国立交通大学
电信工程研究所
硕士论文

IEEE 802.11介質存取控制器
設計與實現：
傳輸部分

Design and Implementation of IEEE 802.11
MAC Controller:
Transmitter Part

研究生：林富雄
指導教授：陳伯寧 博士
李程輝 博士

中华民国八十八年六月
IEEE 802.11介质存取控制器设计与实现：
传 輸 部 份
Design and Implementation of IEEE 802.11 MAC Controller:
Transmitter Part

研究生：林富雄
Student : Fu-Shung Lin

指导教授：陳伯寧 博士
李程輝 博士
Advisors : Dr. Po-Ning Chen
Dr. Tsem-Huei Lee

國立交通大學
電信工程研究所
碩士論文

A Thesis
Submitted to Institute of Communication Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University

In Partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Electrical Engineering
June 1999
Hsinchu, Taiwan, Republic of China

中華民國八十八年六月
授權書
（博士論文）

本授權書所授權之論文為本人在國立交通大學電信工程研究所__組
八十七學年度第__學期所撰__學位論文。

論文名稱: IEEE 802.11 介質存取控制器設計與實現: 傳輸部份

同意    不同意

本人具有著作財產權之論文提要，授予國家圖書館、本人畢業學校及行政院國家
科學委員會科學技術資料中心，得重製成電子資料檔後收錄於該單位之網路，並
與台灣學術網路及科技網路連線，得不限地域時間與次數，以光碟或紙本重製發
行。

同意    不同意

本人具有著作財產權之論文全文資料，授予行政院國家科學委員會科學技術資料中
心，得不限地域時間與次數以微縮、光碟重製後發行，並得專享該中心微縮小組製
作之研究報告、獎勵代表作、博士論文三檔資料等值新台幣伍佰元之服務。本
論文因涉及專利等智慧財產權之申請，請將本論文全文延後至民國88年8月後
再公開。

同意    不同意

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性
發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。

指導教授姓名: 陳伯寧博士，李榮輝博士

研究生簽名: ________    學號: 8613551
（親筆正楷）

日期: 民國88年6月4日

備註: 1. 上述同意與不同意之欄立若未勾選，本人同意視同授權。
    2. 授權第二項者，請再交論文一本予承辦人員。
    3. 本授權書已於民國85年4月10日送請書委會修正定稿。
IEEE 802.11介質存取控制器設計與實現：
傳輸部份

學生：林富雄
指導教授：陳伯寧博士
李煥輝博士

國立交通大學電信工程研究所

中文摘要

隨著最近無線通訊技術的進步，能同時支援語音與數據服務的無線傳輸已經是下一代通訊網路的趨勢。無線傳輸不會被環境所限制，經由簡單的(無需佈線的)網路拓樸就可以提供等效於有線網路的服務。可以預期的是，未來的無線網路，將會著重移動性的網路存取支援，以及即時互動服務。

在無線區域網路的相關標準中，IEEE 802.11 規格[3][4]可以說是第一個制定較完整的無線局部數據傳輸標準。它不僅提供了必備的競爭式數據傳輸服務，也制定了可選擇性支援的延遲限制服務(Time-Bounded Service)。IEEE 802.11 在1997年制訂完成後，數百種相關產品即很快地出現在市場上。在這些產品中，IEEE 802.11介質存取控制器大部分都是採用中央處理器為基礎(或是韌體為基礎的)的架構設計，例如 AMD 公司所製造的「以 80188 為基礎」的 79C30 晶片。然而，我們注意到以韌體來實現介質存取控制，實際上對區域網路所需之低成本及高傳輸速度要求，並不是符合經濟效益的設計方法。同時，我們也觀察到 IEEE 802.11 已經制定完成的介質存取控制規範，於接下來的版本並未有任何修訂。所以，韌體實現架構的易隨標準修訂而修改之彈性化的好處，變成已不是主要的 IEEE 802.11 介質存取控制器設計考量。因此，我們認為必須思考一個新的實現架構。

在本論文中，我們提出以專門設計的標準單元式積體電路來實現 IEEE 802.11 介質存取機制(CSMA/CA)。針對「控制傳輸框架」需要快速回應的
需求，一個「控制傳輸框架處理模組」也包含在我們設計的介質存取控制器內。我們的設計---如果與中央處理器為基礎的設計架構作比較---和主系統及基頻處理器的介面就顯的相當簡單。除此之外，我們的設計也能夠很直接的適用於未來 IEEE 802.11 5/11Mbps 高速的版本上[4] [12]。
Design and Implementation of IEEE 802.11 MAC Controller:
Transmitter Part

Student: Fu-Shung Lin  Advisors: Dr. Po-Ning Chen
Dr. Tsern-Huei Lee

Institute of Communication Engineering
National Chiao Tung University

Abstract

With the recent advance of the wireless technologies, wireless transmission is becoming the trend of the next generation communication for voice and data services. It can provide wired-network equivalent services through simple network topology and without being restricted by the environmental space. As anticipated, the future wireless network will focus more on supporting the mobility access, as well as the interactive services.

IEEE 802.11 [3] [4] is perhaps the first well-rounded standard on wireless LAN. It not only specifies the basic contention-based data transmission services, but also optionally supports the time-bounded services. After the standardization of IEEE 802.11 in 1997, hundreds of compliant products quickly fill in the market. The current implementations of the IEEE 802.11 Medium Access Control (MAC) are mostly based on a CPU-Based or firmware-based architecture, such as the 80188-based 79C30 chip manufactured by AMD Inc. We, however, notice that the firmware-based implementation is actually not a cost-effect solution for a low-cost and high-speed LAN product. We also observe that the IEEE 802.11 MAC scheme is already finalized (and expects to remain intact in the coming revision). Therefore, the flexibility of a firmware-based implementation somewhat loses its ground. Accordingly, a new implementation architecture should be considered.

In this thesis, we propose a specially cell-based IC design to implement the IEEE 802.11 basic medium access mechanism (CSMA/CA). To quickly respond to
the control frames, an internal Control Frame Handler is also included in our MAC controller. In our design (if compared with the CPU-based architecture), the interface to the host system, as well as the interface to the Baseband processor, is quite simple. In addition, our design can easily migrate to the coming 5.5/11 Mbps high-speed version of IEEE 802.11 standard [4][12].
致谢

在研究所这两年很荣幸承蒙李程辉教授与陈伯寧教授悉心指导，在研究一的时候接受李程辉教授的指导，使我学习到做研究的道理与知识。研究二的时候，感谢陈伯寧教授的教诲，使我受益非浅，接受到不同领域的知识，还有对我的论文实作与撰写，感谢老师细心的指导，有了老师的帮忙，才能使我的论文顺利完成。

另外，感谢实验室的郭耀文学长热情的指导，解决我在做论文的困难。也很感谢实验室的同学们陪我渡过了两年的欢乐，在问题的讨论上也让我增进了不少的专业知识。

最后，要感谢我的父母对我的养育之恩，提供我充足的生活照顾，也感谢我的女朋友-佳纹，在我的背后支持我，有了你们我才可以无优无虑地从事我的论文研究，谨以此论文来献你们对我的照顾的恩情与感情。
Contents

Chinese Abstract............................................................................................................... i

English Abstract.......................................................................................................... iii

Acknowledgement....................................................................................................... v

Contents ..................................................................................................................... vi

List of Tables ............................................................................................................... xi

List of Figures ............................................................................................................. xii

Chapter 1  Introduction ............................................................................................... 1

  1.1  A BRIEF SUMMARY OF IEEE 802.11 STANDARD ........................................ 1

      1.1.1  Infrastructure network ........................................................................... 2

      1.1.2  Ad Hoc network .................................................................................... 3

  1.2  OBJECTIVE OF OUR STUDY ......................................................................... 4

Chapter 2  Distributed Coordination Function and its Specific Cell-based Design

Architecture ................................................................................................................ 6

  2.1  DISTRIBUTED COORDINATED FUNCTION (DCF) .................................... 6

      2.1.1  CSMA/CA Mechanism .......................................................................... 6

      2.1.2  Random Back-off Mechanism ............................................................... 8

      2.1.3  RTS/CTS mechanism ............................................................................ 9

      2.1.4  Control of the channel .......................................................................... 9

      2.1.5  Synchronization for an Ad Hoc Network ............................................. 10

      2.1.6  Other enhanced mechanisms ............................................................... 11

  2.2  FUNCTIONS IN OUR SPECIFIC CELL-BASED DESIGN MEDIUM ACCESS

      CONTROLLER ................................................................................................. 12
2.3 Architecture of the MAC Controller................................. 14
  2.3.1 CPU-based design.......................................................... 14
  2.3.2 A specific Cell-based design............................................. 15
    2.3.2.1 BIU module............................................................. 16
    2.3.2.2 DMA module.......................................................... 16
    2.3.2.3 CSMA/CA module with Data FIFO.................................. 17
    2.3.2.4 Control Frame Generator module.................................. 17
    2.3.2.5 Power Enable module............................................... 17
    2.3.2.6 SRAM interfacing module......................................... 18
    2.3.2.7 MII baseband processor interface and SEL module.............. 18
  2.3.3 Comparison between CPU-based design and specific cell-based
design............................................................................. 19

2.4 Interfacing to Host System................................................. 20

Chapter 3 Design and Implementation of the Cell-based MAC Controller.....21
  3.1 CSMA/CA Module with Data FIFO........................................ 21
    3.1.1 Transmission Finite State Machine.................................... 22
    3.1.2 DCF Timer Design....................................................... 30
      3.1.2.1 Design of the random back-off timer and the slot timer........ 30
      3.1.2.2 Design of DIFS timer............................................. 31
      3.1.2.3 Design of ACK timer.............................................. 31
      3.1.2.4 Design of CTS timer............................................. 32
      3.1.2.5 Design of AIFM timer............................................ 32
      3.1.2.6 Design of NAV timer............................................ 33
    3.1.3 FIFO Design............................................................ 33
      3.1.3.1 Design of transmission FIFO................................... 33
      3.1.3.2 Design of reception FIFO..................................... 35
  3.2 Design of Bus Interface Unit............................................ 38
  3.3 Power Control Module.................................................... 40
  3.4 Supplementary Modules.................................................. 41

Chapter 4 Verification and Simulation Results..................................42
  4.1 Design Flow........................................................................ 42
  4.2 Simulation Results............................................................. 44
    4.2.1 CSMA/CA Mechanism Simulation...................................... 44
    4.2.2 RTS/CTS Mechanism Simulation...................................... 45
A.4.3  CSMA/CA MODULE Status/Command Registers

A.4.3.1  Interrupt Tx Status registers
A.4.3.2  Interrupt Rx Status registers
A.4.3.3  Tx Configuration registers
A.4.4  CSMA/CA Rx MODULE Fixed Registers

A.4.4.1  TX Demand Register between RX_FSM and Tx_FSM
A.4.4.2  RX_FSM's Error Status Register
A.4.4.3  MAC Configure Register

A.5  UNIT OPERATION

A.5.1  CSMA/CA Submodule
A.5.2  DMA control module
A.5.3  BIU control module
A.5.4  SRAM module

A.5.4.1  PIN DESCRIPTION
A.5.5  Control Frame Generation module

A.5.5.1  Interface between control module and baseband module
A.5.5.2  Interface between control module and CSMA/CA module
A.5.5.3  Control Frame module architecture
A.5.5.4  MAC and PLCP Header generators
A.5.5.5  Control frame module description

Appendix B Timing Diagram

B.1  PCMCIA I/O READ/WRITE TIMING DIAGRAM
B.2  SRAM TIMING DIAGRAM
B.3  TIMING DIAGRAM BETWEEN TRANSMISSION FIFO AND DMA MODULE
B.4  TIMING DIAGRAM BETWEEN RECEPTION FIFO AND DMA MODULE
B.5  TIMING BETWEEN BASEBAND MODULE AND MAC CHIP
B.6  BLOCK DIAGRAM FOR MAC CONTROLLER
B.7  TX_FIFO AND DMA INTERFACE BLOCK DIAGRAM
B.8  RX_FIFO AND DMA INTERFACE BLOCK DIAGRAM
B.9  BIU MODULE AND DMA MODULE INTERFACE BLOCK DIAGRAM
B.10  CSMA/CA WITH Tx_FSM AND CONTROL MODULE INTERFACE BLOCK

B.11  CSMA/CA MODULE AND BASEBAND MODULE INTERFACE BLOCK
DIAGRAM ............................................................................................................. 101

B.12 PIN ARCHITECTURE .................................................................................. 102

Appendix C Compared CSMA/CD and CSMA/CA Scheme................................. 103

Appendix D ........................................................................................................ 104

D.1 DCF TIMING RELATIONSHIPS ................................................................. 104

D.2 IEEE 802.11 MAC FRAME FORMAT ......................................................... 105
## List of Tables

Table 2.1: Comparison between CPU-based and our specific cell-based designs. 19

Table 2.2: Advantages and disadvantages of CPU-based and specific cell-based designs. .................................................................19

Table 3.1: Back-off windows.................................................................31

Table 3.2: Values of time periods in our design. ..................................................32

Table 3.3: Data Rate for PCMCIA release 2.01. .................................................39

Table 4.1: The Complexity for Our Design......................................................52

Table A.1: Address Location for Registers within BIU module. .........................86

Table A.2: Read and Write function for I/O and Attribute Memory. ..................86
List of Figures

Figure 1.1: Structure of IEEE 802.11 Medium Access Control ..........................2

Figure 1.2: IEEE 802.11 infrastructure system topology ...............................3

Figure 1.3: IEEE 802.11 Ad Hoc system topology ........................................3

Figure 1.4: Alternation of PCF and DCF ....................................................4

Figure 2.1: NAV Update Procedure for Virtual Carrier Sense ......................7

Figure 2.2: CSMA/CA and IFS prioritization schemes .................................8

Figure 2.3: Growth of contention window .................................................9

Figure 2.4: RTS/CTS with Fragmented MSDU ..........................................10

Figure 2.5: PLCP Frame Format .............................................................13

Figure 2.6: Architecture for AMD79C30 ..................................................15

Figure 2.7: Architecture for a specific cell-based design of IEEE 802.11 MAC functionality ..........................................................16

Figure 3.1: Architecture of CSMA/CA module with Data FIFO ..................21

Figure 3.2: Flowchart for the Transmission FSM - Initial Process .............24

Figure 3.3: Flowchart for the Transmission FSM - Data Process ................25

Figure 3.4: Flowchart for the Transmission FSM - Data Random Back-off Process 26

Figure 3.5: Flow Chart for the Transmission FSM - Beacon Process ..........27

Figure 3.6: Flowchart for the Transmission FSM - ATIM Process ..............28
Figure 3.7: Flow Chart for the Transmission FSM - RTS/CTS Procedure. ..........29
Figure 3.8: Flowchart for the Transmission FSM – Transmit Process. ..............30
Figure 3.9: Architecture of transmission FIFO. ........................................34
Figure 3.10: Data transfer in the Transmission FIFO. ....................................35
Figure 3.11: Architecture of Reception FIFO ..............................................35
Figure 3.12: Flow Chart for the Reception FIFO Controller. .........................37
Figure 3.13: Architecture of bus interface unit..........................................38
Figure 3.14: Address Decoder Architecture..............................................39
Figure 3.15: SRAM architecture ...................................................................41
Figure 4.1: Design Flow for Verification......................................................42
Figure 4.2: CSMA/CA Mechanism. ..............................................................44
Figure 4.3: CSMA/CA Mechanism. ..............................................................44
Figure 4.4: RTS/CTS Mechanism. ...............................................................45
Figure 4.5: Transmission FIFO with System Clock......................................46
Figure 4.6: Transmission FIFO with Tx_clk...............................................46
Figure 4.7: Transmission FIFO with Ending Frame .....................................47
Figure 4.8: Reception frame for ACK frame. ..............................................47
Figure 4.9: Reception frame for Data frame. ..............................................48
Figure 4.10: Reception frame for CTS frame. ..............................................48
Figure 4.11: Timer Simulation Results. ......................................................49
Figure 4.12: Grabbed Registers.................................................................50
Figure 4.13: BIU Interface Signals.................................................................50
Figure 4.14: BIU Registers .............................................................................51
Figure 4.15: SRAM Simulation for reading processing....................................51
Figure 4.16: Reception FIFO for the FPGA Simulation....................................52
Figure A.1: MAC Chip Architecture...............................................................66
Figure A.2: CSMA/CA Submodule Architecture............................................67
Figure A.3: Data Frame flow chart .................................................................72
Figure A.4: Beacon transmission flow chart ..................................................74
Figure A.5: ATIM transmission flow chart.....................................................75
Figure A.6: RX_FSM....................................................................................76
Figure A.7: Data STATE of RX_FSM.............................................................77
Figure A.8: Control STATE of RX_FSM.........................................................77
Figure A.9: Management STATE of RX_FSM...............................................78
Figure A.10: Broadcast and Multicast STATE of RX_FSM............................79
Figure A.11: TX FIFO Design........................................................................80
Figure A.12: Rx FIFO Unit ............................................................................81
Figure A.13: DMA block diagram .................................................................82
Figure A.14: BIU module ..............................................................................84
Figure A.15: Address decode format .............................................................85
Figure A.16: SRAM (32 K x 8) Partition .........................................................87
Figure A.17: PLCP Frame Generator ..............................................................90
Figure A.18: Control Frame Selection. ................................................................. 91
Figure B.1: I/O Read/Write Timing without Wait State. ........................................ 93
Figure B.2: I/O Read/Write Timing with Wait State. ........................................... 93
Figure B.3: SRAM Write Timing. .................................................................. 94
Figure B.4: SRAM Read Timing. .................................................................. 94
Figure B.5: Timing between Transmission FIFO and DMA Module............... 95
Figure B.6: Timing between Reception FIFO and DMA Module..................... 95
Figure B.7: Timing between Baseband Module and MAC Chip....................... 96
Figure B.8: Block diagram of MAC Controller............................................. 97
Figure B.9: Tx_FIFO and DMA Interface Block Diagram............................. 98
Figure B.10: Rx_FIFO and DMA Interface Block Diagram.......................... 98
Figure B.11: BIU Module and DMA Modules Interface Block Diagram........... 99
Figure B.12: CSMA/CA with Tx_FSM and Control Module Interface Block Diagram. ........................................................................................................... 100
Figure B.13: CSMA/CA module and Baseband Module Interface Block Diagram. 101
Figure B.14: MAC Chip Pin Architecture....................................................... 102
Figure C.1: Compared between CSMA/CD and CSMA/CA Algorithm......... 103
Figure D.1: DCF Timing Relationships.......................................................... 104
Figure D.2: IEEE 802.11 Data Frame Format............................................. 105
Figure D.3: Management Frame Format...................................................... 105
Figure D.4: RTS Frame Format................................................................. 105
Figure D.5: CTS and ACK Frame Format.................................................... 106
Chapter 1  Introduction

The wireless access to networks has drawn great interest recently. Its main advantage is to provide mobile users with communication capability and information access regardless of locations. Among all standards for wireless communications, IEEE 802.11 [3] [4] [11] is perhaps the first well-rounded standard on wireless LAN. It not only specifies the basic contention-based data transmission services, but also optionally supports the time-bounded services.

1.1  A brief summary of IEEE 802.11 standard

The IEEE 802.11 Standard focuses on the physical layer (PHY) and the medium access control layer (MAC). It is designed to transmit data at 1 Mbps and 2 Mbps over the Industry/Science/Medical frequency band (ISM). The coming version [4] [12] is expected to push the data rate up to 5.5 Mbps and 11 Mbps. According to the usages, frames in IEEE 802.11 are divided into three categories: Data frames, Management frames, and Control frames (cf. Appendix D). There are three modulation schemes specified, which are Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), and infrared radio transmission scheme. In the thesis, we consider only the 1/2 Mbps data rate transmission based on the DSSS radio transmission scheme.

As aforementioned, there are two services specified in the MAC standard of IEEE 802.11: one is the compulsory Distributed Coordination Function (DCF), and the other is the optional Point Coordination Function (PCF) as depicted in the Figure 1.1. DCF is a contention-based medium access scheme. It is based on a distributed random access algorithm, known as Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) [8]. An exponentially increasing Random-backoff window and an optional RTS/CTS frame exchange procedure are also included to increase the system throughput and to avoid the hidden node effects [7] [18].

PCF is an optional service that is only workable in the infrastructure WLAN
topology. It uses a Point Coordinator located at the wireless Access Point to determine which station has the right to transmit. The AP maintains a list for the associated stations, and a Round-Robin Polling scheme is suggested to serve the stations in the list in sequence. By properly controlling the number of stations in the Polling list, time-bounded applications can be fulfilled at wireless LAN.

![Diagram of MAC structure]

*Figure 1.1: Structure of IEEE 802.11 Medium Access Control.*

There are two wireless topologies for the IEEE 802.11 Standard: *Infrastructure* network (which forms a Basic Service Set) and *Ad Hoc* network (which constitutes an Independent Basis Service Set).

1.1.1 Infrastructure network

The infrastructure network topology, as shown in Figure 1.2, contains one or more Access Points and Portals that connect the LAN (or Basic Service Set) to a Distribution System (DS). A wireless station is a device which implements the IEEE 802.11 conformant MAC and PHY interfaces to the wireless medium. An Access Point (AP) is an entity that has station functionality and can provide access to the distribution service via the wireless medium (WM) to its associated stations. The Basic Service Set (BSS) is a set of stations controlled by a single coordination function.
In the infrastructure network, the PCF function can be provided through PCF-enabled Access Point for stations within the same BSS. A station can implement only DCF function or optionally support both DCF and PCF functions at the same time within a BSS. In addition, the stations can also transfer data to a station in another BSS through the help of its associated AP. For two stations within the same BSS, they can exchange data either directly to each other or through the help of AP, depending on the system implementation. Besides, Data can be transferred to wired network through the Portal (similarly to the Bridge).

1.1.2 Ad Hoc network

Figure 1.2: IEEE 802.11 infrastructure system topology.

Figure 1.3: IEEE 802.11 Ad Hoc system topology.
As shown in Figure 1.3, an Ad Hoc topology is a temporarily formed network. It is an independent LAN (Independent BSS) for which no DS can be accessed. Each station takes turn to act as a temporary AP which is responsible for issuing the synchronization information. In an Ad Hoc network, it can only support DCF function because there is no fixed AP to handle the Polling Scheme.

According to the standard, DCF and PCF are alternative in time as shown in Figure 1.4. Its period is determined through the parameter: \textit{CPF Repeated interval}. When PCF is ended by issuing a specific management frame (CF_End), DCF resumes its control to the medium.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure14.png}
\caption{Alternation of PCF and DCF.}
\end{figure}

\subsection{Objective of our study}

After the standardization of IEEE 802.11 in 1997, hundreds of compliant products quickly fill in the market. The current implementations of the IEEE 802.11 Medium Access Control (MAC) are mostly based on a \textit{CPU-Based or firmware-based architecture}, such as the 80188-based 79C30 chip manufactured by AMD Inc. We, however, notice that the firmware-based implementation is actually not a cost-effect solution for a low-cost and high-speed LAN product. We also observe that the IEEE 802.11 MAC scheme is already finalized (and expect to remain intact in the coming revision). Therefore, the flexibility of a firmware-based implementation somewhat loses its ground. Accordingly, a new implementation architecture should be considered.

In this thesis, we propose a specially cell-based IC design to implement the
IEEE 802.11 basic medium access mechanism (*CSMA/CA*). In order to quickly respond to the control frames, an internal Control Frame Handler is also included in our MAC controller. In our design (if compared with the CPU-based architecture), the interface to the host system, as well as the interface to the Baseband processor, becomes quite simple. In addition, our design can easily migrate to the coming 5.5/11 Mbps high-speed version of IEEE 802.11 standard [4] [12].
Chapter 2  Distributed Coordination Function and its Specific Cell-based Design Architecture

2.1  Distributed Coordinated Function (DCF)

The IEEE 802.11 MAC standard specifies two functions: Distribution Coordination Function (DCF) and Point Coordination Function (PCF). The former one is the basic access scheme and the latter is an optional access method. In our design, we only consider the implementation regarding the DCF functionality over an Ad Hoc network topology as shown in Figure 1.3. In what follows, we will summarize the CSMA/CA mechanism [8] and some enhancements specified in IEEE 802.11.

2.1.1  CSMA/CA Mechanism

It is known that collision detection is not feasible in the wireless environment. Therefore, in order to reduce the probability of collision, the standard employs the carrier sense with collision avoidance mechanism.

The Carrier sense scheme involves both the MAC and PHY layers. On the MAC layer, in addition to CSMA/CA mechanism, a virtual carrier sense function is also specified to further improve the system throughput. As shown in Figure 2.1, there is a 16-bit duration field (cf. Appendix D) in each frame to indicate the expected frame exchange time between stations. The remaining stations monitor the duration field of every frame on air, and update its Net Allocation Vector (NAV) timer using this value. The NAV timer will decrease as time passed by. Whenever the NAV timer is greater than zero, the media is highly likely to be busy; and hence, there is no need to perform the true carrier sensing on the media. When the NAV timer is reduced to zero, the previous frame exchange procedure should end; and stations, which desire to transmit, can start to contend with the channel. Since during the period of NAV being non-zero, no real carrier sensing is performed, the scheme is therefore named virtual carrier sense.
On the other hand, when performing real carrier sensing, the PHY layer will report a *Clear Channel Assessment* (CCA) signal to the MAC layer [9]. There are three modes in CCA reporting. In the first mode, the channel will be reported busy, if the energy (RSSI) measured by PHY layer is above the ED (Energy Detection) threshold. The second mode performs the conventional Carrier Sense scheme. The third mode is a combination of the first mode and the second mode method, i.e., a channel is reported busy only when both modes 1 and 2 sense the channel busy.

![NAV Update Procedure for Virtual Carrier Sense](image)

*Figure 2.1: NAV Update Procedure for Virtual Carrier Sense.*

The DCF also employs different time offsets to prioritize the frames as shown in Figure 2.2. In general, Control frames, such as RTS, CTS and ACK, should have higher priority for medium usage than the normal Data frames. Accordingly, if a station wants to transmit a Data frame, it should defer by a longer *Distributed IFS* (DIFS) before launching the contention procedure. However, if the frame that is about to transmit by a station is a Control frame, the station will only defer a *Short IFS* (IFS) interval. The current standard suggests that over the DSSS PHY, the SIFS and the DIFS are respectively 10 μs and 50 μs (cf. Appendix D).
2.1.2 Random Back-off Mechanism

Random back-off mechanism is an important part of the CSMA/CA scheme. This mechanism is used when the station detects the channel busy at its first attempt to transmit. By combining the IFS prioritization scheme and the back-off mechanism, the station which has a Data frame to transmit and senses the channel busy will defer a DIFS interval plus a random back-off period (cf. Figure 2.1). The Random back-off mechanism can reduce the probability of collision [13]. The formula for the random back-off period is:

$$\text{Back-off Time} = \text{INT} (\text{CW} \times \text{Random}()) \times \text{Slot-Time},$$

where Random() is a random number drawn from a uniform distribution over the interval[0,1], CW is an integer lied between CWmin and CWmax.

The Back-off time is an integer multiple of the slot time. As shown as in Figure 2.3, the initial value of CW is CWmin (which is 7 in IEEE 802.11 standard), and will be sequentially ascending integer power of 2 minus 1 when a frame fails during transmission. If the number of retransmission is equal to five, the contention window stops increasing and remains the value of the CWmax until the frame is successfully transmitted.

Besides, if the medium is determined to be busy at any time during a Back-off slot, then the back-off procedure is suspended, that is, the back-off timer shall
not decrement for that slot. The medium shall be determined to be idle for the duration of a DIFS period or EIFS period [3] [4], before the back-off procedure is allowed to resumed [3]. Transmission shall start whenever the Back-off Timer reaches zero.

![Graph](image)

**Figure 2.3:** Growth of contention window.

### 2.1.3 RTS/CTS mechanism

In addition to the above two mechanisms, the DCF also uses the RTS/CTS frame exchange procedure to improve the system throughput as shown in Figure 2.1. In order not to highly degrade the system throughput, this scheme is effective only when the length of the Data frame is larger than RTStreshold. In other words, if the length of the Data frame is greater than RTStreshold, the stations will exchange the RTS/CTS Control frames before the transmission of the Data frame. During the procedure, if a station does not receive the CTS frame before the expiration of the CTS timer, it will retransmit the RTS frame (cf. Appendix D).

### 2.1.4 Control of the channel

Each first fragment of the first MSDU (Sequence Number is zero) executes the CSMA/CA mechanism and RTS/CTS frame exchange procedure, however, the latter procedure will not be executed for the MSDUs following the first fragment.
The procedure is shown in Figure 2.4. NAV in the Data and Acknowledgment frames specifies the total duration of the next fragment and acknowledgment. For the fragments of the first MSDU, they use the SIFS following the acknowledgment to reserve the channel for the source station to continue (if necessary) with another fragment. Once the station has contended for the channel, that station shall continue to send these fragments until all fragments of a single MSDU or MMPDU has been sent. When a station has transmitted the first MSDU, it shall defer a DIFS and perform Random back-off procedure to contend the channel with other stations for the remaining MSDUs again. This procedure allows all station to contend the channel with equal probability.

![Figure 2.4: RTS/CTS with Fragmented MSDU.](image)

2.1.5 Synchronization for an Ad Hoc Network

A Timing Synchronous function (TSF) keeps the timers for all STAs in the BSS synchronized. The TSF shall be implemented via a distributed algorithm that shall be performed by all members of the BSS, based on the following mechanism [4].

1. Suspend the decrementing of the back-off timer for any pending non-beacon or non-ATIM transmission.
2. Calculate a random delay uniformly distributed in the range between zero and twice a \( CW_{min} \times a\text{SlotTime} \).
3. Wait for the period and decrementing the timer as for back-off.
4. If a beacon arrives before the random delay has expired, the remaining timer
and the current beacon transmission are canceled.

5. If the random delay has expired and no beacon has arrived during the period, send a beacon.

All members of the IBSS participate in beacon generation and each STA shall maintain its own TSF timer. Besides, Each STA in an IBSS shall adopt the timing received from any beacon or probe response [4] that has a TSF value later than its own TSF timer.

2.1.6 Other enhanced mechanisms

The DCF also contains the frame recovery and the acknowledge procedure for unicast frames. If a frame is transmitted but the station does not receive the ACK frame before the expiration of the ACK timer, the station has to retransmit the previous frame. In such a retransmission mechanism, duplicated frames could be received. Therefore, in order to remove the duplications, a station needs to maintain a list of previous received frames with information of the Source Address, Frame Sequence number, and Fragment number. Searching for duplications in the list will be launched when the retry bit of the currently received frame is set (cf. Appendix D).

When the broadcast/multicast frames are transferred from a STA with a ToDS bit clear (cf. Appendix D), only the basic access scheme shall be used and no RTS/CTS and ACK mechanisms shall be used. Any broadcast/multicast frames transferred from a STA with a ToDS bit set, in addition to the basic access procedure of CSMA/CA, obey the RTS/CTS and ACK mechanisms, because the MPDU is directed to the AP. Besides, there is no MAC-level recovery on broadcast/multicast frames, except for those frames sent with the ToDS bit set.

Finally, the power management scheme for the STAs within an IBSS depends on the following mechanism. When an MSDU is to be transmitted a destination STA that is in a PS mode, the transmitting STA first transmits an ATIM frame during the ATIM window, in which all the STAs including those operating in a PS mode are awake [3][4]. The Back-off mechanism, ACK procedure, and Retry
procedure for ATIM frames are similar to normal Data frames. The Contention Window is from zero to 255 and the ATIM window starts at each TBTT with 4K \( \mu \text{s} \) size (default value).

### 2.2 Functions in our specific cell-based design Medium Access Controller

In this thesis, the features for our design, corresponding to the current IEEE 802.11 standard, are listed below:

1. 1/2 Mbps half-duplex data transmission in an Ad Hoc network;
2. IEEE 802.11 CSMA/CA mechanism;
3. Random back-off procedure;
4. RTS/CTS mechanism;
5. Broadcast and multicast MPDU procedure;
6. Power management mode;
7. Direct (or on-chip) response to Control frames;

In our design, the Control frames are directly handled by the MAC controller; and the Data frames and the Management frames, such as the Beacon frames and Probe frames, are processed by the Host driver or application.

According to the IEEE 802.11, the preamble and PLCP header should always be transmitted at 1 Mbps data rate [3]. However, the MAC header and the frame body can be transmitted at either 1 Mbps or 2 Mbps, which is indicated by the Signal field within the PCLP header as shown in Figure 2.5. In the IEEE 802.11 standard, the transmission is half-duplex, i.e., a station cannot transmit and receive at the same time. When a station finishes its transmission (or reception), it will be automatically back to the CCA state; and stay in this state until the reception of the next trigger signal to activate it to enter either transmission or reception state.
The target network topology of our design is the Independent Basic Service Set (IBSS), which incorporates a temporary access point (AP) selected through a contention-based algorithm, and all the stations involved in IBSS perform the CSMA/CA mechanism to contend with the media. The contention-based algorithm for AP selection is also a random back-off algorithm, but its back-off window is twice of the Minimum Contention Window (CWmin), and does not increase exponentially as the random back-off algorithm for CSMA/CA.

To summarize, the IEEE 802.11 CSMA/CA mechanism is a typical CSMA/CA algorithm with exponentially increasing back-off windows enforced by an additional Interframe Space Scheme to prioritize transmitted frames. The range of the back-off window lies between 0 and 255 slot times. Also included in the IEEE 802.11 standard is (1) the NAV function (or virtual carrier sensing) to reserve the media for transmission, (2) a MAC-level Acknowledgement and (3) RTS/CTS mechanism. All the above will be considered and implemented in our design.

Note that for the broadcast or multicast frames, the RTS/CTS mechanism and the MAC-level acknowledgement will be disabled according to the IEEE 802.11.

Power management mode is also a concern in our design, since power consumption is one of the important factors for the battery-supported wireless products. In an Ad Hoc network, a station needs to maintain an ATIM (Ad Hoc
Traffic Indication Message) window, which is about 4 ms in length according to the MIB attribute of the Standard. By using the ATIM window, the power saving stations know exactly when to enter the Doze state (or the Awake state.) In addition, our MAC controller will issue a specific pin to control the on-and-off of the Baseband/RF/IF components so that the power consumption is economized.

We close the subsection by pointing out that the Control frames, such as RTS/CTS/ACK, are directly handled by the MAC controller to speed up the response time in the MAC layer, and to ease the burden of the host driver.

## 2.3 Architecture of the MAC controller

### 2.3.1 CPU-based design

Most of the currently WLAN products employs a CPU-based implementation in MAC layer. A famous example is the AMD79C30 MAC controller. Its architecture is depicted in Figure 2.6. The AMD79C30 contains three parts: Bus Interface Unit (BIU), an embedded 80188 core, and the Transceiver Attachment Interface (TAI).

The BIU module supports either of two common interfaces: PCMCLA and ISA. The choice of this interface is determined by a “PCMCLA pin” [2]. The configuration registers in the BIU module are decoded by the 80188 core.

The embedded 80188 core provides the basic processing capability for firmware implementation of the IEEE 802.11 MAC functionality. It controls two external memory components: one SRAM and one Flash memory, which share a common memory interface. The SRAM is basically for temporary data storage, and the flash memory is used to store the MAC firmware code.

The TAI module contains the Tx/Rx FIFOs, Tx/Rx CRC32 generator, Data scrambler, and interfaces to baseband processor [2]. It provides the necessary functionality to directly connect to a variety of transceiver interface styles. In addition, there are several 64-byte registers [2] in the TAI module for the
configuration of operational parameters, commands, temporary data storage, and status reporting.

2.3.2 A specific Cell-based design

In our design, similar modules to the TAI module and the BIU module of AMD79C30 are included. The 80188 core, however, is replaced by a pure cell-based design for IEEE 802.11 MAC. Our proposed architecture for the IEEE 802.11 MAC functionality is illustrated in Figure 2.7. As shown in the figure, it includes the Bus Interface Unit module (BIU), Direct Memory Access Controller module (DMA), CSMA/CA module with Data FIFO, Control Frame Generator module, and the Power enable module. In addition to these, the MAC controller provides an internal interface to a 32K×8 SRAM for the temporary storage of the transmitted frames and the received frames. Simple descriptions on the functionality of these modules are listed below. Detailed designs will be provided in Chapter 3.
2.3.2.1 BIU module

Our current implementation of the BIU module only supports the PCMCIA release version 2.0 [14] [15]. It is an asynchronous interface with no timing source or clock. Some proper control lines control the handshake between the host and the adapter. For instance, the chip enable, the write enable, and the read enable control the read/write of the data bus. One key issue in this interface is that the address line must be stable before the control signals. Also in the BIU module, decoders (cf. Appendix A) for the decoding of the address pins of the SRAM and the configuration registers are implemented.

2.3.2.2 DMA module

The DMA module handles the data access of the BIU and the CSMA/CA module. It arbitrates the directions of the transmitted frames, which includes (1) BIU to SRAM, (2) SRAM to CSMA/CA module for transmission, (3) SRAM to BIU, and (4) the CSMA/CA module to SRAM for reception.
2.3.2.3 CSMA/CA module with Data FIFO

The CSMA/CA module implements the basic access scheme of IEEE 802.11. Its functionality includes the DCF procedure, such as the carrier sense mechanism, random back-off algorithm, IFS interval calculation, MAC-level acknowledge timer, RTS/CTS mechanism and its associated timer, and the IEEE 802.11 state machine for the Transmit, Receive and CCA states. Moreover, we use a 48-byte Transmit FIFO and a 1-byte Receive FIFO to buffer the frames inside MAC controller for necessary usage. Some information should be extracted or grabbed into the Grabbed Register from the frame header in order to take the proper action on the frame. This information and their format can be found in Appendix A.

2.3.2.4 Control Frame Generator module

The module generates the IEEE 802.11 CTS and ACK Control frames in response to RTS and Data frames. It also initiate the RTS frames when the length of the transmitted Data frame is longer than RTS threshold. As a result, some fields in the frame header require the information from CSMA/CA module and BIU module (such as the Transmitter address (TA) and the Power Management Mode of the current transmission frame), as well as the information from the received frame (such as the Receiver address). In order to generate a complete Control frame, the attachment of the preamble and PCLP header is also performed in this module.

2.3.2.5 Power Enable module

The power management in the design of this thesis is achieved by a single controlled signal. Its status is decided by the statuses of the BIU and the CSMA/CA modules. The baseband/IF/RF components can therefore ramp down when the signal becomes inactive, which corresponds to Doze state in IEEE 802.11 standard. In order to have a more elaborate power management scheme in which the ramp-up time of each component is taken into consideration, more power management signals should be provided. This will be deferred as the future
work in this thesis.

2.3.2.6 SRAM interfacing module

The maximum frame size for the IEEE 802.11 WLAN is 2346 bytes. For simplicity, we partition the 32K×8 SRAM into 12 equal-size segments. Thus, each segment is 2.5K bytes in its size, which is larger than 2345 bytes. These 12 segments are further divided into two regions: 2 transmit segments and 10 receive segments. One of the two transmit segments is dedicated to the transmitted Data frame; and the other is reserved for the transmitted Management frame. Similarly, the first nine receive segments are particularly for the received Data frames; and the remaining one is reserved for the received Management frame. The reason for separating two kinds of segments is that the priority of Management frames is higher than Data frames. And the next received Management frame can be filled onto the SRAM right now. Therefore, we only allocate a segment for Management frame.

2.3.2.7 MII baseband processor interface and SEL module

The interface between the Baseband module and the MAC module follows the 4-bit-wide MII specification. A simple SEL module is implemented for selecting the correct data path for the Data/Management frames and the Control frames. Its basic rule is that if the CSMA/CA module is active, the SEL module will allow the Data/Management frames to flow from the FIFO in the CSMA/CA module to the baseband processor; otherwise, it will let the Control frames from the Control generator module to enter the MII bus which connects to the baseband processor.
2.3.3 Comparison between CPU-based design and specific cell-based design

<table>
<thead>
<tr>
<th></th>
<th>CPU-based design</th>
<th>Specific cell-based design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin counts</td>
<td>144</td>
<td>147</td>
</tr>
<tr>
<td>MAC functionality implementation</td>
<td>Microprocessor (80188)</td>
<td>Cell-based IC</td>
</tr>
<tr>
<td>Baseband interface</td>
<td>serial data</td>
<td>4-bit MII parallel data</td>
</tr>
<tr>
<td>Host interface</td>
<td>PCMCIA or ISA</td>
<td>PCMCIA</td>
</tr>
<tr>
<td>External memory bus interface</td>
<td>SRAM and Flash memory</td>
<td>SRAM and EEPROM</td>
</tr>
</tbody>
</table>

*Table 2.1: Comparison between CPU-based and our specific cell-based designs.*

<table>
<thead>
<tr>
<th></th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific cell-based design</td>
<td>Simple</td>
<td>Non-flexible to Standard revision</td>
</tr>
<tr>
<td></td>
<td>High speed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Easy for joint simulation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>with baseband/RF/IF, as well</td>
<td></td>
</tr>
<tr>
<td></td>
<td>as system integration</td>
<td></td>
</tr>
<tr>
<td>CPU-based design</td>
<td>Flexible</td>
<td>Complex and non-cost effective</td>
</tr>
</tbody>
</table>

*Table 2.2: Advantages and disadvantages of CPU-based and specific cell-based designs.*
2.4 Interfacing to host system

Since the handling of the Management frames are not included in our MAC controller, the processing of the management frames therefore has to be done by the host driver. We would like to point out that some of the Management frames (and hence, their associated functions) can actually be handled internally by the MAC controller. But we defer their implementations as the future work. The discussion on the interfacing to host system in this section is based on the current implementation.

First, for the Data and the Management frames, the CRC16 field within the PLCP header is calculated by the host driver; however, the CRC32 field of the MAC header is performed by the MAC controller, since it takes more time in its computation.

Second, the TSF timer is maintained at the host driver. The host driver will update the TSF timer after receiving a Beacon frame. As a consequence, when the station becomes a temporary Access Point in an Ad Hoc WLAN, the TSF timer field of its transmitted Beacon frame will be provided by the host driver, not the MAC controller.

Third, the configuration registers for the MAC controller should certainly be configured through the host driver. For simplicity, we fix the I/O ports that are used by the host driver to access these configuration registers, which are 280H, 281H, and 282H. An extension to programmable I/O port numbers should be easy in implementation.

Finally, the length of the transmitted frames shall be provided by the host driver through writing the information into the proper configuration register. Likewise, the length of the received frames will be placed at the first two-byte in each 2.5K-byte receive segment of the SRAM so that the host driver can setup the length counter at the beginning of uploading the receive frames in the SRAM.
Chapter 3  Design and Implementation of the Cell-based MAC Controller

In this chapter, we will provide the implementation details. There are five modules in our MAC controller as shown in Figure 2.7. This thesis is responsible for the design and implement action of three modules: BIU module, Power control module, and the CSMA/CA module. The implementations of other modules are addressed in [20].

3.1  CSMA/CA Module with Data FIFO

This module implements the Distributed Coordination Function of the IEEE 802.11, and provides necessary interfacing signals to the baseband processor [9]. Its architecture is depicted in Figure 3.1. As shown in the figure, we divide the module into three units: Transmission Finite State Machine, DCF timer, and FIFO.

![Architecture of CSMA/CA module with Data FIFO.](image)

*Figure 3.1: Architecture of CSMA/CA module with Data FIFO.*
3.1.1 Transmission Finite State Machine

A common approach in the implementation of a MAC protocol is to interpret it as a finite state machine. The flow charts for the Transmission Finite State Machine are depicted in Figure 3.2. The Initial State is the *Idle State*, which corresponds to the *CCA State* in [3, pp. 202]. When a transmission request (*Tx_Request*) from the BIU module is received, the frame desired to be transmitted is then downloaded onto the proper transmit segment of the SRAM, followed by the access of the first few bytes of the transmitted frame into the Transmit FIFO. The next state of the FSM is determined after necessary header information is fetched into the Grabbed Register.

If the current frame is not the first fragment in an MSDU (MAC Service Data Unit), the FSM will directly enter the *Transmission State*, and transmission of the current frame will follow. (No deferral of SIFS is needed here because SIFS, by definition, is the shortest time for a station to respond to a received frame.) Otherwise, the module distinguishes the transmitted frame type between the Data frame type and Management frame type.

If the transmitted frame is a Data frame, a further check on the retry bit of the Data frame will be executed. According to the standard, a retry frame shall launch the Random back-off procedure without DIFS deferral. In case that the frame is not a retry one, deferring of DIFS will begin. Note that since a non-retry frame should always defer a DIFS time before its transmission or commencement of its backoff procedure, its access priority to channel is statistically lower than a retry one.

Whenever the channel is busy within the DIFS timer, the FSM will mark the “Busy_frame” bit and enter the *Idle State*. When a “DIFS_end” signal is received which indicates the channel is free for a consecutive DIFS time, the FSM will enter the *Back-off State*. Upon the expiration of a back-off timer, check on the necessity of RTS/CTS procedure will be performed.

A multicast or broadcast frame does not initiate the RTS/CTS procedure in
the Ad Hoc Network. In addition, if the frame length of a unicast frame is shorter than RTSThreshold, the RTS/CTS procedure will be disabled as well. Besides, a CTS timer is implemented to time the period between the transmission of the RTS frame and the reception of a returned CTS frame as shown in Figure 3.7. If the CTS timer is expired, we will increase the Contention Window exponentially and enable the Control Module to re-transmit the RTS frame.

On the other hand, if the transmitted frame is a Management Frame, it will be distinguished among Beacon frames, ATIM frames and other type of management frames. Different random back-off window sizes should be used for the Beacon frames and ATIM frames. Other type of management frames will be concluded into the same back-off procedure as Data frames to simplify our design. When the back-off timer has reached zero, the FSM will enter the Transmission State. Finally, if a “Tx_complete” signal is received after a frame is transmitted, the FSM will reset the MAC controller, go back to the Idle State, and wait for the next trigger.

The random back-off number for the above three kinds of back-off windows (i.e., back-off window for Beacon frames, back-off window for ATIM frames, and back-off window for data frames and other types of management frames) is generated in terms of a common Automatic Sequence Counter (ASC). Whenever the channel is busy (CCA=1), a Flag will be enabled and the decrement process of the back-off timer will be stopped (by locking the timer). At the channel-idle period, FSM will unlock the Random Back-off timer by enabling the Flag and timer will continue to decrease. Upon the expiration of the back-off timer, both the Flag and registers regarding the Random timers will be reset. It need to be pointed out that the back-off timer will load different window size according the current status, such as Flag, as shown in Table 3.1.
Figure 3.2: Flowchart for the Transmission FSM - Initial Process.
Figure 3.3: Flowchart for the Transmission FSM - Data Process.
Figure 3.4: Flowchart for the Transmission FSM - Data Random Back-off Process
Figure 3.5: Flow Chart for the Transmission FSM - Beacon Process.
Figure 3.6: Flowchart for the Transmission FSM - ATIM Process.
Figure 3.7: Flow Chart for the Transmission FSM - RTS/CTS Procedure.
3.1.2 DCF Timer Design

According to [3], there are lots of timers required from the MIB attributes [17]. In this thesis, we focus on those timers that are pertinent to the DCF scheme. These are the Random back-off timer, slot timer, DIFS timer, ACK timer, CTS timer, ATIM timer, and the NAV timer. In our design, we use the system clock (44Mhz) provided by the Baseband processor as our basis to these timers.

3.1.2.1 Design of the random back-off timer and the slot timer

The Random back-off module contains two parts. The first part deals with the calculation of a slot time. As suggested by the Standard, a slot time is 20 us for the DSSS scheme. Thus, it requires a 10-bit counter for 44MHz system clock (880 count). The second part is the random back-off timer, which decrease by 1 per slot time (cf. Figure 2.3). As illustrated in Table 3.1, we need to deal with three kinds of back-off windows. These are done through a common sequential counter, which will keep counting up according to the system clock, starting from the power-up of the system. The random number with a proper rounding up due to
the necessary contention windows will be taken from the counter at the beginning of each random back-off procedure. Note that the randomness of the back-off number can be further enhanced by using a common maximum-length shift register counter.

<table>
<thead>
<tr>
<th>type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contention Window for Data frame</td>
<td>CWmin – CWmax (7 - 255)</td>
</tr>
<tr>
<td>Contention Window for Beacon frame</td>
<td>Zero – 2×CWmin (0 – 14)</td>
</tr>
<tr>
<td>Contention Window for ATIM frame</td>
<td>Zero – CWmax (0 – 255)</td>
</tr>
</tbody>
</table>

*Table 3.1: Back-off windows.*

3.1.2.2 Design of DIFS timer

The DIFS time is equal to two slot times and one SIFS time by the standard. However, since the SIFS time is actually the necessary processing time for the Physical and MAC layer, plus the air propagation time and the power ramp-on time, the system will naturally defer SIFS when processing. Hence, we can legitimately neglect this time. Accordingly, the actual deferral time for our DIFS process is 40 us. For convenience, we implement it as a 12-bit counter for 44MHz system clock.

3.1.2.3 Design of ACK timer

Unicast data frames and ATIM management frames need a MAC-level Acknowledgement. According to the standard, the ACK control frame should be returned within 10 us plus the transmission time of an ACK frame. The transmission time depends on the Signal field in the PLCP header of the previous received frame. Its value is depicted in Table 3.2. If the source station receives no ACK frame within the ACK time, the Transmission Status Register of the MAC controller will send an interrupt signal to the host so that the host can set the retry bit in the MAC header of the retransmission frame.
3.1.2.4 Design of CTS timer

If the length of the current frame is larger than the RTS-threshold value configured via the BIU module, the FSM will enable the Control module to transmit RTS. And the CTS timer starts to increase from zero, after the RTS is transmitted. The value of the CTS time is shown in Table 3.2. If the CTS timer is timeout, the FSM will enter the back-off state to proceed the random back-off procedure before the next RTS transmission in order to resolve the collision of the first RTS.

3.1.2.5 Design of ATIM timer

This module is implemented for stations with power-saved capability in an Ad Hoc network. In the initial state, host will configure the default value of the ATIM window in the BIU module. The ATIM timer will use the value as the counting basis if receiving an enable signal in the FSM. When this timer reached the limit, the module will output a power enable signal by justifying the configured power-save bit, and the input control signals, such as Beacon-transmitted and ATIM-transmitted (cf. Appendix A).

<table>
<thead>
<tr>
<th>type</th>
<th>timer value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot Time</td>
<td>20 us</td>
</tr>
<tr>
<td>DIFS Time</td>
<td>2*aSlotTime</td>
</tr>
<tr>
<td>ACK Time</td>
<td>10 us plus {112 us for 1Mbps} or {56 us for 2 Mbps}</td>
</tr>
<tr>
<td>CTS Time</td>
<td>2<em>10 us plus 2</em>aSlotTime plus {112us for 1Mbps} or {56 us for 2Mbps}</td>
</tr>
<tr>
<td>ATIM Time</td>
<td>ATIM-Window * 44000</td>
</tr>
</tbody>
</table>

Table 3.2: Values of time periods in our design.
3.1.2.6 Design of NAV timer

When a frame is received, its duration field will be extracted. The time unit of the duration filed is microsecond. Therefore, a proper adjustment from the value of the duration field to the actual count is necessary. The conversion formula is shown below:

\[
\text{NAV counter} = (\text{Rx\_MAC\_Header}[31:16] \times 44) \text{ for } 1\text{Mbps} \\
\text{NAV counter} = (\text{Rx\_MAC\_Header}[31:16] \times 88) \text{ for } 2\text{Mbps}
\]

In our design, the Baseband processor only sends the MAC body to the MAC controller. Hence, the counting process of the NAV timer begins after receiving the completely MPDU and when the receiving MPDU is not addressed to its. Besides, updating the NAV timer will be only when the new NAV value is greater than the current NAV value. This flow chart is shown in Figure 3.12.

3.1.3 FIFO Design

As shown in Figure 3.1, the data path is a FIFO. A FIFO controller is also implemented to handle the data flow. The data may flows to two different directions. One is the transmission path to Baseband processor, and the other is the reception path to the external SRAM. Both directions employ the MII Interface Specification. The data bus of MII interface is 4-bit wide. Therefore, the transmission clock, as well as the reception clock, becomes 500KHz and 250KHz from 2Mbps and 1 Mbps data rates. Details are described below.

3.1.3.1 Design of transmission FIFO

Figure 3.9 depicts the architecture for the 48-byte Transmission FIFO. It contains three parts: Transmission FIFO controller, Transmission FIFO counter, and the Data path. The Transmission FIFO controller is responsible for the communication with the DMA module for the transmission “Process” and with the CSMA/CA module for the “initial” transmission signal. If host has downloaded a frame into the SRAM, the DMA will send the Tx_fifo signal to the
FIFO controller. The reason for using a 48-byte FIFO is that we must store the information from the Preamble field up to the Sequential Control Field within the MAC header so that the other modules can use the information. The FSM will fetch some fields of these 48 bytes into the grabbed registers. Meanwhile, if the CSMA/CA procedure is not completed, the Tx_start_fsm is disabled and the transmission FIFO controller will send a fifo_stop to the DMA module to stop the loading of the next 48-byte data. Upon the end of a CSMA/CA procedure, the Transmission FIFO controller will enable the data path to transmit the next 48-byte data into the FIFO at the time the previous 48-byte data is transmitted.

![Diagram of transmission FIFO architecture](image)

**Figure 3.9:** Architecture of transmission FIFO.

On the other hand, there are two counters in the FIFO module. One is used to count the 48-byte loaded data according to the system clock (44 MHz) in response to the Tx_fifo signal. The other is used to count the total data flowed through the FIFO based on the transmission clock (500KHz or 250KHz) in response to the Tx_start_fsm signal. An additional signal is also implemented, which is constantly changed with the transmission clock to indicate to the baseband processor whether the current data is the most significant 4-bit (Tx_Data_fifo[7:4]) or the least significant 4-bit (Tx_Data_fifo[3:0]).
The signals from Transmission FIFO controller and Transmission FIFO counter are pointers pointing to the Data path. The Data path includes a 48-byte FIFO and a Grabbed register. The FIFO is 8-bit wide, but the output bus is only 4-bit in width. The detailed contents of the Grabbed registers are shown in the Appendix A.

Our design also takes the advantage of the high clock difference between the system clock (44MHz) and the transmission clock (250KHz and 500KHz). As shown in Figure 3.10, at the time that the last 4-bit data is transmitted using the slow transmission clock, we can quickly move the next 48-byte data from the SRAM to the FIFO upon the high-rate system clock. This trick facilitates the design of the FIFO module.

![Figure 3.10: Data transfer in the Transmission FIFO.](image)

### 3.1.3.2 Design of reception FIFO

![Figure 3.11: Architecture of Reception FIFO.](image)
The architecture of the Reception FIFO (cf. Figure 3.11) is quite similar to the Transmission FIFO. It also contains three parts: Reception FIFO controller, Reception FIFO counter, and the Data path. The difference is that the Reception Data path is only one byte in length, and hence, only one counter is required. Since the system clock that is used to load data from the FIFO to the SRAM is much smaller than the reception clock, one-byte should be sufficient. When the Rx_ready is set, the Reception FIFO controller will begin the reception of a frame. The controller will then distinguish the coming frame among Data, Management and Control frames. If a Control frame is received, it will be processed internally, and will not be loaded onto the SRAM. The data or Management frames will be moved to the SRAM directly. Figure 3.12 is the flowchart of the Reception FIFO controller.

The controller will also send enable signals to the other modules. For example, when the Rx_counter is equal to five, NAV timer will load the duration field to the NAV counter and the FSM_en will activate the Reception FSM. Finally, upon the end of reception of a frame, the controller will signal a Rx_end to DMA module, reset itself, and NAV timer will justify whether the NAV timer needs to decrease or not by these information: Rx_end, Address1, and CRC32_error, as shown in Figure 3.12. When the receiving frame reaches the ending byte, the NAV timer will use the Rx_end to start its Finite State Machine, use the Address 1 to justify whether the received frame is its or not, and use the CRC32_error from Rx_FSM to detect whether the frame is valid or not. Besides, the NAV timer will update the current NAV value by the new received value if the new value is larger than the old one. Then, the NAV timer will decrease the NAV counter by system clock. Finally, the NAV timer will indicate the channel status to “Idle” after the counter becomes zero.

The Counter provides the index of the frame, i.e., which byte of the received frame is currently receiving. Part of the MAC header will also be stored into a 24-byte register, which provided necessary information to Reception FSM and reception CRC module.
Figure 3.12: Flow Chart for the Reception FIFO Controller.
3.2 Design of bus interface unit

The bus interface to the host is the PCMCIA [14] [15]. Its architecture is illustrated in Figure 3.13, which includes an Address decoder, a control logic function, and the configuration registers.

![Diagram of bus interface unit]

**Figure 3.13:** Architecture of bus interface unit.

The function of the address decoder is shown in Figure 3.14. The address bus employed is 26 bits in length. For simplicity, the I/O port addresses for the MAC controller are fixed, which are 280H, 281H and 282H. The 280H and 281H I/O ports are dedicated to the least significant byte [7:0] and the most significant byte [15:8] of the address for SRAM data access and registers. The 282H I/O port is used for data input and output to the host. These data include those contained in SRAM, CIS and configuration registers of the MAC controller. As depicted in Figure 3.14, if bit 16 of the address is set, the data though port 282H is for SRAM; otherwise, it is for the CIS and the Configuration registers. Furthermore, the address [15:0] of the CIS is ranged from zero to 01FFH, and the address [15:0] of the Configuration registers is ranged from 200H to FFFFH. Details can be found in the Appendix A.
Figure 3.14: Address Decoder Architecture.

The main function of the control logic is to deal with the signals between the PCMCIA interface and the DMA module, as well as CSMA/CA module. The forward path is to transfer the signals from either DMA module or CSMA/CA module to the host via PCMCIA. The reverse path is just the opposite of the forward path.

The supported data rates for the PCMCIA asynchronous standard are listed in Table 3.3 (cf. The PCMCIA Timing Diagram in Appendix B). These data rates are calculated based on the Address Setup/Hold time plus the Read/Write time [15]. The high data rates of PCMCIA CardBus Standard can be found in [14], which uses a 33MHz clock to transfer the data. The resultant data rates of CardBus Standard can reach 133Mbps.

<table>
<thead>
<tr>
<th>Type</th>
<th>Memory Card</th>
<th>I/O Card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Access</td>
<td>20 MB/sec</td>
<td>3.92 MB/sec</td>
</tr>
<tr>
<td>Byte Access</td>
<td>10MB/sec</td>
<td>7.84 MB/sec</td>
</tr>
</tbody>
</table>

Table 3.3: Data Rate for PCMCIA release 2.01.
3.3 Power Control Module

In this module, we generate a power enable signal to control the on-and-off of the other components on WLAN card. There are three factors that will affect the decision of the power enabling. First, if a station transmits a Beacon frame in an Ad Hoc network, it will keep awake until the end of the next TBTT. During this period, the FSM will activate the power enable signal. Secondly, if the third bit of the Configuration Status Register (CSR) at the BIU module is set, the power enable signal will become one. Finally, if a station receives an ATIM frame or transmits its ATIM frame [19], it will be awake until the end of the next TBTT.
3.4 Supplementary Modules

In this thesis, we focus on the design of BIU module, CSMA/CA module, and the Power control module. Other necessary modules, i.e., the DMA module and the Control Frame Generation module, are described in [20]. In addition, as shown in Figure 2.7, a SEL logic, as well as the design of partition scheme of the SRAM buffer, is also considered. The SEL logic is a logic function that selects one of the three sources of Data as the source to Baseband processor. The three sources are the Transmission FIFO, CRC32 and the Control Frame generation module. The SRAM is presumed to be 32K X 8 in size. The partition of SRAM for data storage is illustrated in Figure 3.15.

![Diagram](image)

*Figure 3.15: SRAM architecture.*
Chapter 4  Verification and Simulation Results

4.1 Design Flow

In this thesis, we use ASIC to verify our proposed architecture. The design flow is shown in Figure 4.1. The first task is to define the Specification for our MAC controller. The design specification is placed in in Appendix A. Based on the design specification, a “Verilog” code [21][23][24] is written. The “Synopsys” tool [22] is then used to synthesize the verilog code for gate-level simulation.

The detailed relation between these tools is also shown in Figure 4.1. First, for the behavior code, we use the verilog-XL to simulate its functions. Then, we transfer the behavior code to a netlist by means of the Synopsys tool and simulate it again based on the Compass Cell Library. This is called the Soft-Verification flow.

![Design Flow for Verification](image)

Figure 4.1: Design Flow for Verification.
Second, a hard-verification based on FPGA [25][26] is employed. As shown in Figure 4.1, we use an FPGA compiler come with the Synopsys tool to transfer our gate-level code to its FPGA netlist. The FPGA verification software is from Altera. We then use the Altera Max-Plus II to download the netlist into the Altera FLEX10K FPGA device. Finally, we design a demo-board, including a Baseband emulator, to verify our functions through communications with the host driver and the Baseband emulator.
4.2 Simulation Results

4.2.1 CSMA/CA Mechanism Simulation

Figure 4.2: CSMA/CA Mechanism.

Figure 4.2 shows a simulation result of IEEE 802.11 basic access mechanism. In this simulation, a station in the initial state desires to transmit a frame. It therefore waits for a DIFS time. The “CCA” signal, however, becomes active within this time, and hence, the station should enter the Rx_state. This simulation also confirm the situation that after the expiration of DIFS defer time, the station will execute the random timer (state = 07), and continue the follow-up transmission steps.

Figure 4.3: CSMA/CA Mechanism.
On the other hand, Figure 4.3 shows another algorithm within the CSMA/CA mechanism. If the station receives no ACK frame when the Data has been transmitted, it will retransmit the previous frame. At this time, the contention window will become 0f (HEX) in stead of the 07 (HEX) and the station will retransmit its frame (state is equal to 0c) right now without executing RTS/CTS mechanism.

4.2.2 RTS/CTS Mechanism Simulation

Figure 4.4 is the simulation result for the RTS/CTS mechanism. The simulation scenario is described as follows. When we at the state 0b (HEX), we are transmitting the RTS frames. For the first RTS transmission, if no CTS frame is received within the CTS timer, the system will start up the random back-off timer (state = 07). Upon the expiration of the back-off timer at which time the second RTS is about to transmit, an erroneous CTS frame, however, is received. We will re-execute the CSMA/CA mechanism as shown in Figure 4.4.

![Image of simulation result](image_url)

Figure 4.4: RTS/CTS Mechanism.

4.2.3 Transmission FIFO Simulation

Figure 4.5 simulates the FIFO transmission behavior that comes with the system clock. If the signal “Tx_fifo” becomes active, the FIFO will receive Data, as well as fifo_length, from DMA (D_Tx[7:0]). The fifo_length in this simulation
is 30H. Figures 4.6 and 4.7 simulate the transmission behavior that comes with the *Transmission clock*. In Figure 4.7, the signal “tx_end” is active, indicating the end of the transmission.

*Figure 4.5:* Transmission FIFO with System Clock.

*Figure 4.6:* Transmission FIFO with Tx_clk.
4.2.4 Reception FIFO Simulation

Figure 4.8: Reception frame for ACK frame.

Figure 4.8 is the ACK frame and we can see the subtype [7:4] is the 4’b1101 by the Standard. If we have received the duration field in the fifth byte
(rx_frame_cnt is equal to 005), the reception controller will enable the signal “NAV_en” shown in this Figure. Besides, for the 14-bytes ACK frame, if we receive the ending byte, we will enable the FSM_en, Length_counted, and the Rx_end. Figure 4.9 is for the Data frame and Figure 4.10 is for the CTS frame. The above control signals are similar to the ACK frame.

**Figure 4.9:** Reception frame for Data frame.

**Figure 4.10:** Reception frame for CTS frame.

4.2.5 Timer Simulation
Figure 4.11 shows the simulations of ACK timer, the NAV timer, and the CTS timer. When a frame is received, the signal “NAV_end” will become low; otherwise, it will remain (1). Also, if a RTS frame is transmitted, the signal “CTS_flag” will remain high until either CTS frame is received or the CTS timer is timeout. Similarly, the signal “ACK_flag” will remain high after a frame transmission until either an ACK frame is received or the ACK timer is timeout.

![Figure 4.11: Timer Simulation Results.](image)

4.2.6 Grabbed Registers

The grabbed registers fetch some fields within the frame headers, including the PLCP header and the MAC header. Figure 4.12 illustrates the situation of transmitting a retransmission frame (by \texttt{retry\_reg} = 1) whose fragment number (\texttt{frag\_reg}) is zero, and whose signal field (\texttt{Data\_rate\_Tx}) is \texttt{8'b0000_0010} (1 Mbps).
4.2.7 BIU Module Simulation

Our PCMCIA interface consumes three I/O addresses \( A[25:0] \): 280H, 281H, and 282H (HEX format). As shown in Figure 4.13, we use CE1, OE, and REG signals to access the data from SRAM.

In addition, the BIU module also contains the Configuration Registers regarding to the Source Address \( SA[47:0] \), RTSthreshold \( RTSt_{h_{-}reg} \), Transmission Data length \( Data\_length\_w[15:0] \) etc as shown in Figure 4.14.
4.2.8 SRAM interface simulation

We use the oe, CSB, WEB, and a[14:0] to control the Data bus between the MAC chip and the SRAM. Figure 4.15 shows that the MAC chip reads the data from the SRAM after the host places the data into the SRAM, followed by the transfer of the data to Tx_FIFO module. io[7:0] and a[14:0] are respectively the data bus and the address bus.

Figure 4.15: SRAM Simulation for reading processing.
4.2.9 FPGA Simulation

Figure 4.16: Reception FIFO for the FPGA Simulation.

Figure 4.16 simulates the situation that an ACK frame is received at 1 Mbps data rate. As a result, “Signal[1]” is one and Length is 112 us. Besides, Rx_Data[3:0] is transferred into D_Rx[7:0].

4.3 Complexity

By the Synopsys and the FPGA tools, we conclude the clock rate and the gate counts of our design in Table 4.1. It needs to be pointed out that the basic unit of the gate counts for the Synopsys is four transistors; however, the basic unit for the FPGA is in fact the logic cell. From Table 4.1, the maximum system clock of our system is 50 MHz for the Compass Library, and 41.6 MHz for the FPGA Library.

<table>
<thead>
<tr>
<th>Tools</th>
<th>Gate counts</th>
<th>Clock rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synopsys</td>
<td>27,000 gates</td>
<td>20ns</td>
</tr>
<tr>
<td>FPGA</td>
<td>4600 logic cells &amp;</td>
<td>24ns</td>
</tr>
<tr>
<td></td>
<td>1500 FFs</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: The Complexity for Our Design.
4.4 Problems and Solutions for Our Design

Based on our experience through these simulations and implementations, we found that there are potentially some additional improvements that can be made to our current design. First, in our current design, we take the advantage of the large speed difference between the system clock and the transmission clock (or reception clock) to implement our FIFO (cf. Figure 3.10). However, if such speed difference becomes small, our approach may become infeasible. In such case, a “Double Buffer” implementation approach could be a good choice instead. It could also be beneficial for the oncoming high-speed version of IEEE 802.11.

On the other hand, the SRAM controller is in fact the bottleneck for the processing speed of our MAC chip. Its clock rate is upper-bounded by 50MHz. This is because the chosen external memory requires twice of 10ns for the Address extension time. Therefore, we do not have enough address setup time. One solution is to choose the SRAM with smaller access time.
Chapter 5  Conclusions and Future Work

In this thesis, we implement the MAC layer functionality of the IEEE 802.11. The major implemented features include the full IEEE 802.11 multiple access scheme (CSMA/CA Algorithm) and the Control frames Handler for Ad Hoc networks. Simulates of our Verilog code, as well as its transferred FPGA counterpart, are also performed, followed by a hardware verification through a specially designed circuit.

Our experience on a cell-based design of the IEEE 802.11 MAC layer functionality proves that it is a feasible alternative in implementation. The cell-based design approach facilitates the joint simulations with other chips, such as the baseband processor and RF/IF chipsets, and hence, can easily converge to the System On a Chip (SOC) trend. In this thesis, we actually provide a complete solution for a MAC controller since the interface modules (PCMCIA) to host system (PCMCIA) and to baseband processors (MII), as well as DMA modules, are also implemented.

Some functions of IEEE 802.11 that can possibly be incorporated into the MAC controller are deferred to future work. The first one is the Synchronous Procedure, such as the Probe frame (Probe Request and Probe Response frames) for the initial network setup. Secondly, the Beacon frames, as well as their associated TSF timer and/or TIM timer, can be generated/maintained by the MAC controller (in this case, the host driver only needs to prepare the data, and hence, release almost all the burdens of maintaining the host driver). Finally, the power management function can be enhanced to minimize the power consumption of the WLAN adapter.
Bibliography


Appendix A

Medium Access Control Chip & Bus
Interface Unit Module Spec
A.1 Introduction

A.1.1 Synopsis of this spec

This Specification specifies the functionality of the MAC chip module, DMA module, and Bus Interface Unit (BIU) module. The data used by this chip is presumed to be stored in an external 32K SRAM.

A.1.2 Document Revision History

<table>
<thead>
<tr>
<th>DATE</th>
<th>REV #</th>
<th>WHO</th>
<th>DESCRIPTION OF CHANGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>98/08/17</td>
<td>1.0</td>
<td>M.-T. Hong &amp; F.-S. LIN</td>
<td>802.11 Tx State Machine</td>
</tr>
<tr>
<td>98/09/18</td>
<td>1.1</td>
<td>M.-T. Hong &amp; F.-S. LIN</td>
<td>FIFO design in CSMA/CA module</td>
</tr>
<tr>
<td>98/11/10</td>
<td>1.2</td>
<td>M.-T. Hong &amp; F.-S. LIN</td>
<td>Syntax error type updates</td>
</tr>
<tr>
<td>98/12/31</td>
<td>1.3</td>
<td>M.-T. Hong &amp; F.-S. LIN</td>
<td>Modify the Spec’s content</td>
</tr>
<tr>
<td>99/04/06</td>
<td>1.4</td>
<td>M.-T. Hong &amp; F.-S. LIN</td>
<td>Modify the Spec’s content</td>
</tr>
<tr>
<td>99/04/22</td>
<td>1.5</td>
<td>M.-T. Hong &amp; F.-S. LIN</td>
<td>Modify the Spec’s content</td>
</tr>
<tr>
<td>99/05/24</td>
<td>1.6</td>
<td>M.-T. Hong &amp; F.-S. LIN</td>
<td>Modify the Spec’s content</td>
</tr>
</tbody>
</table>
A.2 System Level Description

A.2.1 Feature Set

1. BIU module acts as a data access interface between Host system and LAN card.

2. MAC module, consisting of CSMA/CA submodule, and Control submodule and Power Control submodule, is the implementation of the IEEE 802.11, such as the random back-off mechanism and generation of IEEE 802.11 Control Frames.

3. The data exchange scheme between Host system and LAN card is using the Direct Memory Access (DMA) which consists of one part: local DMA.

A.2.1.1 features

• Supports 1 Mbps or 2 Mbps half duplex data transmission

• Supports IEEE 802.11 CSMA/CA function (see Content 5th)

• Supports IEEE 802.11 Power Management

• Supports Ad Hoc Network (IBSS)

• PCMCIA interface (see Content 5th)

• Fast response for Control frames (see Content 5th)

• Uses a 32K x 8 SRAM for data storage (see Content 5th)
A.2.2 System Level Description

The MAC module is placed after the BIU (PCMCIA) module and before the Baseband module as shown in Figure A.1.

A.2.3 Software Issues

When the Host system wants to transmit Data/Management frames, it shall enable the BIU module (certainly, after the LAN card is initialized), and transmit the Data/Management frames to the 32KB SRAM on the LAN card through the BIU module. The MAC module will then access the Data/Management frames, and transmit them to Baseband module.

According to our design, the Host system does not have to transmit the Control frames before transmitting the Data/Management frames, which is required by the IEEE 802.11 Standardization (such as RTS frame). This is because in our design, the MAC module itself will add the necessary Control frames before transmitting the Data/Management frames. In the other hand, the Host system needs to prepare the CRC16 field in PLCP Field of the transmission frames and the LAN card prepares the CRC32 field in the MAC format.

If some errors occur during the transmission of the MAC module, the MAC module will put the necessary status code into the Status/Command registers, and set the Interrupt register (Active high). The Host system, after informed by Interrupt from MAC module, should read the Status registers and respond with necessary error recovery actions (The error recovery actions of the Host system is beyond the scope of the MAC chip design).

The Host system shall assign reasonable ATIM Window, Data Length, Source Address and RTSThreshold to the corresponding registers of the MAC module, which will be referred by the operations of the MAC module.

As suggested by IEEE 802.11 Specification, the Management frame should have higher priority than the Data frames. Accordingly, if there are Management
frame and Data frame in the SRAM at the same time, the MAC module will transmit the Management frame.
A.3 Interfacing Notes

A.3.1 CSMA/CA and control interface

The internal connections between the CSMA/CA submodule and the Control submodule are as follows.

<table>
<thead>
<tr>
<th>Pin</th>
<th>I/O</th>
<th>Bus Width</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>En_Tx</td>
<td>O</td>
<td>1</td>
<td>Enable Control submodule to read the TX frame status registers</td>
</tr>
<tr>
<td>En_Rx</td>
<td>O</td>
<td>1</td>
<td>Enable control submodule to read the RX frame status registers</td>
</tr>
<tr>
<td>RTS_confirm</td>
<td>I</td>
<td>1</td>
<td>Control submodule complete RTS transmission</td>
</tr>
<tr>
<td>Ctrl_retry</td>
<td>O</td>
<td>1</td>
<td>Set to 1, if transmission failed.</td>
</tr>
<tr>
<td>Add_1_Tx</td>
<td>O</td>
<td>48</td>
<td>Receiver address</td>
</tr>
<tr>
<td>Add_2_Tx</td>
<td>O</td>
<td>48</td>
<td>Source address</td>
</tr>
<tr>
<td>Power_manage</td>
<td>O</td>
<td>1</td>
<td>Power management bit: one = PS state, and zero = Awaken state</td>
</tr>
<tr>
<td>Data_rate_Tx</td>
<td>O</td>
<td>1</td>
<td>0 for 1 Mbps, 1 for 2 Mbps</td>
</tr>
<tr>
<td>Ctrl_subtype_Tx</td>
<td>O</td>
<td>4</td>
<td>Data frame subfield in MAC header: RTS, CTS, ACK</td>
</tr>
</tbody>
</table>

A.3.2 CSMA/CA and Baseband interface

The interface between the CSMA/CA submodule and the Baseband module are as follows.

<table>
<thead>
<tr>
<th>Pin name</th>
<th>Bus width</th>
<th>I/O type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tx_Data</td>
<td>4</td>
<td>O</td>
<td>Data pins. 4 bits, if using MII.</td>
</tr>
<tr>
<td>Tx_clk</td>
<td>1</td>
<td>I</td>
<td>250KHz or 500KHz</td>
</tr>
<tr>
<td>Rx_Data</td>
<td>4</td>
<td>I</td>
<td>Data pins. 4 bits, if using MII.</td>
</tr>
<tr>
<td>Rx_clk</td>
<td>1</td>
<td>I</td>
<td>250KHz or 500KHz</td>
</tr>
<tr>
<td>Length</td>
<td>16</td>
<td>I</td>
<td>44 Mhz, indicate the current MPDU length</td>
</tr>
<tr>
<td>Signal</td>
<td>8</td>
<td>I</td>
<td>44 Mhz, indicate the current MPDU data rate</td>
</tr>
<tr>
<td>Service</td>
<td>8</td>
<td>I</td>
<td>44 Mhz, reserved</td>
</tr>
<tr>
<td>Pin name</td>
<td>Bus width</td>
<td>I/O type</td>
<td>Function</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>----------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>IORD_</td>
<td>1</td>
<td>I</td>
<td>Read enable for I/O function, active low</td>
</tr>
<tr>
<td>IOWR_</td>
<td>1</td>
<td>I</td>
<td>Write enable for I/O function, active low</td>
</tr>
<tr>
<td>CE1_</td>
<td>1</td>
<td>I</td>
<td>Chip enable for 8 bits data bus, active low</td>
</tr>
<tr>
<td>CE2_</td>
<td>1</td>
<td>I</td>
<td>Chip enable for 16 bits data bus, active low</td>
</tr>
<tr>
<td>OE_</td>
<td>1</td>
<td>I</td>
<td>Output enable for attribute memory, active low</td>
</tr>
<tr>
<td>WE_</td>
<td>1</td>
<td>I</td>
<td>Write enable for attribute memory, active low</td>
</tr>
<tr>
<td>REG_</td>
<td>1</td>
<td>I</td>
<td>Enable for attribute memory, active low</td>
</tr>
<tr>
<td>Reset</td>
<td>1</td>
<td>I</td>
<td>Reset pin, active high</td>
</tr>
<tr>
<td>A</td>
<td>26</td>
<td>I</td>
<td>Address bus for data</td>
</tr>
<tr>
<td>TOIS16_</td>
<td>1</td>
<td>O</td>
<td>Input or output is 16 bits, active low</td>
</tr>
<tr>
<td>IREQ_</td>
<td>1</td>
<td>O</td>
<td>Interrupt request to software, active low</td>
</tr>
<tr>
<td>INPACK_</td>
<td>1</td>
<td>O</td>
<td>No use</td>
</tr>
<tr>
<td>STSCHG_</td>
<td>1</td>
<td>O</td>
<td>No use</td>
</tr>
<tr>
<td>SPKR_</td>
<td>1</td>
<td>O</td>
<td>No use</td>
</tr>
<tr>
<td>Wait_</td>
<td>1</td>
<td>O</td>
<td>Wait the MAC chip to deal with work for the Host system</td>
</tr>
<tr>
<td>D</td>
<td>8</td>
<td>I/O</td>
<td>Data Bus</td>
</tr>
</tbody>
</table>
A.3.4 SRAM Interface

Shown in the section A.5.4.1.

A.3.5 BIU module and CIS ROM interface

<table>
<thead>
<tr>
<th>Pin name</th>
<th>Bus width</th>
<th>I/O type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address_CIS</td>
<td>15</td>
<td>I</td>
<td>Address bus</td>
</tr>
<tr>
<td>D_CIS</td>
<td>8</td>
<td>I/O</td>
<td>Data bus</td>
</tr>
<tr>
<td>CE_CIS</td>
<td>1</td>
<td>I</td>
<td>Chip enable for CIS, active low</td>
</tr>
<tr>
<td>OE_CIS</td>
<td>1</td>
<td>I</td>
<td>Read enable for CIS, active low</td>
</tr>
<tr>
<td>WE_CIS</td>
<td>1</td>
<td>I</td>
<td>Write enable for CIS, active low</td>
</tr>
</tbody>
</table>
A.4 Theory of operation

A.4.1 MAC MODULE Block Diagram in IBSS Network

Based on the above diagram, we will describe the block functionality in details below.

BIU module acts as an interface between Host system and LAN card. It not only contains the card initialization information but also enables the MAC controller.

DMA controller handles when and how the frames transmit from SRAM to the FIFO. It uses a handshaking operation with FIFO control when transmitting the frames. DMA controller also justifies whether or not there is any frame in SRAM waiting for transmission. If so, the DMA controller will inform the CSMA/CA submodule by sending a control signal to it.
Local memory is assumed to be a 32K X 8 SRAM module. It is partitioned into two parts: Transmitting part and Receiving part, which respectively contains 2 and 10 sections, where each section is 2.5 K bytes in size. Note that the maximal size of the IEEE 802.11 Frames is 2346 bytes, which is less than 2.5 K bytes.

CSMA/CA submodule performs the IEEE 802.11 CSMA/CA mechanism. It contains many timers, such as the IFS timer, random back-off timer and ATIM timer. Its functional block diagram is shown in Figure A.2.

![Figure A.2: CSMA/CA Submodule Architecture.](image)

### A.4.2 CSMA/CA Tx MODULE Fixed Registers

<table>
<thead>
<tr>
<th>Name</th>
<th>Bits</th>
<th>R/W</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal</td>
<td>8 bits</td>
<td>R/W</td>
<td>CTS/ACK timer</td>
</tr>
<tr>
<td>Unicast or M/B</td>
<td>1 bit</td>
<td>R/W</td>
<td>Unicast or Broadcast/multicast</td>
</tr>
<tr>
<td>Ctrl_subtype_Tx</td>
<td>6 bits</td>
<td>R/W</td>
<td>100000: Data frame</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>001000: Beacon frame</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>001001: ATIM frame</td>
</tr>
<tr>
<td>Retry</td>
<td>1 bit</td>
<td>R/W</td>
<td>0: First frame, 1: Retry frame</td>
</tr>
<tr>
<td>Power_manage</td>
<td>1 bit</td>
<td>R/W</td>
<td>0: Active mode, 1: Power save mode</td>
</tr>
<tr>
<td>Add_1_Tx</td>
<td>48 bits</td>
<td>R/W</td>
<td>Provide for control module</td>
</tr>
</tbody>
</table>
### A.4.3 CSMA/CA MODULE Status/Command Registers

#### A.4.3.1 Interrupt Tx Status registers

<table>
<thead>
<tr>
<th>Name</th>
<th>Bits</th>
<th>R/W</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Txend</td>
<td>2 bits</td>
<td>R/W</td>
<td>Indicate that the Beacon, ATIM frames was successfully transmitted to Baseband module. 01: Beacon frame, 10: ATIM frame</td>
</tr>
<tr>
<td>ATIMend</td>
<td>1 bit</td>
<td>R/W</td>
<td>Notify the Host system to stop transmitting the ATIM frames.</td>
</tr>
<tr>
<td>Data_ACK_to</td>
<td>1 bit</td>
<td>R/W</td>
<td>If the data ACKtimer is timeout, set to one. Note that ACKtimer limit depends on the current data rate (1 or 2 Mbps).</td>
</tr>
<tr>
<td>Magn_ACK_to</td>
<td>1 bit</td>
<td>R/W</td>
<td>If the magn ACKtimer is timeout, set to one. Note that ACKtimer limit depends on the current data rate (1 or 2 Mbps).</td>
</tr>
</tbody>
</table>

#### A.4.3.2 Interrupt Rx Status registers

<table>
<thead>
<tr>
<th>Name</th>
<th>Bits</th>
<th>R/W</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rxddata</td>
<td>1 bit</td>
<td>R/W</td>
<td>Indicate that a Data frame was received</td>
</tr>
<tr>
<td>Rxmagn</td>
<td>1 bit</td>
<td>R/W</td>
<td>Indicate that a Management frame was received</td>
</tr>
<tr>
<td>ACKrx</td>
<td>1 bit</td>
<td>R/W</td>
<td>Indicate that a ACK frame was received</td>
</tr>
<tr>
<td>Beaconrx</td>
<td>1 bit</td>
<td>R/W</td>
<td>Indicate that a beacon frame was received</td>
</tr>
</tbody>
</table>

#### A.4.3.3 Tx Configuration registers

<table>
<thead>
<tr>
<th>Name</th>
<th>Bits</th>
<th>R/W</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATIMtimer</td>
<td>3 bits</td>
<td>R/W</td>
<td>It is used only in IBSS network under power save mode.</td>
</tr>
<tr>
<td>PS</td>
<td>1 bit</td>
<td>R/W</td>
<td>Power save mode, 0: active mode, 1: power save mode.</td>
</tr>
<tr>
<td>RTSThreshold</td>
<td>13 bits</td>
<td>R/W</td>
<td>If the length of the Data frame is larger than RTSThreshold, the MAC module will launch the</td>
</tr>
<tr>
<td>Name</td>
<td>Bits</td>
<td>R/W</td>
<td>Function</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>-----</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ATIM_received</td>
<td>1</td>
<td>R/W</td>
<td>Inform TX_FSM about receiving ATIM</td>
</tr>
<tr>
<td>CTS_received</td>
<td>1</td>
<td>R/W</td>
<td>Inform TX_FSM about receiving CTS</td>
</tr>
<tr>
<td>ACK_received</td>
<td>1</td>
<td>R/W</td>
<td>Inform TX_FSM about receiving ACK</td>
</tr>
<tr>
<td>Beacon_received</td>
<td>1</td>
<td>R/W</td>
<td>Inform TX_FSM about receiving Beacon</td>
</tr>
</tbody>
</table>

### A.4.4.1 TX Demand Register between RX_FSM and Tx_FSM

### A.4.4.2 RX_FSM's Error Status Register
### A.4.4.3 MAC Configure Register

<table>
<thead>
<tr>
<th>Name</th>
<th>Bus</th>
<th>R/W</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board_test_mode</td>
<td>1</td>
<td>R/W</td>
<td>Convenient to do on_board test</td>
</tr>
<tr>
<td>Override_mode</td>
<td>1</td>
<td>R/W</td>
<td>Ignore overall packet error; For monitoring LAN purpose</td>
</tr>
<tr>
<td>BSSID</td>
<td>48</td>
<td>R/W</td>
<td>Indicating current attached BSS</td>
</tr>
<tr>
<td>SA</td>
<td>48</td>
<td>R/W</td>
<td>The LAN card number</td>
</tr>
<tr>
<td>Power_management</td>
<td>1</td>
<td>R/W</td>
<td>Current STA's power mode</td>
</tr>
</tbody>
</table>
A.5 Unit Operation

A.5.1 CSMA/CA Submodule

The function of each unit in CSMA/CA submodule is described below.

1. 802.11 mechanism unit is the most important one in performing the IEEE 802.11 protocol (CSMA/CA). This unit is roughly divided into four state machines: Idle, CCA, TX, and RX. When this unit is in operation, it will obtain information from other units, such as Random timer unit and DIFS Timer unit. In addition, this unit uses the PS (Power Save Mode) to decide whether to enter the Power save mode or not.

2. ATIM counter is used when the station is in power save mode. As specified in IEEE 802.11, the PS station will maintain the ATIM timer during the period between Beacons. During the ATIM time period, every station will awake, and check whether or not there is a frame for itself. If there is any, the PS station will keep awake to receive those frames. If the PS station wants to transmit data, it will first transmit the ATIM frame to contend for the transmission right during the ATIM timer period. In summary, the ATIM counter unit is to ensure that every station will awake at the right time, and the transmission frames will not be lost.

3. DIFS Timer is to count the DIFS. It is about 50 us in length. Similarly, Slot time unit is used to count the slot time (20 us).

4. Retry count unit is to calculate the numbers of the re-transmission. This number is also used in calculating the random back-off contention window, which increases exponentially with respect to retry count.

5. Random Back-off Mechanism defers the transmission for a predefined time when the system wants to transmit data. It will choose different ways to compute the deferred time according to the frame type: i.e., Data frame,
Beacon frame or ATIM frame. For Data frames, the contention window varies from 0 to CW (7-255) time slots. For Beacon transmission, the contention window varies between 0 and 2×Cwmin (14) time slots. As for ATIM frames, the contention window lies within 0 and CWmax (255) time slots. Note that a time slot is 20μs in length.

6. Tx FIFO is 48-byte and Rx FIFO 1-byte in length. The FIFO control is to transmit data from SRAM to FIFO, and at the same time, count the number of data current in the FIFO in order to prevent from data overrun.

What follows is the flow chart performed in our MAC chip. (cf. Section 1 and Section 2, also Figure A.2)

I.1 Section 1: 802.11 protocol Tx flow chart in CSMA/CA submodule

1 Data frame flow chart (subfield = 10000):

![Diagram](image)

Figure A.3: Data Frame flow chart.
For a Data frame, 802_11 mechanism unit will first check whether the fragment number of that frame is zero or not.

If the fragment number is zero, 802.11 protocol mechanism will be performed the flow chart as shown in Figure 3.2, which is exactly the 802.11 CSMA/CA mechanism. It includes DIFS timer and Random back-off timer. In the initial state, if some other station is currently transmitting data (under which the CCA should be one), then the station will wait for the next contention period, and the detect bit will be set to one as shown in Figure A.3. In the next contention period, it will use the random timer to reduce the collision probability in case many stations want to contend for the media at the same time. Within the random back time, if some other station get the right to transmit, and hence, starts to transmit data (under which the CCA should be one), the random timer will stop. The random timer will continue to decrease in the next contention period. If the random timer is decreased to zero (and hence, the station gets the right to transmit), it will justify whether to transmit RTS or not by RTStotal stored in the registers (default value is 3000 bytes). If RTS frame is successfully transmitted, 802_11 mechanism unit will inform FIFO control unit to transmit data from FIFO to Baseband module in 4-bit wide MII bus. The detail of FIFO transmission will be discussed in the Section 2.

If the fragment number is not zero, we will transmit the next fragment (frame) as soon as it could.

*Beacon frame flow chart (subfield = 001000)*:
Figure A.4: Beacon transmission flow chart.

This flow chart describes another frame operation. Beacon frame is transmitted in every beacon interval indicating by TSF timer (default is 100 Kus). Before transmission, the system will justify whether the station is in power save or in awaken state. If it is in power save state, it will enable ATIM timer. If, in addition, it has data to transmit, it shall transmit ATIM request to the destination station and wait for the acknowledgement until the expiration of the ATIM window (default is 4 Kus). In such case, the system will keep awaking until the next beacon period.

If it is in awaken state, it will not start the ATIM timer. Instead, it transmits its own Beacon frame at each beacon interval. In case that the Beacon frame is successfully transmitted, the system will become a temporary Access point (AP); otherwise, it will not transmit its Beacon frame. Note that the latter situation happens when the system receives Beacon frame from some other station before the expiration of its random timer.
ATIM frame flow chart (subfield = 001000):

Figure A.5: ATIM transmission flow chart.

ATIM frame is a frame that is used only in power save mode under ad hoc network. If receiving ATIM from some other station within the ATIM window, the LAN card will stop the random timer. The random timer will continue to decrease in the next transmission time (previous ATIM has been acknowledged). When its random timer decreases to zero, the LAN card will transmit its ATIM. When receiving the ATIM, the LAN card must decide whether the ATIM is for itself or not. Furthermore, the LAN card decides whether it is unicast or multicast. For a unicast ATIM, the LAN card will respond with an ACK; otherwise, the LAN card will continue to decrease the ATIM random back-off timer.

1.2 Section 1: 802.11 protocol Rx flow chart in CSMA/CA submodule

When receiving the frame without the PLCP header removed by the Baseband module, the RX_FSM will remove the MAC header, and store the remaining part of the frame into the registers defined above.
Figure A.6: RX_FSM.

The Figure A.6 is the main part of the RX_FSM in the CSMA/CA module. It describes not only STATE transition but also the each statement within every STATE. The RX_FSM is enabled by Rx_request, which are FSM_en from RX FIFO H/W or Rx_req command from S/W. If any error in each statement happens, it will transit into Abort STATE to discard the received frame. If the CRC_16 is correct, it will check the version and Address 1 fields of the MAC header to determine the received frame is Broadcast or Unicast or Not_me. The detail of Broadcast STATE is Figure A.10. If it transits into Unicast STATE, according to the type field value of the received frame, it will enter Data or Ctrl or Manage STATE respectively. The details of Data, Ctrl and Magm STATES are Figure A.7, 8, 9. After the CRC_32 field of the received frame is calculated without error, it will return to Idle STATE to wait for next frame.
Figure A.7: Data STATE of RX_FSM.

Figure A.8: Control STATE of RX_FSM.

If the received frame is the control frame, it will transit into Ctrl STATE. Because the control frame contains only few MAC header, the CRC_32 check will
be done faster than Data and Management type frames. The STATE first examines
the correctness of the CRC_32, and secondly it transfers the 8-bit SIGNAL register
given by Baseband chip into 1-bit Data_rate_Rx register for Ctrl generation
module to determine the SIGNAL field of response control frame. The third, it
compares the received subtype field value with RTS subtype value, because in the
Ad-hoc mode only RTS, CTS and ACK three kinds of control frames will be
supported and only receiving RTS needs responding (i.e. enable the Ctrl generation
module).

![FLOWCHART]

**Figure A.9:** Management STATE of RX_FSM.

In the Manage STATE, the most statements are similar to the statements of
the Data STATE. Note that when receiving ATIM and Probe response frames,
Control frame generator will be enabled to generate ACK frames. Other subtypes
of management frames do not need acknowledgements.
Figure A.10: Broadcast and Multicast STATE of RX_FSM.

The BC/MC STATE needs a special conditional judgement by using Address 2 field (source address of the received frame) to filter out the owner Broadcast or Multicast frame.

II.1 Section 2: Transmission FIFO Design in CSMA/CA submodule
Figure A.11: TX FIFO Design.

The Tx FIFO is 8-bit wide and 48 bytes in length. Data/MPDU from SRAM will be sent to the FIFO. After its completeness of pumping data to the FIFO from SRAM, the LAN card will send a control signal to notify the FIFO controller to start transmitting the data from FIFO to MII (4 bits). The clock cycle of such data transmission is provided by the Baseband module through Txclk. Therefore, the Txclk decides the data rate. If the data rate is 1 Mbps, the Txclk shall be 250 KHz. If the data rate is 2 Mbps, the Txclk shall be 500 KHz. The sysclk is 44 MHz. The sysclk is also provided by the Baseband module.

II.2 Section 2: Reception FIFO Design in CSMA/CA submodule
As shown in the Figure A.12, we will know that the Rx FIFO is a 1-byte register in compared with the 48 bytes Tx FIFO. This is because we must wait the Rx_Data by the Rx clk, which is smaller than the sysclk. In the Rx FIFO design, there will have a 24 bytes registers to store some information from the Rx frame’s MAC header and 802_11 mechanism unit (Figure A.2) will use these information. At the same time we will use the duration field in grabbed registers to start the NAV timer. If the NAV timer is over, it will inform NAVend signal to Tx_FSM in 802_11 mechanism unit (Figure A.2). In the Rx_FIFO design, Rx_Ready is an important signal from the Baseband module. If the Rx_Ready is active, then we will receive Rx_Data. In addition, if we grab the total 24 bytes information from MAC header, we will inform a FSM_en signal to 802_11 mechanism unit (Figure A.2) and the 802_11 mechanism unit will start to execute its function. Finally, if the reception errors occur, 802_11 mechanism unit will send an abort signal to reset the whole Rx_FIFO Unit.
A.5.2 DMA control module

![DMA block diagram]

Figure A.13: DMA block diagram.

The main purpose of the DMA module is to transmit/receive data from SRAM to FIFO or from FIFO to SRAM. As shown in the Figure A.13, the data transmission or reception is performed by 3 submodules.

First, the data controller controls when or how the data flows from SRAM to FIFO or from FIFO to SRAM. The TX data length will be stored in the configuration registers. The Rx data length will be added in the front of the MPDU in SRAM and the Rx length is two bytes.

Secondly, the address counter and controller will decide the start address of the data in SRAM and count the MPDU length.

The third controller is an interface between CSMA/CA submodule (protocol and FIFO controller) and DMA module. It also needs to prevent from FIFO overrun, and to control the signal handshaking.

The followings are the detailed functions for DMA module:

1. 8 bits bus from SRAM to FIFO:

   In the initial state, the DMA controller will send the MPDU to FIFO within a
Txclk because the sysclk is faster than Txclk. If the txstart for FIFO to MII is enabled, the FIFO control will move the data from FIFO to MII until the whole MPDU is transmitted, and then wait for the acknowledgement of that MPDU.

2. DMA counter and Pointer:

Because the Rx frames in SRAM are either data or management frames, therefore, we will have two pointers and one flag. That is, the two pointers are for Data frames, and the flag is to indicate the vitality of the management frames. Besides, each of the DMA module and Host system has a pointer to point to the current frame in SRAM. When the DMA module moves the frame from SRAM to FIFO, there also must be a counter to monitor the number of data bytes moved. If the counter is equal to the value in length field (the value will be stored in a register first) of the MPDU, we will know that this is the last byte of the MPDU. Then TX complete control signal will be sent to 802.11 mechanism unit (Figure A.2). Therefore, we can move the next MPDU after we receive the previous ACK.
A.5.3 BIU control module

As shown in the Figure A.14, the BIU module includes address decoder and combinational control logic. Its interface to the Host system follows PCMCIA, which contains the initial setup (decode CIS) and I/O function (decode registers and SRAM). The address decoder consists of four parts.

The configuration registers store the initial values about 802.11 protocol, such as RTSThreshold value, ATIM window size, etc. These values are shown in the previous defined registers.

The SMI (Status, Command, and Interrupt) registers that store the important information to notify the Driver or the MAC chip, such as transmit complete, transmit error and transmit request.

The CIS that stores the initialization and setup information of the LAN card, such as card ID and the manufacturer ID, etc.

On the other hand, the combination logic assigns some control signals
according to inputs from Host system. In summary, the BIU module performs the Wireless LAN card interface functionality. According to PCMCIA standardization, there are 26 address pins, but we only use 16 of them. Details will be shown below.

<table>
<thead>
<tr>
<th>Reserved</th>
<th>SRAM / CIS / Con_Reg</th>
<th>Address 15-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address 17-25</td>
<td></td>
<td>Address 16</td>
</tr>
</tbody>
</table>

*Figure A.15: Address decode format.*

The I/O port Address uses the 280H, 281H, and the 282H. The 280H is for the Address[7:0] and the 281H is for the Address[15:0]. Finally, the 282H is for the data bus.

If the Address[16] is one, it represents the Address is for SRAM address, otherwise, it represents the CIS or other Registers.

The followings are the relative address locations for the Configuration registers and the CIS data.

<table>
<thead>
<tr>
<th>Address location</th>
<th>Registers name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1FFH</td>
<td>CIS Data</td>
</tr>
<tr>
<td>200H</td>
<td>Configuration Option Register</td>
</tr>
<tr>
<td>202H</td>
<td>Card Configuration and Status Register</td>
</tr>
<tr>
<td>204H</td>
<td>Pin Replacement Register</td>
</tr>
<tr>
<td>206H</td>
<td>Socket and Copy Register</td>
</tr>
<tr>
<td>208H</td>
<td>Status Tx Register</td>
</tr>
<tr>
<td>20aH</td>
<td>Status Rx Register</td>
</tr>
<tr>
<td>20cH</td>
<td>ATIM window[3:0] and Power-Saved mode[4]</td>
</tr>
<tr>
<td>20eH</td>
<td>RTS threshold [7:0]</td>
</tr>
<tr>
<td>210H</td>
<td>RTS threshold [15:8]</td>
</tr>
<tr>
<td>212H</td>
<td>Tx_Data_Length[7:0]</td>
</tr>
<tr>
<td>214H</td>
<td>Tx_Data_Length[15:8]</td>
</tr>
<tr>
<td>216H</td>
<td>Tx_Nagm_Length[7:0]</td>
</tr>
<tr>
<td>21eH</td>
<td>Source Address[7:0]</td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>220H</td>
<td>Source Address[15:8]</td>
</tr>
<tr>
<td>222H</td>
<td>Source Address[23:16]</td>
</tr>
<tr>
<td>224H</td>
<td>Source Address[31:24]</td>
</tr>
<tr>
<td>226H</td>
<td>Source Address[39:32]</td>
</tr>
<tr>
<td>228H</td>
<td>Source Address[47:40]</td>
</tr>
<tr>
<td>22AH</td>
<td>Tx_Point_Register</td>
</tr>
<tr>
<td>22CH</td>
<td>Rx_Point_Register</td>
</tr>
</tbody>
</table>

*Table A.1: Address Location for Registers within BIU module.*

Table A.2 is the function for reading or writing the Attribute memory or the I/O function.

<table>
<thead>
<tr>
<th>Function mode</th>
<th>REG_</th>
<th>CEI_</th>
<th>IOWR_</th>
<th>IORD_</th>
<th>A0</th>
<th>OE_</th>
<th>WE_</th>
<th>D[7:0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute Memory Read</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>Even Byte</td>
</tr>
<tr>
<td>Attribute Memory Write</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>Even Byte</td>
</tr>
<tr>
<td>I/O Read Even Byte</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>Even Byte</td>
</tr>
<tr>
<td>I/O Read Odd Byte</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>Odd Byte</td>
</tr>
<tr>
<td>I/O Write Even Byte</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>Even Byte</td>
</tr>
<tr>
<td>I/O Write Odd Byte</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>Odd Byte</td>
</tr>
</tbody>
</table>

*Table A.2: Read and Write function for I/O and Attribute Memory.*
A.5.4 SRAM module

A.5.4.1 PIN DESCRIPTION

<table>
<thead>
<tr>
<th>Pin</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0-A14</td>
<td>I / O</td>
<td>Address inputs</td>
</tr>
<tr>
<td>D0-D7</td>
<td>I / O</td>
<td>Data inputs/outputs</td>
</tr>
<tr>
<td>CS</td>
<td>I</td>
<td>Chip enable</td>
</tr>
<tr>
<td>WE</td>
<td>I</td>
<td>Write enable</td>
</tr>
<tr>
<td>OE</td>
<td>I</td>
<td>Output enable</td>
</tr>
<tr>
<td>VDD</td>
<td>I</td>
<td>Power supply</td>
</tr>
<tr>
<td>VSS</td>
<td>I</td>
<td>Ground</td>
</tr>
</tbody>
</table>

FUNCTION DESCRIPTION

![Diagram of SRAM partition](image)

Figure A.16: SRAM (32 K × 8) Partition.

As shown in the Figure A.16, we will divide the SRAM into two portions: Tx portion and Rx portion. The Tx and Rx portions respectively consist of 2 units and 10 units for which each unit is 2.5K bytes in length. The main reason for unbalanced partitions for transmitting and receiving is because during transmission, the system can store the transmitted data in the Host memory, while during receiving, it is better to store the received data immediately as they are received in order to prevent from the frame loss.
The first portion will be further divided into two sub-portions: one is
dedicatedly for Management frame, and the other, for Data frame. Since the
priority of the Management frame is higher than the data frame according to IEEE
802.11, we will transmit the Management frame instead of the Data frame if both
of them are in the SRAM at the same time. Therefore, we will have a register to
store the frame’s information that is to justify whether the packet is data or
management packet.

Finally, it needs to be pointed out that the access time of SRAM is assumed
to lie between 10ns and 20ns, and the sysclk used by MAC chip is 44 MHz.
A.5.5 Control Frame Generation module

A.5.5.1 Interface between control module and baseband module

<table>
<thead>
<tr>
<th>Pin</th>
<th>Bus</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXCLK</td>
<td>1</td>
<td>I</td>
<td>Reference clock of TXD provided by BB</td>
</tr>
<tr>
<td>TXD</td>
<td>4</td>
<td>O</td>
<td>Data pins</td>
</tr>
<tr>
<td>MCLK</td>
<td>1</td>
<td>I</td>
<td>System clock provided by BaseBand</td>
</tr>
<tr>
<td>TX_START</td>
<td>1</td>
<td>O</td>
<td>Enable baseband to transfer data</td>
</tr>
</tbody>
</table>

A.5.5.2 Interface between control module and CSMA/CA module

<table>
<thead>
<tr>
<th>Pin</th>
<th>Bus</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>En_Tx</td>
<td>1</td>
<td>I</td>
<td>Enable Control Module by TX_FSM</td>
</tr>
<tr>
<td>En_Rx</td>
<td>1</td>
<td>I</td>
<td>Enable Control Module by RX_FSM</td>
</tr>
<tr>
<td>clr_</td>
<td>1</td>
<td>I</td>
<td>Reset</td>
</tr>
<tr>
<td>Ctrl_retry</td>
<td>1</td>
<td>I</td>
<td>RTS retransmission</td>
</tr>
<tr>
<td>RTS_confirm</td>
<td>1</td>
<td>O</td>
<td>RTS was transmitted</td>
</tr>
<tr>
<td>Data_mtc_TX</td>
<td>1</td>
<td>I</td>
<td>PLCP SIGNAL field values from TX_FSM</td>
</tr>
<tr>
<td>Data_mtc_RX</td>
<td>1</td>
<td>I</td>
<td>PLCP SIGNAL field values from RX_FSM</td>
</tr>
<tr>
<td>Ctrl_subtype_TX</td>
<td>4</td>
<td>I</td>
<td>MAC_header Subtype field from TX_FSM</td>
</tr>
<tr>
<td>Ctrl_subtype_RX</td>
<td>4</td>
<td>I</td>
<td>MAC_header Subtype field from RX_FSM</td>
</tr>
<tr>
<td>Power_manage</td>
<td>1</td>
<td>I</td>
<td>Power Management field value</td>
</tr>
<tr>
<td>More_frag</td>
<td>1</td>
<td>I</td>
<td>For calculating ACK’s DURATION field</td>
</tr>
<tr>
<td>AID</td>
<td>16</td>
<td>I</td>
<td>MAC_header AID field values</td>
</tr>
<tr>
<td>Add_1_TX</td>
<td>48</td>
<td>I</td>
<td>MAC_header Add_1 field from TX_FSM</td>
</tr>
<tr>
<td>Add_2_TX</td>
<td>48</td>
<td>I</td>
<td>MAC_header Add_2 field from TX_FSM</td>
</tr>
<tr>
<td>Add_2_RX</td>
<td>48</td>
<td>I</td>
<td>MAC_header Add_2 field from RX_FSM</td>
</tr>
</tbody>
</table>
A.5.5.3 Control Frame module architecture

- Preamble generator:
  SYN + SFD fields are constant

*Figure A.17: PLCP Frame Generator.*
A.5.5.4 MAC and PLCP Header generators

![Diagram showing Control Frame Selection]

*Figure A.18: Control Frame Selection.*

A.5.5.5 Control frame module description

It is necessary to respond to the Control frames (such as RTS frames, Data frames, ATIM frames and Probe Response frames) within the SIFS interval which is suggested to be 10us in IEEE 802.11. In our design, the Control Frame Module will process the responding to the Control frames in order to shorten the SIFS interval, and hence, to speed up the overall MSDU transmission. This is the main purpose for the design of this Control Frame Module.

The Control Frame Module shall generate the complete control frame including Preamble, the PLCP header and MAC header. Thus, this module needs CRC_32 and CRC_16 generators.
The CRC_32 generator polynomial is

\[ G(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1. \]

The CRC_16 generation polynomial is

\[ G(X) = X^{16} + X^{12} + X^5 + 1 \]

The initial condition of the CRC_32 calculation is FFFF_FFFFH; and the final remainder of the CRC_32 operation is C704_DD7BH. The CRC circuits adopted here are not conventional serial architecture. These adopt parallel I/O architecture for compatible with MII interface (4-bit data pins).
Appendix B  Timing Diagram

B.1 PCMCIA I/O Read/Write Timing Diagram

![Timing Diagram]

*Figure B.1:*  I/O Read/Write Timing without Wait State.

![Timing Diagram with Wait State]

*Figure B.2:*  I/O Read/Write Timing with Wait State.
B.2 SRAM Timing Diagram

![SRAM Timing Diagram](image)

*Figure B.3: SRAM Write Timing.*

![SRAM Timing Diagram](image)

*Figure B.4: SRAM Read Timing.*
B.3 Timing Diagram Between Transmission FIFO and DMA Module

Figure B.5: Timing between Transmission FIFO and DMA Module.

B.4 Timing Diagram Between Reception FIFO and DMA Module

Figure B.6: Timing between Reception FIFO and DMA Module.
B.5 Timing Between Baseband Module and MAC Chip

\[ \text{Figure B.7: Timing between Baseband Module and MAC Chip.} \]
B.6 Block Diagram for MAC Controller

Figure B.8: Block diagram of MAC Controller.
B.7 Tx_FIFO and DMA Interface Block Diagram

![Diagram of TX_FIFO and DMA Module interface](image)

*Figure B.9: Tx_FIFO and DMA Interface Block Diagram.*

B.8 Rx_FIFO and DMA Interface Block Diagram

![Diagram of RX_FIFO and DMA Module interface](image)

*Figure B.10: Rx_FIFO and DMA Interface Block Diagram.*
B.9 BIU Module and DMA Module Interface Block Diagram

Figure B.11: BIU Module and DMA Modules Interface Block Diagram.
B.10  CSMA/CA with Tx_FSM and Control Module Interface Block Diagram

![Diagram showing CSMA/CA Module with Tx_FSM connected to Control Module]

Figure B.12:  CSMA/CA with Tx_FSM and Control Module Interface Block Diagram.
B.11 CSMA/CA Module and Baseband Module Interface Block Diagram

![Diagram showing the interface between the CSMA/CA Module and the Baseband Module.]

Figure B.13: CSMA/CA module and Baseband Module Interface Block Diagram.
B.12 Pin Architecture

![MAC Chip Pin Architecture](image)

*Figure B.14: MAC Chip Pin Architecture.*
## CSMA/CD VS. CSMA/CA

<table>
<thead>
<tr>
<th>CHANNEL CONDITION</th>
<th>CSMA/CD (IEEE 802.3)</th>
<th>CSMA/CA (IEEE 802.11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel is <em>Busy</em> before transmitting</td>
<td>Continue detecting and transmit data until channel is <em>Idle</em></td>
<td>Defer the <em>busy</em> medium (includes DIFS) until channel becomes <em>Idle</em> and then execute random back-off Mechanism</td>
</tr>
<tr>
<td>Channel is <em>Idle</em> before transmitting</td>
<td>Transmit data and detect channel</td>
<td>(1). 1st MSDU&lt;br&gt;⇧ defer IFS time and transmit data&lt;br&gt; (2). 2nd or above MSDU&lt;br&gt;⇧ defer IFS time and choose a random back-off, then transmit it</td>
</tr>
<tr>
<td>Occurred collision after transmitting</td>
<td>Send a <em>jamming signal</em> and then execute random back-off</td>
<td>Timers with ARQ scheme is timeout, then retransmit according to DCF procedure</td>
</tr>
</tbody>
</table>

**Random Back-off Mechanism (R)**<br> *(Slot time)*<br>\[ R = \{0, 1, 2 \}^{n-1} \]<br>\[ K = \min(n, 10), \]<br>where \( n \) is the number of collision<br>\[ R = \text{Random (K)} \]<br>\[ K = [0, CW], \]<br>where \( CW_{\text{min}} \leq CW \leq CW_{\text{max}} \)
Appendix D

D.1 DCF Timing Relationships

Figure D.1: DCF Timing Relationships.
D.2 IEEE 802.11 MAC Frame Format

![IEEE 802.11 MAC Frame Format](image)

**Figure D.2:** IEEE 802.11 Data Frame Format.

Bytes:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>2</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>6</th>
<th>0-2312</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Control</td>
<td>Duration ID</td>
<td>Addr 1</td>
<td>Addr 2</td>
<td>Addr 3</td>
<td>Sequence Control</td>
<td>Addr 4</td>
<td>Frame Body</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>

**Figure D.3:** Management Frame Format.

Bytes:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>2</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0-2312</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Control</td>
<td>Duration ID</td>
<td>DA</td>
<td>SA</td>
<td>BSSID</td>
<td>Sequence Control</td>
<td>Frame Body</td>
<td>FCS</td>
<td></td>
</tr>
</tbody>
</table>

**Figure D.4:** RTS Frame Format.
<table>
<thead>
<tr>
<th>Bytes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frame Control</th>
<th>Duration</th>
<th>RA</th>
<th>FCS</th>
</tr>
</thead>
</table>

**Note:**

1. RA : Receiver Address
2. TA : Transmitter Address
3. FCS : Frame Check Sequence

*Figure D.5:* CTS and ACK Frame Format.