030
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
24–0	SBA	Software beacon alert/next CFP start time. When AP = 1 in STA_ID1 register or ad hoc = 1, TIMER2 = Next software beacon alert in 1/8 TU. When AP = 0 in STA_ID1 register, TIMER2 = Next CFP start in 1/8 TU. Refer to "STA Address 1 (STA_ID1)" on page 154.
31–25	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.13 ATIM Window (ATIM_WIN)

This timer register contains the time when the ATIM window will end.

Address offset: 0x8034
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	ATIM_WIN	ATIM window time. The end of the next ATIM window (ad hoc) in TU.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.14 Maximum CFP Duration (MAX_CFP_DUR)

This register contains the maximum time for a CFP.

Address offset: 0x8038
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	MAX_CFP_DUR	Maximum CFP duration in TUs/1024 µs.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.15 Receive Filter (RX_FILTER)

This register determines Rx frame filtering.

Address offset: 0x803C
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
0	UNICAST	Unicast frame Enable. Enable reception of unicast (directed) frames that match the STA address. 0 = Disable — No ACK will return. 1 = Enable.
1	MULTICAST	Multicast frame Enable. Enable reception of multicast frames that match the multicast filter. 0 = Disable. 1 = Enable.
2	BROADCAST	Broadcast frame Enable. Enable reception of non beacon broadcast frames that originate from the BSS whose ID matches BSSID. 0 = Disable. 1 = Enable.
3	CONTROL	Control frame Enable. Enable reception of control frames. 0 = Disable. 1 = Enable.
4	BEACON	Beacon frame Enable. Enable reception of beacon frames. 0 = Disable. 1 = Enable.
5	PROMISCUOUS	Promiscuous Receive Enable. Enable reception of all frames, including errors. 0 = Disable. 1 = Enable.
6	RES	Reserved. Must be written with zero. Contains zeros when read.
7	PROBE_REQ	Probe request enable. Enables reception of all probe request frames.
31–8	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.16 Multicast Filter 0 (MCAST_FIL0)

This register contains the lower 32 bits of the multicast filter mask. The upper 32 bits of the multicast filter mask are contained in the MCAST_FIL1 register. Refer to "Rx frame Filtering" on page 67 for a description of the multicast filter.

Address offset: 0x8040
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
31–0	MCAST_FIL0	Multicast filter mask low. Lower 32 bits of multicast filter mask.

9.7.17 Multicast Filter 1 (MCAST_FIL1)

This register contains the upper 32 bits of the multicast filter mask. The lower 32 bits of the multicast filter mask are contained in the MCAST_FIL0 register. Refer to "Multicast Filter 0 (MCAST_FIL0)" on page 161 for a description of the multicast filter.

Address offset: 0x8044
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
31–0	MCAST_FIL1	Multicast filter mask high. Upper 32 bits of multicast filter mask.

9.7.18 PCU Diagnostic (DIAG_SW)

This register controls the operation of the PCU, including the enabling/disabling of acknowledgements, CTS, transmission, reception, encryption, loopback, FCS, channel information, and scrambler seeds.

Address offset: 0x8048
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
0	DIS_WEP_ACK	Acknowledgement disabled when a valid key is not found for the received frames in the key cache. 0 = Enable hardware ACK when a valid key is not found for the received frames. 1 = Disable hardware ACK when a valid key is not found for the received frames.
1	DIS_ACK	Acknowledgement generation disabled for all frames. 0 = Enable hardware ACK. 1 = Disable hardware ACK.
2	DIS_CTS	CTS Generation Disable. 0 = Enable hardware CTS. 1 = Disable hardware CTS.
3	DIS_ENC	Encryption Disable. 0 = Enable hardware encryption. 1 = Disable hardware encryption.
4	DIS_DEC	Decryption Disable. 0 = Enable hardware decryption. 1 = Disable hardware decryption.
5	DIS_RX	Reception Disable. 0 = Enable frame reception. 1 = Disable frame reception.
6	LOOP_BACK	Transmit Data Loopback Enable. 0 = Disable loopback. 1 = Enable loopback.
7	CORR_FCS	Corrupt FCS Enable. Enabling this bit causes an invalid FCS to be appended to a frame during transmission. 0 = Generate valid Tx FCS. 1 = Generate invalid Tx FCS.
8	CHAN_INFO	Channel Information Enable. Enabling this bit causes 56 bytes of channel information to be stored in the receive buffer before the frame data is stored. 0 = Disable channel information. 1 = Enable channel information.
16–9	RES	Reserved. Must be written with zero. Contains zeros when read.
17	PROTOCOL_DIS	Protocol field check disable. 0 = Enable check of protocol field. 1 = Disable check of protocol field.
31–18	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.19 Time Synchronization Function L32 (TSF_L32)

Address offset: 0x804C
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
31–0	TSF_L32	Lower 32 bits of the synchronization TSF.

9.7.20 Time Synchronization Function U32 (TSF_U32)

This register provides controls of IFS and EIFS clock cycles and enables carrier sense.

Address offset: 0x8050
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
31–0	TSF_U32	Upper 32 bits of the TSF.

9.7.21 ADC/DAC Test (TST_ADDAC)

Address offset: 0x8054
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
0	AD_TX	Continuous Tx mode. 0 = Disable continuous Tx test mode. 1 = Enable continuous Tx test mode.
1	AD_MODE	Test mode. 0 = Disable test mode. 1 = Enable test mode.
2	AD_EN	Test loop enable. 0 = Disable test loop. 1 = Enable test loop.
13–3	AD_LEN	Loop length minus 1.
14	AD_UPPER	Use upper 8 bits. 0 = Return lower eight bits of data. 1 = Return upper eight bits of data.
15	AD_MSB	State of MSB.
16	AD_SEL	Trigger select. 0 = rx_clear 1 = external

Bit	Bit Name	Description
17	AD_POL	Trigger polarity. 0 = trigger on falling edge. 1 = trigger on rising edge.
18	AD_CAPTURE	Continuous DAC write or ADC capture.
19	AD_TRIG	Begin capture.
20	AD_ARM	Arm Rx buffer for capture.
31–21	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.22 Default Antenna (DEF_ANTENNA)

Address offset: 0x8058
Access: Read/Write
Cold reset: 0x0000_0000
Warm reset: 0x0000_0000

Bit	Bit Name	Description
3–0	DEFANT	Used to listen for and receive the start of a frame sequence from a STA, except when 802.11b fast antenna diversity is in use. For receive, specifies whether to use antenna 1 or antenna 2. The USE_DEFANT bit (refer to "USE_DEFANT" on page 154) is set by software, then the DEFANT is selected for transmitting frames originating from the host. ■ AP—The default antenna does not change after it is programmed by software. sector AP mode—normally programmed to 001. The 1 omni antenna is used for listening and transmitting RTS (refer to "RTS_USE_DEF" on page 154). omni AP mode—can be set to either 001 or 010 ■ STA—For omni STA mode, the default antenna adapts to the best antenna for communication with the AP. It can be initialized to either 001 or 010, but it will change over time. Configuration Values: 0000 = Not Valid. 0001 = Antenna 1. Sectored AP mode or omni AP mode/omni STA mode. 0010 = Antenna 2. Omni AP mode/omni STA mode. 0011 = Sector antenna. 0100 = Sector antenna. 0101 = Sector antenna. 0110 = Sector antenna. 0111 = Sector antenna. 1000 = Sector antenna. 1001 = Sector antenna. 1010 = Sector antenna. 1011 = Sector antenna. 1100 = Sector antenna. 1101 = Sector antenna. 1110 = Sector antenna. 1111 = Not applicable.
31–4	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.23 Last Timestamp (LAST_TSTP)

This threshold register contains the lower 32 bits of the timestamp of the last beacon.

Address offset: 0x8080
Access: Read Only
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
31–0	LAST_TSTP	Beacon timestamp. Lower 32 bits of timestamp of the last beacon received.

9.7.24 NAV Value (NAV)

This register contains the current value of NAV.

Address offset: 0x8084
Access: Read Only
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
25–0	NAV	Current NAV value.
31–26	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.25 RTS OK (RTS_OK)

This register counts the number of successful RTS exchanges. The counter stops at 0xFFFF. After a read, this register is automatically reset to 0.

Address offset: 0x8088
Access: Read Only
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	RTS_OK	RTS/CTS exchange success counter.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.26 RTS Fail Count (RTS_FAIL)

This register counts the number of failed RTS exchanges. The counter stops at 0xFFFF. After a read, this register is automatically reset to 0.

Address offset: 0x808C
Access: Read Only
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	RTS_FAIL	RTS/CTS exchange failure counter.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.27 ACK Fail Count (ACK_FAIL)

This register counts the number of failed acknowledgements. The counter stops at 0xFFFF. After a read, this register is automatically reset to 0.

Address offset: 0x8090
Access: Read Only
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	ACK_FAIL	DATA/ACK failure counter.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.28 FCS Fail Count (FCS_FAIL)

This register counts the number of failed frame check sequences. The counter stops at 0xFFFF. After a read, this register is automatically reset to 0.

Address offset: 0x8094
Access: Read Only
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	FCS_FAIL	FCS failure counter.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.29 Beacon Count (BEACONCNT)

This register counts the number of valid beacon frames received. The counter stops at 0xFFFF. After a read, this register is automatically reset to 0.

Address offset: 0x8098
Access: Read Only
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
15–0	BEACONCNT	Valid beacon counter.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.30 SLEEP 1 (SLP1)

The Sleep 1 register, in conjunction with the Sleep 2 and Sleep 3 registers, control when the AR5212 should wake when waiting for receive traffic from the AP. These registers are only used when the AR5212 is in STA mode.

Address offset: 0x80D4
Access: Read/Write
Cold reset: 0x0512_AAAA
Warm reset: 0x0512_AAAA

Bit	Bit Name	Description
18–0	NEXT_DTIM	The time in 1/8 TU the PCU should wake up and wait for a DTIM Beacon. When this value matches the lsb's of the TSF it is time to wake up and wait for DTIM Beacon. Software should program this value to be slightly ahead of the next TBTT. When this time expires then the SLP_DTIM_PERIOD is added to this value and once again this logic waits for TSF to catch up to this value before waking up again.
19	ASSUME_DTIM	A mode bit which indicates whether we should assume that we missed the beacon when the SLP_BEACON_TIMEOUT occurs with no received beacons. In which case we will assume we missed the DTIM and just wait for CAB.
20	SLEEP_ENABLE	A mode bit which enable the beacon powersave state. The effect is that the TIM/DTIM trigger generated by the expiration of NEXT_TIM/ NEXT_DTIM will be ignored.
23–21	RES	Reserved. Must be written with zero. Contains zeros when read.
31–24	CAB_TIMEOUT	The time in TU that the PCU will wait for CAB after receiving the beacon or the previous CAB. This will insure that if no CAB is received after the beacon is received or if there is a long gap between CAB's, that the CAB powersave state will return to idle.

9.7.31 SLEEP 2 (SLP2)

The Sleep 2 register, in conjunction with the Sleep 1 and Sleep 3 registers, control when the AR5212 should wake when waiting for receive traffic from the AP. These registers are only used when the AR5212 is in STA mode.

Address offset: 0x80D8
Access: Read/Write
Cold reset: 0x020_5555
Warm reset: 0x020_5555

Bit	Bit Name	Description
18–0	NEXT_TIM	The time in 1/8 TU the PCU should wake up and wait for a TIM Beacon. The mechanism for programming and updating is the same as for SLP_NEXT_DTIM.
23–19	RES	Reserved. Must be written with zero. Contains zeros when read.
31–24	BEACON_TIMEOUT	The time in TU that the PCU will wait for Beacon after waking up. If this time expires and we woke due to SLP_NEXT_DTIM and SLP_ASSUME_DTIM is active then assume that we missed the beacon and go directly to watching for CAB. Otherwise when this time expires, the beacon powersave state will return to idle.

9.7.32 SLEEP 3 (SLP3)

The Sleep 3 register, in conjunction with the Sleep 1 and Sleep 2 registers, control when the AR5212 should wake when waiting for receive traffic from the AP. These registers are only used when the AR5212 is in STA mode.

Address offset: 0x80DC
Access: Read/Write
Cold reset: 0x0003_0002
Warm reset: 0x0003_0002

Bit	Bit Name	Description
15–0	SLP_TIM_PERIOD	Time (in TU) between TIM beacon messages.
31–16	SLP_DTIM_PERIOD	Time (in TU) between DTIM beacon messages.

9.7.33 STA Mask Lower 32-bits Register (STAMSKL)

This STA register provides multiple BSSID support when the AR5212 is in AP mode.

Address offset: 0x80E0
Access: Read/Write
Cold reset: 0xFFFF_FFFF
Warm reset: 0xFFFF_FFFF

Bit	Bit Name	Description
31–0	BSS_MASK_L	STA Mask lower 32-bit register. Provides multiple BSSID support.

9.7.34 STA Mask Upper 16-bits Register (STAMSKU)

This STA register provides multiple BSSID support when the AR5212 is in AP mode.

Address offset: 0x80E4
Access: Read/Write
Cold reset: 0xFFFF
Warm reset: 0xFFFF

Bit	Bit Name	Description
15–0	BSS_MASK_U	STA Mask upper 16-bit register.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.7.35 Transmit Power Control Register (TPC)

This register set the transmit power for self-generated response frames.

Address offset: 0x80E8
Access: Read/Write
Cold reset: 0x3F3F3F
Warm reset: 0x3F3F3F

Bit	Bit Name	Description
5–0	ACK_PWR	ACK self-generated response frames.
7–6	RES	Reserved. Must be written with zero. Contains zeros when read.
13–8	CTS_PWR	CTS self-generated response frames.
31–14	RES	Reserved. Must be written with zero. Contains zeros when read.

9.8 Baseband Interface MIB Counters

The Transmit Frame Counter register, in conjunction with the Receive Frame Counter, Receive Clear Counter, and Cycle Counter registers are used to provide software a means of profiling the behavior at the MAC baseband interface for performance analysis. These counters are based on the system clock domain which turns off during sleep. The MIB Control register controls the behavior and will hold the values of these counters (refer to "Management Information Base Control Register (MIBC)" on page 98). The MIBC register clear (CMC) bit will zero out all values of these registers. The MIBC register counter strobe bit (MCS) will increment all the registers each cycle.

These registers do not roll over or saturate, but rather use a time-weighted average to maintain the ratios of these MIB counter registers. When the cycle counter register reaches 32'hFFFFFFFF, all the registers shift their values right by 1. So the new cycle counter register will be 32'h7FFFFFFF. Writes are allowed to verify averaging behavior.

9.8.1 Transmit Frame Counter Register (TFC)

The Transmit Frame Counter register counts the number of cycles the tx_frame signal is active.

Address offset: 0x80EC
Access: Read/Write
Cold reset: 0x0
Warm reset: 0x0

Bit	Bit Name	Description
31–0	TX_FRAME_CNT	Counts the number of cycles the tx_frame signal is active.

9.8.2 Receive Frame Counter Register (RFC)

This register counts the number of cycles the rx_frame signal is active.

Address offset: 0x80F0
Access: Read/Write
Cold reset: 0x0
Warm reset: 0x0

Bit	Bit Name	Description
31–0	RX_FRAME_CNT	Counts the number of cycles the rx_frame signal is active.

9.8.3 Receive Clear Counter Register (RRC)

This register counts the number of cycles the rx_clear signal is active.

Address offset: 0x80F4
Access: Read/Write
Cold reset: 0x0
Warm reset: 0x0

Bit	Bit Name	Description
31–0	RX_CLEAR_CNT	Counts the number of cycles the rx_clear signal is active.

9.8.4 Cycle Counter Register (CC)

This register counts the number of clock cycles.

Address offset: 0x80F8
Access: Read/Write
Cold reset: 0x0
Warm reset: 0x0

Bit	Bit Name	Description
31–0	CYCLE_CNT	Counts the number of clock cycles.

9.8.5 Quiet Time Register 1 (QT1)

The Quiet Time registers 1 and 2 implement the quiet time function specified in the proposed 802.11h extension supporting radar detection.

Address offset: 0x80FC
Access: Read/Write
Cold reset: 0x0002_0000
Warm reset: 0x0002_0000

Bit	Bit Name	Description
15–0	NEXT_QUIET	The time offset in TU's which is compared with the bits [25:10] of the TSF. When these values match, the chip is in “quiet time”. The current value of NEXT_QUIET will be added to the QUIET PERIOD to create the new NEXT_QUIET value. All transmits in progress will be terminated and no new transmit will be sent until the QUIET_DURATION has elapsed.
16	QUIET_ENABLE	Determines whether we are in quiet mode or not. If high then the chip is in quiet time mode which means that it periodically terminates/suspends transmits for a set duration.
17	QUIET_ACK_CTS_ENABLE	
31–18	RES	Reserved. Must be written with zero. Contains zeros when read.

9.8.6 Quiet Time Register 2 (QT2)

The Quiet Time registers 1 and 2 implement the quiet time function specified in the proposed 802.11h extension supporting radar detection.

Address offset: 0x8100
Access: Read/Write
Cold reset: 0x0002_0001
Warm reset: 0x0002_0001

Bit	Bit Name	Description
15–0	QUIET_PERIOD	The length of time in TU's between “quiet time”.
31–16	QUIET_DURATION	The length of time in TU's what the chip is required to be quiet.

9.8.7 TSF Parameters Register (TSF)

This register sets the amount to increment TSF each time the microsecond prescaler counts to zero. (See USEC_32.)

Address offset: 0x8104
Access: Read/Write
Cold reset: Unaffected
Warm reset: Unaffected

Bit	Bit Name	Description
7–0	TSF_INCREMENT	TSF increment count.
31–8	RES	Reserved. Must be written with zero. Contains zeros when read.

9.8.8 QOS No Ack Register (NOACK)

This register provides a mechanism to locate the NoACK information in the QoS field and determine which encoding means NOACK.

Address offset: 0x8108
Access: Read/Write
Cold reset: 0xCC
Warm reset: 0xCC

Bit	Bit Name	Description
3–0	NOACK_2_BIT_VALUES	These are values of a two bit field which indicate “No Ack”. *NOACK_2_BIT_VALUE:* *Encoding matching No ACK:* xxx1 00 xx1x 01 x1xx 10 1xxx 11
6–4	NOACK_BIT_OFFSET	The bit offsets from the byte where the “No Ack” information should be stored. This offset can range from 0 to 6 only.
8–7	NOACK_BYTE_OFFSET	The number of bytes from the byte after end of the header of a data packet to the byte location where the “No Ack” information is stored. (The end of the header is at byte offset 25 for 3 address packets and 31 for 4 address packets.)
31–9	RES	Reserved. Must be written with zero. Contains zeros when read.

9.8.9 PHY Error Mask Register (PHYERR)

This register provides the ability to choose which PHY errors from the baseband will be filtered. The error number is used as an offset into this register. If the mask value at the offset is 0, then this error will be filtered and not show up on the receive queue. Only the first 32 of the 256 possible PHY errors have a mask. All others will be filtered.

Address offset: 0x810C
Access: Read/Write
Cold reset: 0x0
Warm reset: 0x0

Bit	Bit Name	Description
0	ERROR TRANSMIT_UNDERRUN	
3–1	RES	Reserved. Must be written with zero. Contains zeros when read.
4	ERROR PANIC	
5	ERROR RADAR_DETECT	Radar signal.
6	ERROR ABORT	
7	ERROR TX_INTERRUPT_RX	
16–8	RES	Reserved. Must be written with zero. Contains zeros when read.
17	ERROR OFDM TIMING	False detection for OFDM.
18	ERROR OFDM SIGNAL_PARITY	
19	ERROR OFDM RATE_ILLEGAL	
20	ERROR OFDM LENGTH_ILLEGAL	
21	ERROR OFDM POWER_DROP	
22	ERROR OFDM SERVICE	
23	ERROR OFDM RESTART	
24	RES	Reserved. Must be written with zero. Contains zeros when read.
25	ERROR CCK TIMING	False detection for CCK.
26	ERROR CCK HEADER_CRC	
27	ERROR CCK RATE_ILLEGAL	
29–28	RES	Reserved. Must be written with zero. Contains zeros when read.
30	ERROR CCK SERVICE	
31	ERROR CCK RESTART	

9.8.10 ACK SIFS Register (ACKSIFS)

This register accounts for changes in the ACK-SIFS table used for duration calculations to accommodate Turbo mode.

Address offset: 0x8114
Access: Read/Write
Cold reset: 0x0
Warm reset: 0x0

Bit	Bit Name	Description
3–0	ACKSIFS_INCREMENT	Value to be subtracted from ACKSIFS table. Value is in µsec.
31–4	RES	Reserved. Must be written with zero. Contains zeros when read.

9.8.11 Rate to Duration Register (DURS)

This register accounts for changes in the ACK-SIFS table used for duration calculations to accommodate Turbo mode.

Address offset: 0x8700-0x877C
Access: Read/Write
Cold reset: 0x0
Warm reset: 0x0

Bit	Bit Name	Description
15–0	RATE_TO_DURATION	The duration field of data packets depends on the rate since the ACK time will be different based on the changing rates due to multirate. An assumption is made that all stations will follow this same ACK rate rule. The rate data transfer will index into this table and insert the result into the duration field of data packets. There are 32 entries in this table to correspond to the 32 possible rates. The width of the entry is 16 bits to match the width of the duration field in the packet. The duration field is only overwritten if the duration update enable bit in the transmit descriptor is low.
31–16	RES	Reserved. Must be written with zero. Contains zeros when read.

9.9 Key Table

The key table contains 128 entries that include the keys used to encrypt Tx frames and decrypt Rx frames. Each entry contains a variable length key, which is stored along with the key length, the MAC address associated with that key, and a key-valid flag. The key table starts at offset address 0x8800 and each entry occupies eight consecutive 32-bit word addresses. Each eight 32-bit word entry contains the following fields.

9.9.1 TKIP Key

When the key type is 4 (TKIP) and the key is valid, the entry + 64 will contain the Michael key. TKIP keys are not allowed to reside in the entries 64-127 since they require the Michael key. Entries 64-127 are always reserved for Michael. Table 9-9 on page 175 shows the entries for Michael.

32-bit Word	Bit	Description
0	31–0	Key bits 31–0.
1	15–0	Key bits 47–32.
	31–16	Reserved.
2	31–0	Key bits 79–48.
3	15–0	Key bits 95–80.
	31–16	Reserved.
4	31–0	Key bits 127–96.
5	2–0	Key Type: 000 = WEP 40-bit key 001 = WEP 104-bit key 010 = Reserved 011 = WEP 128-bit key 100 = TKIP and Michael 101 = AES/OCB 128-bit key 110 = AES/CCM 128-bit key 111 = Disable cipher
	3	000 = Previous tranmsit was on antenna 1. 001 = Previous transmit was on antenna 2.
	31–4	Reserved.
6	31–0	MAC address bits 32–1.
7	14–0	MAC address bits 47–33.
	15	Key Valid. 0 = Invalid key and MAC address. 1 = Valid key and MAC address.
	31–16	Reserved.

Table 9-9. **TKIP Key Values**

Intrakey Offset	Bits	Description
8*N = 800	31:0	Michael key 0.
8*N = 804	15:0	Reserved.
8*N = 808	31:0	Michael key 1.

Table 9-9. **TKIP Key Values**

Intrakey Offset	Bits	Description
8*N = 80C	15:0	Reserved.
8*N = 810	31:0	Reserved.
8*N = 814	15:0	Reserved.
8*N = 818	31:0	Reserved.
8*N = 81C	14:0	Reserved.
	15	Key valid = 0.

10.Electrical Characteristics

10.1 Absolute Maximum Ratings

Symbol	Parameter	Max. Rating	Unit
V_{DD2_5}	Supply voltage	-0.3 to 3.6	V
V_{DD3_3}	I/O supply voltage	-0.3 to 4.6	V
T_{store}	Storage temperature	-65 to 150	°C
ESD	Electrostatic discharge tolerance	1750	V

Absolute maximum ratings are those values beyond which damage to the device can occur. Functional operation under these conditions, or at any other condition beyond those indicated in the operational sections of this document, is not implied.

10.1.1 Recommended Operating Conditions

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{DD2_5}	Supply voltage	- 5% to 10%[1]	2.375	2.5	2.75	V
V_{DD3_3}	I/O supply voltage	± 10%	3.0	3.3	3.6	V
T_{case}	Case temperature		0	25	85	°C
T_J	Junction temperature		0	50	110	°C
θ_{JA}	Junction to ambient temperature				25	°C/W

[1]The recommended power-on sequence is to have V_{dd3_3} lag V_{dd2_5}

10.1.2 General DC Electrical Characteristics

The following conditions apply to all DC characteristics unless otherwise specified:

V_{dd} = 3.3 V, T_{amb} = 25 °C

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{IH}	High level input voltage	$V_{out} = V_{OH\ (min)}$	2.0	—	V_{dd} + 0.3	V
V_{IL}	Low level input voltage	$V_{out} = V_{OL\ (min)}$	-0.3	—	0.8	V
I_{IL}	Input leakage current	without pull-up or pull-down	—	± 5	—	µA
		with pull-up or pull-down	—	± 65	—	µA
V_{OH}	High level output voltage	No load (I_o = 0)	V_{dd} - 0.3	—	—	V
		I_o = 12 mA	V_{dd} - 0.8	—	—	—
V_{OL}	Low level output voltage	No load (I_o = 0)	—	—	0.20	V
		I_o = 12 mA	—	—	0.27	V
I_O	Output current (for LED and GPIO pins)	V_o = 0 to V_{dd}	—	—	12	mA
C_{IN}	Input capacitance	—	—	6	—	pF

10.1.3 PCI DC Electrical Characteristics

The following conditions apply to all PCI DC characteristics unless otherwise specified:

$V_{dd} = 3.3$ V, $T_{amb} = 25$ °C

Symbol	Parameter	Condition	Min	Max	Units	Notes
V_{IH}	High level input voltage	—	0.5 V_{dd}	V_{dd} + 0.5	V	—
V_{IL}	Low level input voltage	—	-0.5	0.3 V_{dd}	V	—
V_{IPU}	Input pull-up voltage	—	0.7 V_{dd}	—	V	1
I_{IL}	Input leakage current	$0 < V_{in} < V_{dd}$	—	± 10	µA	2
V_{OH}	High level output voltage	I_{out} = -500 µA	0.9 V_{dd}	—	V	—
V_{OL}	Low level output voltage	I_{out} = 1500 µA	—	0.1 V_{dd}	V	—
C_{IN}	Input capacitance	—	—	10	pF	3
C_{CLK}	PCI_CLK pin capacitance	—	5	12	pF	—
C_{IDSEL}	PCI_IDSEL pin capacitance	—	—	8	pF	4
L_{PIN}	Pin inductance	—	—	20	nH	

1. By design. Minimum voltage to which pull-up resistors are calculated to pull a floated network. Applications sensitive to static power utilization must ensure that the input buffer is conducting minimum current at this input voltage.
2. Input leakage currents include hi-Z output leakage for all bi-directional buffers with tri-state outputs.
3. Absolute maximum pin capacitance for a PCI input is 10 pF (except for PCI_CLK).
4. Lower capacitance on this input-only pin allows for nonresistive coupling to PCI_AD[XX].

10.1.4 CardBus DC Electrical Characteristics

The following conditions apply to CardBus DC characteristics unless otherwise specified:

V_{dd} = 3.3 V, T_{amb} = 25 °C

Symbol	Parameter	Condition	Min	Max	Units	Notes
V_{IH}	High level input voltage	—	0.475 V_{dd}	V_{dd} + 0.5	V	—
V_{IL}	Low level input voltage	—	-0.5	0.325 V_{dd}	V	—
V_{ITOH}	Input device turn-off voltage (high)	—	0.7 V_{dd}	—	V	1
V_{ITOL}	Input device turn-off voltage (low)	—	—	0.2 V_{dd}	V	—
I_{IL}	Input leakage current	$0 < V_{in} < V_{dd}$	—	± 10	µA	2
V_{OH}	High level output voltage	I_{out} = -150 µA	0.9 V_{dd}	—	V	—
V_{OL}	Low level output voltage	I_{out} = 700 µA	—	0.1 V_{dd}	V	—
C_{CARD}	Card input pin capacitance	—	5	17	pF	3
C_{CLK}	PCI_CLK pin capacitance	—	10	22	pF	

1. It is the V_{in} value at which current through the input totem pole is essentially shut off.
2. Input leakage currents include hi-Z output leakage for all bi-directional buffers with high-Z outputs.
3. The max value assumes 5 pF for the card trace, 10 pF for the buffer, and 2 pF for the connector and vias. The min value assumes 1 pF for the card trace, 3 pF for the buffer, and 1 pF for the connector and vias.

10.1.5 Power Consumption

These conditions apply to the following typical characteristics unless otherwise specified:

V_{DD2_5} = 2.5 V, V_{DD3_3} = 3.3 V,
AVDD = 2.5 V, T_{amb} = 25 °C

The following tables depict the typical power drain on each of the three on-chip power supply domains as a function of the AR5212's operating mode.

Operating Mode	2.5V Supply (VDD2_5, AVDD, AVDD_PLL)	3.3V Digital Supply (VDD3_3)	Unit
Sleep[1]	3	0.5	mA
Sleep[2]	18	0.5	mA
Idle (Rx)	190	9	mA
5 GHz Tx (802.11a)	120	11	mA
5 GHz Rx (802.11a)	213	9	mA
2.4 GHz Tx (802.11b)	63	14	mA
2.4 GHz Rx (802.11b)	250	6	mA
2.4 GHz Tx (802.11g)	135	12	mA
2.4 GHz Rx (802.11g)	255	8	mA

[1]Sleep mode with 32 kHz clock.
[2]Sleep mode with 32 MHz clock.

Turbo Mode[1]	2.5V Supply (VDD2_5, AVDD, AVDD_PLL)	3.3V Digital Supply (VDD3_3)	Unit
Sleep[2]	3	0.5	mA
Sleep[3]	18	0.5	mA
Idle (Rx)	190	9	mA
5 GHz Turbo Tx (802.11a Turbo)	188	12	mA
5 GHz Turbo Rx (802.11a Turbo)	217	9	mA

[1]Turbo mode refers to the enhanced turbo mode supported by the AR5001 chipset.
[2]Sleep mode with 32 kHz clock.
[3]Sleep mode with 32 MHz clock.

11.AC Specifications

11.1 PCI/CardBus Interface Timing

The AR5212 PCI/CardBus interface supports PC Card 7.1 and PCI 2.3 standards. Refer to the applicable standard for further details.

11.1.1 AC Specifications for 3.3 V Signaling

Table 11-1. **AC Specifications for all PCI/CardBus Signaling Except PCI_CLK**

Symbol	Parameter	Conditions	Min.[1]	Max.[1]	Unit
t_{rcb}	Output rise time	0.2 V_{dd} - 0.6 V_{dd} CardBus mode (PCI_mode = 0)	0.25	1.0	V/ns
t_{rcb}	Output fall time	0.6 V_{dd} - 0.2 V_{dd} CardBus mode (PCI_mode =0)	0.25	1.0	V/ns
$slew_r$	PCI output rise slew rate[2]	0.2 V_{dd} - 0.6 V_{dd} PCI mode (PCI_mode =1)	1	4	V/ns
$slew_f$	PCI output fall slew rate[2]	0.6 V_{dd} - 0.2 V_{dd} PCI mode (PCI_mode =1)	1	4	V/ns
I_{cl}	Low clamp current	$-3<V_{in}<-1$ (both PCI and CardBus)	$-25+(V_{in}+1)/0.015$	—	mA
I_{ch}	High clamp current	$V_{dd}+4>V_{in}>V_{dd}+1$ (both PCI and CardBus)	$25+(V_{in}-V_{dd}-1)/0.015$	—	mA

1. Based on the minimum capacitive load that a driver recognizes (10 pF).
2. The cumulative edge rate across the specified range, rather than the instantaneous rate at any point within the transition range. Rise slew rate does not apply to open drain outputs.

11.1.2 PCI_CLK Specifications

The clock waveform delivered to the AR5212, as measured at the AR5212 input pins must meet the specifications shown in Figure 11-1, Figure 11-2, Table 11-2 and Table 11-3.

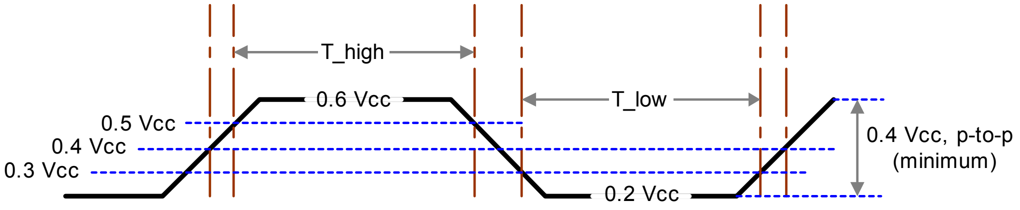


Figure 11-1. **PCI Interface Clock Waveform**

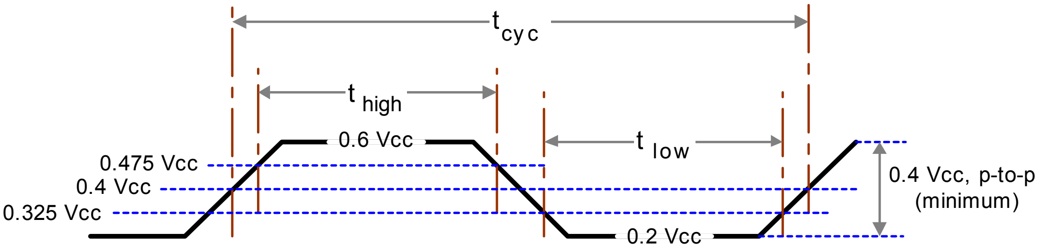


Figure 11-2. **CardBus Interface Clock Waveform**

Table 11-2. **PCI Interface Clock Specifications**

Symbol	Parameter	Min.	Max.	Unit
t_{cyc}	PCI_CLK cycle time for PCI	30	see Note [1]	ns
t_{high}	PCI_CLK high time for PCI	11		ns
t_{low}	PCI_CLK low time for PCI	11		ns
	PCI_CLK slew rate[2] for PCI	1	4	V/ns

[1]The host interface attached to the AR5212 PCI/CardBus interface must operate at frequency of 33 MHz unless otherwise specified.

[2]The slew rate specification must be met across the minimum p-to-p (peak-to-peak) portion of the clock waveform.

Table 11-3. **CardBus Interface Clock Specifications**

Symbol	Parameter	Min.	Max.	Unit
t_{cyc}	PCI_CLK cycle time for CardBus	30	see Note [1]	ns
t_{high}	PCI_CLK high time for CardBus	12		ns
t_{low}	PCI_CLK low time for CardBus	12		ns
	PCI_CLK slew rate[2] for CardBus	1	4	V/ns

1. The host interface attached to the AR5212 PCI/CardBus interface must operate at frequency of 33 MHz unless otherwise specified.
2. The slew rate specification must be met across the minimum p-to-p portion of the clock waveform.

11.1.3 PCI/CardBus Timing Parameters

Table 11-4 and Table 11-5 provides the timing parameters for the AR5212 PC/CardBus interface.

Table 11-4. **PCI Interface Timing Parameters**

Symbol	Parameter	Min.	Max.	Unit
t_{val}	PCI_CLK to signal valid delay[1],[2] - PCI bused signals	—	11	ns
$t_{val(ptp)}$	PCI_CLK to signal valid delay[1],[2] - PCI point to point	—	12	ns
t_{su}	Input setup time to PCI_CLK[1],[2] - PCI bused signals	7	—	ns
$t_{su(ptp)}$	Input setup time to PCI_CLK[1],[2] - PCI point to point	10, 12	—	ns
t_h	Input hold time from PCI_CLK [1]	0	—	ns

1. Refer to the PCI specifications for measurement conditions, assumed 35 pF load.
2. PCI_REQ_L and PCI_GNT_L are point-to-point signals and have different output valid delay and input setup times than bused signals. PCI_GNT_L has a setup of 10; PCI_REQ_L has a setup of 12. All other signals are bused.

Table 11-5. **CardBus Interface Timing Parameters**

Symbol	Parameter	Min.	Max.	Unit
t_{val}	PCI_CLK to signal valid delay [1],[2] for CardBus	—	18	ns
t_{su}	Input setup time to PCI_CLK [2] for CardBus	7	—	ns
t_h	Input hold time from PCI_CLK [2] for CardBus	0	—	ns

1. t_{val} includes the time to propagate data from internal registers to the output buffer, and drive the output to a valid level. Refer to the CardBus specifications for measurement conditions.
2. Times are specified with 35 pF equivalent load.

11.2 External Serial EEPROM Interface Timing

11.2.1 EPRM_CLK Specification

The EPRM_CLK signal to the external EEPROM is output at a rate of 312.5 KHz when the AR5212 is in normal mode. When the AR5212 is in turbo mode, the EPRM_CLK signal is output at 625 KHz.

11.2.2 EEPROM Timing

Figure 11-3 define the timing parameters for the EEPROM interface.

Symbol	Parameter	Min.	Max.	Unit
t_{w_su}	Write data setup time from rising edge of EPRM_CLK	—	1600[1] 800[2]	ns
t_{w_h}	Write data hold time from rising edge of EPRM_CLK	—	1600[1] 800[2]	ns
t_{r_val}	Read data valid time from rising edge of EPRM_CLK	—	400[3], [4]	ns
t_{r_su}	Read data setup time from t_{r_val} to falling edge of EPRM_CLK	1200[1] 400[2]	—	ns
t_{r_h}	Read data hold time from falling edge of EPRM_CLK [3]	—	—	ns

1. AR5212 in normal mode.
2. AR5212 in turbo mode.
3. Value is a function of the EEPROM device selected.
4. The given value is a maximum number for a typical EEPROM. This value reduces the setup time available to the AR5212.

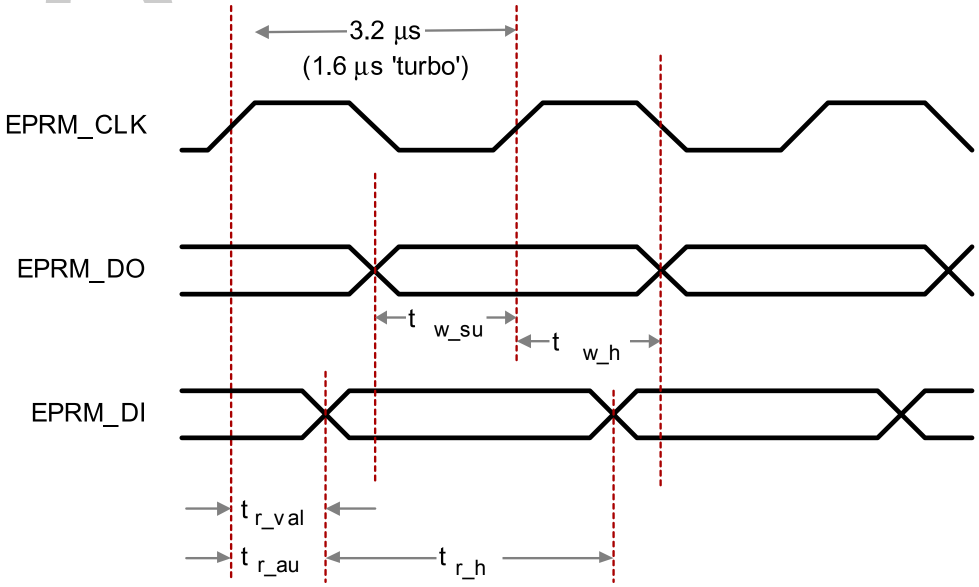
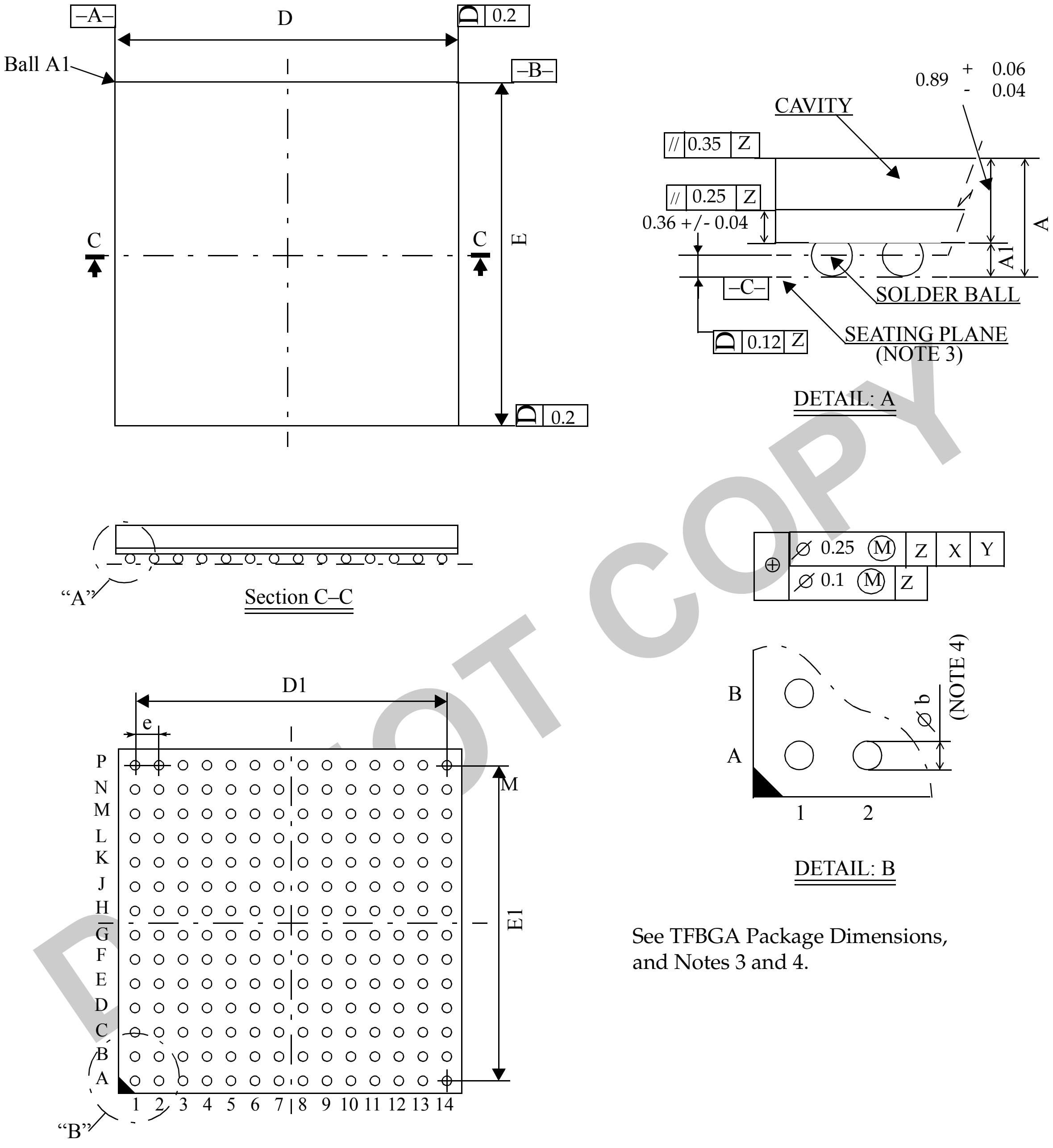


Figure 11-3. **EEPROM Interface Timing**

12. Package Dimensions



See TFBGA Package Dimensions, and Notes 3 and 4.

Figure 12-1. **PBGA Package Drawing**

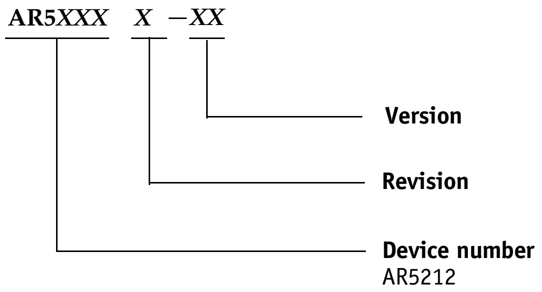
Table 12-1. **PBGA Package Dimensions**

	Measurement in Millimeters			Measurement in Inches		
Dimension Label	**Min.**	**Nom.**	**Max.**	**Min.**	**Nom.**	**Max.**
A	1.30	1.40	1.50	0.051	0.055	0.059
A1	0.30	0.40	0.50	0.012	0.016	0.020
D	14.90	15.00	15.10	0.587	0.591	0.594
E	14.90	15.00	15.10	0.587	0.591	0.594
D1	—	13.00	—	—	0.512	—
E1	—	13.00	—	—	0.512	—
e	—	1.00	—	—	0.039	—
b	0.40	0.50	0.60	0.016	0.020	0.024
MD/ME	14/14			14/14		

1. All dimensions and tolerances conform to ASME Y14.5M–1994.
2. Reference document: JEDEC MO–192.
3. Primary datum c and seating plane are defined by the spherical crowns of the solder balls.
4. Dimension b is measured at the maximum solder ball diameter, parallel to primary datum c.
5. MD is number of balls along D dimension.
   ME is number of balls in E dimension.

Ordering Information

Atheros products are available in a variety of packages and shipping containers. The order number is determined by the selection of these options. An example is shown below.



An order number, *AR5212 A-00* specifies a version of the AR5212 device packaged in a ball grid array (plastic) chip carrier.

Revision History

Revision	Description of Changes
November 2002	Preliminary information.
February 2003	Updated register specifications.
September 2003	Removed “Package B” information from Package Dimensions chapter.

Index

The information in this document has been carefully reviewed and is believed to be accurate. Nonetheless, this document is subject to change without notice. Atheros assumes no responsibility for any inaccuracies that may be contained in this document, and makes no commitment to update or to keep current the contained information, or to notify a person or organization of any updates. Atheros reserves the right to make changes, at any time, in order to improve reliability, function or design and to attempt to supply the best product possible.

Document Number: 981-00012-005

Atheros Communications
Incorporated
529 Almanor Avenue
Sunnyvale, CA 94086
t: 408/773-5200
f: 408/773-9940
www.atheros.com