Rapid Implementation of the MAC and Interface Circuits for the Wireless LAN Cards Using FPGA

Songchar Jiang

Abstract: **This paper studies the rapid design and implementation of the medium access control (MAC) and related interface circuits for 802.11 wireless LANs based on the field programmed gate array (FPGA) technology. Our design is thus aimed to support both the distributed coordination function (DCF) and the point coordination function (PCF) with the aid of FPGA technology. Furthermore, in an infrastructure network, some stations may serve as the access points (APs) which may function like a learning bridge. This paper will also discuss how to design for such application. The hardware of the MAC and interface may at least consist of three major parts: wireless transmission and reception processes and interface, host (bus) interface, and the interface to the distributed system (optional). Through the increasing popularity of FPGA design, this paper presents how Complex Programmable Logic Devices (CPLD) can be utilized for speedy design of prototypes. It also demonstrates that there is much room for low-cost hardware prototype design to accelerate the processing speed of the MAC control function and for field testing.**

Index Terms: **802.11, MAC, DCF, PCF, FPGA.**

I. INTRODUCTION

In the 802.11 standard [1], the physical layer can support either the radio frequency (RF) medium or the InfraRed medium. The RF medium includes either direct sequence spread spectrum or frequency-hopping spread spectrum. For such a reason, the standard first defines the lowest layer as the physical medium dependent (PMD) layer, and then uses the physical layer convergence procedure (PLCP) as the primitive and interface for medium access control (MAC). Also note that within IEEE 802.X, the logical link control (LLC) layer has to be transparent to any stations associated with the local area network. Therefore, some LLC functions which are not necessary for other types of local area networks (LANs) must be performed by the MAC layer according to the 802.11 specification. Observing that, we realize that the MAC control in 802.11 is much more complicated than in other 802.X. In practice, however, we may implement the hardware as much as possible and thus reduce the software load to the least. The prospective advantages are obvious in the speed and the relieved loading of CPU in host computer. In [2], we have shown the PLCP and the physical medium dependent layer, including the direct sequence spread spectrum system front end and the baseband processing part of an 802.11 wireless LAN card, and the MAC control can be realized through the off-the-shelve devices and commercial MAC controllers. This paper will demonstrate how the MAC control and the required interface circuits can be rapidly implemented through the field programmed gate array (FPGA) technology.

The 802.11 standard employs the carrier sense multiple access with collision avoidance (CSMA/CA) as the medium access control protocol. This protocol utilizes the distributed coordination function (DCF), where collision avoidance is achieved by using a contention window and the backoff procedure, with the optional point coordination function (PCF). The DCF scheme can be directly placed on the ad-hoc networks. An ad-hoc network is an independent basic service set (BSS) that is created spontaneously and can be operated without assistance from expert engineers. The DCF scheme can be employed for channel access method and, when traffic congestion occurs or time-bounded service is required, one among those stations will serve as a point coordinator to add the PCF operation. The PCF scheme is basically a polling protocol. Note that the extension of the BSS network by connecting to other BSSs or networks is then called an infrastructure network. An infrastructure network may use both the PCF and DCF operated in contention-free and contention ways respectively [1]–[4].

During the past years, FPGAs have been extensively utilized to rapidly implement many kinds of digital systems. The obvious advantages are friendly to use, flexibility for modification (reprogrammed), and easy for verification and device programming. The FPGA design tool we use is the MAX+PLUS II, which is developed by Altera and can be easily realized through Complex PLD (CPLD) or Erasable PLD chips [5]–[8]. In practice, four steps must be followed: design entry, compilation, verification, and device programming. There are four different types of design entry: graphic entry, text entry, waveform entry, and industry-standard CAE entry (EDIF files). The verification may utilize either simulation or timing analysis, or both.

Our prototype design is shown in Fig. 1, where we have included the optional distributed system (DS) interface. It is constituted by six major parts: The RF front end and baseband processing, the PLCP unit, the MAC control unit, the core control unit, the host interface, and the optional distributed system (DS) control state machine and interface unit. The core control unit is the operation center that is directed by the device drivers programmed into the core microprocessor. The MAC control unit is the hardware that is aimed to accelerate the MAC processes.

Manuscript received January 1, 1999; approved for publication by Dong Ho Cho, Division III Editor, September 17, 1999.

Songchar Jiang is with Department of Electrical Engineering, National Yunlin University of Science and Technology, 123 University Road, Sec. 3, Touliu City, Yunlin, Taiwan, 640 R.O.C., e-mail: jiangsj@flame.yuntech.edu.tw.

This work was supported by National Science Council under contracts NSC84-2221-E-224-002, NSC85-2221-E-224-001, and NSC86-2221-E-224-009.

1229-2370/99/$10.00 © 1999 KICS

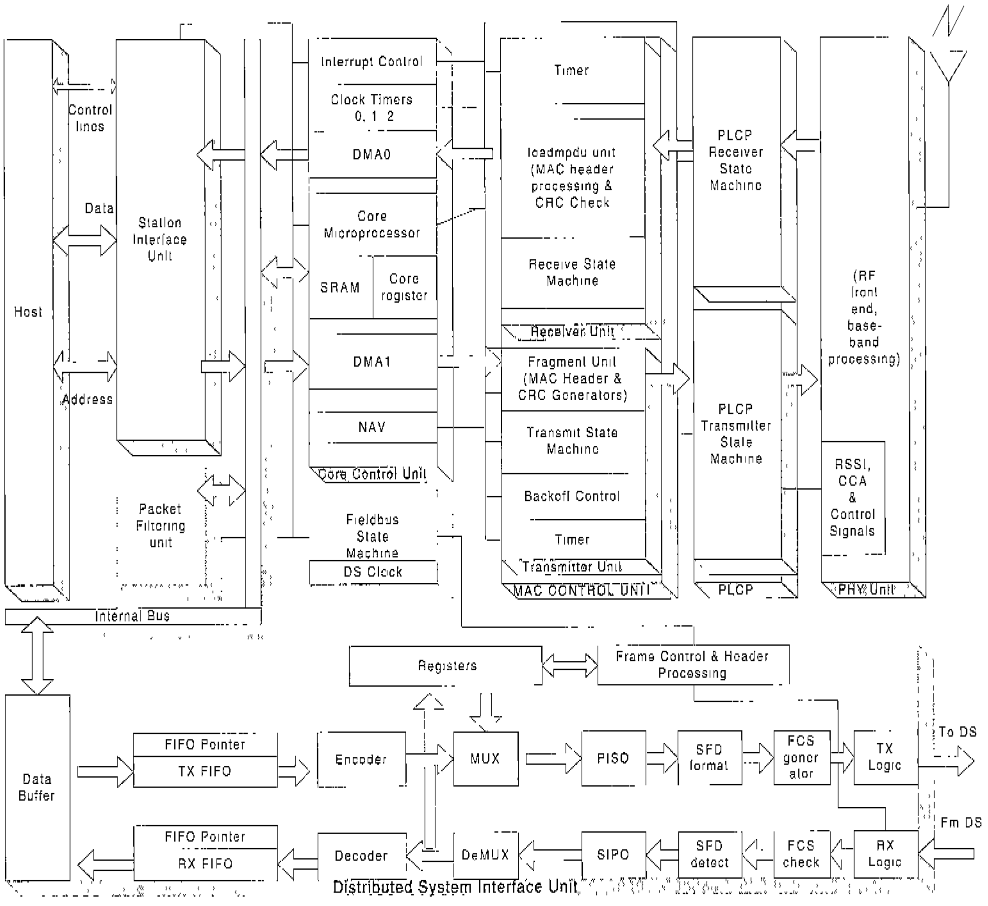


Fig. 1. The proposed architecture of the wireless LAN card prototype.

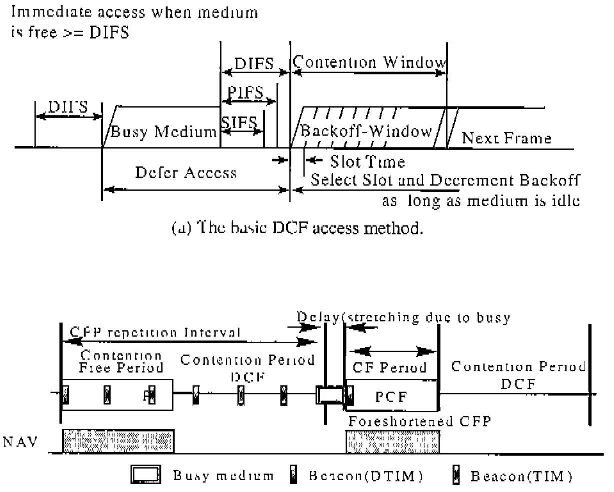
The DS part will be briefly introduced in this paper, but the readers can refer to other references [3], [9], [10] for more details. This paper is organized as follows. The MAC function is briefly illustrated in Section II. Section III describes the design and implementation of host system interface, and the MAC Transmitter Unit implementation is demonstrated in Section IV. The MAC Receiver Unit implementation is shown in Section V, and Section VI describes the device drivers for the core control unit and the DS interface part. Section VII then concludes this paper.

II. THE MEDIUM ACCESS CONTROL

A. The DCF and PCF Protocols

As mentioned, the 802.11 employs the CSMA/CA which can be implemented by using either the physical carrier sense (energy detection or detection) or virtual carrier sense (by network allocation vector (NAV)) [1]–[3]. Frames to be transmitted may contain a duration field that can be used to set the NAV values of all stations to a non-zero duration. Stations receiving a valid frame shall update their NAV with the information contained in the duration field, but only when the new NAV value is greater than the current NAV value, and only when the frame is not addressed to the receiving station. To relieve the hidden terminal problem, it also provides an optional handshaking (request-to-send (RTS) and clear-to-send (CTS)) [1], [4] prior to formal transmission of the packet. Compared to the DCF inter-frame space (DIFS) and PCF inter-frame space (PIFS) times, the idle time space for both RTS and CTS signals is shorter and is called the short inter-frame space (SIFS), the same as that for the acknowledgement frame, so that they have higher priority to seize the channel access right. A brief summary of the DCF operation is shown in Fig. 2(a), where DIFS is the DCF inter-frame space, PIFS is the PCF inter-frame space, and SIFS is the short inter-frame space. Among them, the SIFS time is the shortest time space that a control packet has to wait for idle medium, and the DIFS is the largest time space that the DCF contention packet has to wait for idle channel. It can be described as follows. When a station tries to transmit a packet, it must defer access until the physical layer reports the medium being idle. Then it initiates the backoff procedure after a DIFS period in order to gain the access right. The procedure is as follows:

- Choose the backoff time as
  BACKOFF TIME = INT(CW*Random())* SLOT TIME
  where the CW is the contention window and is an integer between 8 and 256, while Random() denotes a uniform distribution over (0,1].
- During the backoff timer, each station must continue sensing the medium. The BACKOFF TIME will be reduced by a SLOT TIME each time when an idle slot is detected, while the timer will be stopped when the medium is busy.
- When the BACKOFF TIMER decreases to zero and the medium is idle, then it can transmit its packet.



(a) The basic DCF access method.

(b) The operation of PCF.

Fig. 2. The illustration of DCF and PCF.

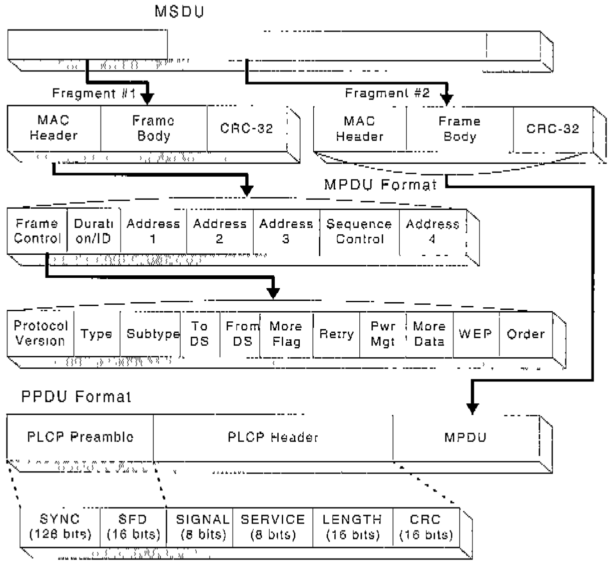


Fig. 3. The data frame formats.

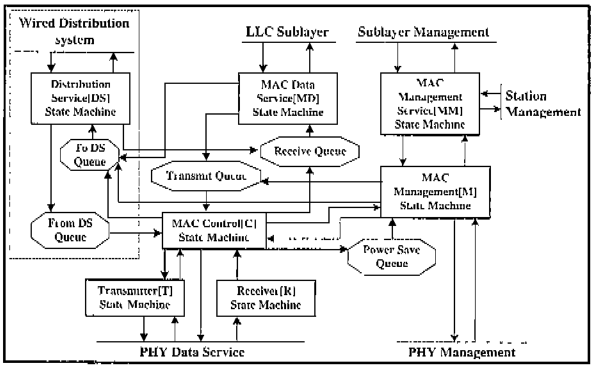

Wired Distribution System
LLC Sublayer
Sublayer Management
Distribution Service[DS] State Machine
MAC Data Service[MD] State Machine
MAC Management Service[MM] State Machine
Station Management
To DS Queue
Receive Queue
Transmit Queue
MAC Management[M] State Machine
From DS Queue
MAC Control[C] State Machine
Power Save Queue
Transmitter[T] State Machine
Receiver[R] State Machine
PHY Data Service
PHY Management

Fig. 4. Overview of the MAC state machine.

- If the access is successful, then the value of CW resets to the minimum (8); otherwise it may double until reaching the maximum (256).

Due to difficulty in collision detection, the receiver station has to acknowledge the sender station (except in the case of broadcasting). As usual, the header of MPDU or acknowledgement frame can also include the information about transmission of next frame. Another kind of access method in 802.11 is the PCF in which a point coordinator (often the access point) uses a shorter PCF Inter-frame space (PIFS) instead of DIFS so that it has higher priority to gain the access right. The PCF is a centralized control in which the point coordinator has to maintain a polling list of stations. Stations on the list will be polled by the point coordinator to gain access to the channel without contention. In 802.11, both DCF and PCF access times are integrated into a superframe; thus a station can only enter the polling list through the contention in the DCF period. To avoid access request by a station that is not on the list, all stations except the coordinator have to set its NAV to a determined maximum value. To resume the contention period, the coordinator has to broadcast a frame to reset the NAV values of all stations immediately in order not to waste bandwidth [1]–[3]. Its operation is shown in Fig. 2(b). The PCF protocol can be utilized to support time-bounded service or even more, the periodic traffic such as the voice packets [3]–[11].

We note that a MAC service data unit (MSDU) can be segmented into several MAC protocol data units (MPDUs) for continued transmission. The frame formats used in the MAC layer and in the physical layer are shown in Fig. 3, where we can see that an MSDU may be fragmented into several MPDUs with added MAC headers and CRC-32s. The MAC header must include frame control field, duration/ID, address field, and sequence control information. The frame control field is to indicate the type of traffic, including frame type, traffic source medium and destination medium, new or retry, power management, subsequent transmission indication, and wired equipment privacy (WEP) active request, etc. The MAC frame format has to be converted into the physical protocol data unit (PPDU) that the physical medium dependent (PMD) layer can recognize prior to physical transmission. It is done by preceding the MPDU with the PLCP preamble and header. The PLCP transmit and receive state machines have been outlined in the 802.11 standard, and its implementation has been described in [2].

B. The MAC Control Function

An overview of the MAC control state machine is shown in Fig. 4. A complete state machine must include the MAC data frame access control, DS service control, and the management service control. In this paper, we focus on the essential MAC control state machine which provides both the basic DCF and the PCF operations. For other detailed operations such as broadcast, association, disassociation, authentication, WEP, etc., which are beyond the scope of this paper, please refer to the standard [1].

The MAC Control State Machine

The MAC control state machine provides operation of the DCF and the PCF (optional) for transfer of frames over the wireless medium. It also provides fragmentation, reassembly, and part of power management (such as wake-up service). In other words, the control state machine provides the primitive

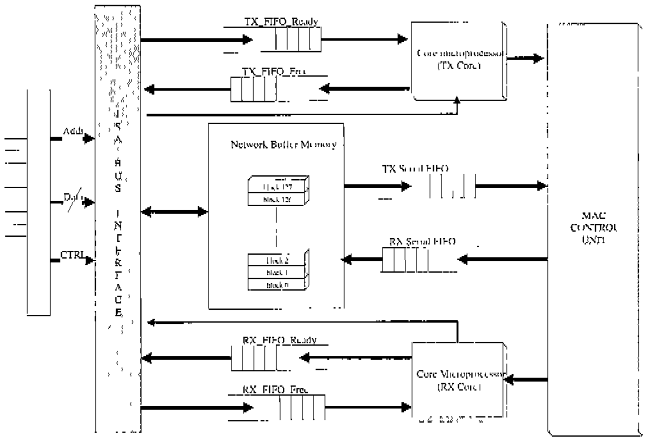


Fig. 5. The host interface structure.

that bridges the physical medium dependent layer and the MAC layer. The detailed descriptions of DCF transmission control and top-level reception control, and polling procedure and operation of a point coordinator at an access point can be found in [1].

The Transmitter and Receiver State Machines

The transmitter state machine handles transfer of the MPDU octets to the physical layer. It also calculates the CRC value of the MPDU header and payload, and transfers it after the end of MPDU is reached, and then ends the transmission and returns to its idle state awaiting the next transmit request from the MAC control state machine. Note that the CRC generation must be accomplished in this state machine in order to accommodate Beacon and Probe response frames, where the contents of the timestamp field in the MPDU are not known until immediately prior to transmission. The receiver state machine handles transfer, validation, and duplicates filtering of frames passed from the physical layer pursuant to reception from the wireless medium. This state machine also performs the NAV updates, while other state machines perform all NAV resets (which reduce the NAV value, generally to zero). Analysis of the MAC header is necessary in order to determine the proper NAV value. Note that the NAV update of the destined station must be performed before address decoding because the NAV is updated based on all valid frames the station received, not just frames addressed to the station. The detailed description can be found from the standard.

III. THE HOST INTERFACE DESIGN

The host interface we consider is the ISA-bus which is shown in Fig. 5, where it is similar to the structure often used in both the FDDI interface and the single-copy protocol stack network [12], and is sometimes called the WITLESS (Workstation Interface that's Low-cost, Efficient, Scalable and Stupid) architecture [13]. The interface structure for transmission we have implemented is shown in Fig. 6, which we explain as follows (The receiving interface is the reversal process which we shall omit). As shown, we utilize the SRAM as the data buffer where its address is given from $(00000)_H$ to $(3FFFF)_H$. We also use TX_FIFO_READY and TX_FIFO_FREE as the pointers of SRAM, which are $256(FF)_H$ bytes FIFO (First-in, First-out). The TX_FIFO_FREE is to record the addresses of available memory blocks in SRAM, and the RX_FIFO_READY is to record those blocks in SRAM that are already in use. Note that since the maximum length of a MPDU is 2312 bytes according to the 802.11 standard, we allow the data in SRAM to be read or to write the SRAM in a fixed-size block $(2312(908)_H$ bytes) within one transfer cycle, even the real packet size may be smaller. The operation of transmission control involves both the host computer and the core microprocessor, which we briefly describe as follows. Note that in this structure, the data copying from SRAM to DMA is a parallel-to-serial operation which is compliant with the PLCP procedure and convenient for MAC core controller.

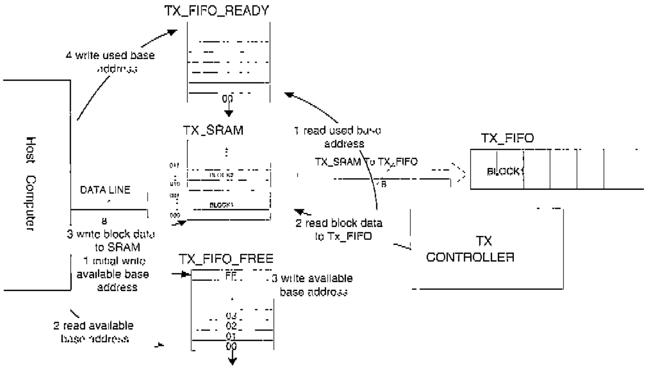

TX_FIFO_READY
4 write used base address
00
TX_SRAM
1 read used base address
TX_FIFO
TX_SRAM To TX_FIFO
8
BLOCK
Host Computer
DATA LINE
BLOCK
8
3 write block data to SRAM
2 read block data to Tx_FIFO
TX CONTROLLER
1 initial write available base address
TX_FIFO_FREE
FF
3 write available base address
2 read available base address
03
02
01
00

Fig. 6. The illustration of the host interface operation.

Host Computer

1. Initialization. Write $(00)_H \sim (FF)_H$ data into TX_FIFO_FREE, indicating all addresses $(00000)_H \sim (3FFFF)_H$ are free for use.
2. When the host wants to transmit data, it will read the available base address in TX_FIFO_FREE that indicates the available block in SRAM. The first time it reads is $(00)_H$ and will be $(01)_H$ next time in order.
3. The host writes its data block of 2312 bytes into the corresponding SRAM, according to the pointer index in TX_FIFO_FREE. For example, $(00)_H$ corresponds to the block $(00000)_H \sim (00908)_H$ in SRAM.
4. The host writes the available base address (the pointer of TX_FIFO_FREE); for example, the first time is $(00)_H$, back into the RX_FIFO_READY to indicate that there are data in the corresponding block, such as the $(00000)_H \sim (00908)_H$, in SRAM waiting for transmission.
5. Repeats steps 2 $\sim$ 4 until all data transfers are finished.

The transmission operation associated with the MAC controller is shown in Fig. 7.

TX Core of Core Microprocessor

1. TX_CORE reads the pointer in RX_FIFO_READY, for example, $(00)_H$ the first time, and verifies that there are data in SRAM, for example, data in block $(00000)_H \sim (00908)_H$, ready for transmission.
2. Moves data in the corresponding block of SRAM into the

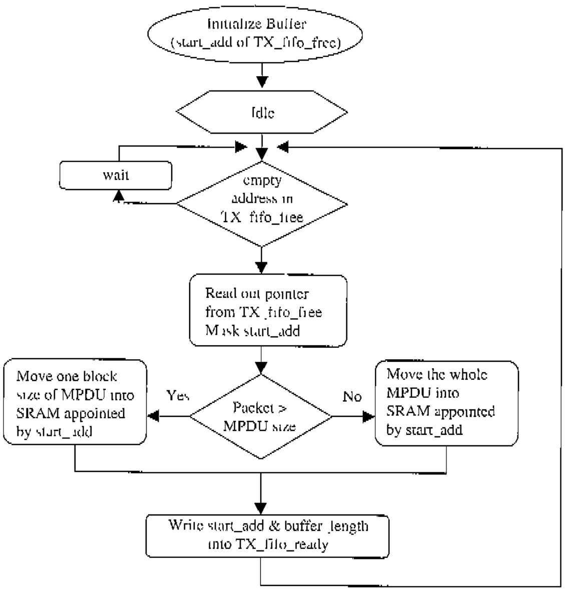


Fig. 7. Host bus transmission operation flow.

TX_FIFO (DMA1 in the Core Control Unit) and informs the Transmitter Unit in the MAC Control unit for further processing.

3. Updates TX_FIFO_FREE pointer to indicate that its corresponding block in SRAM is being read, and thus released for new data.
4. Repeats steps 1∼3 until all data in SRAM are downloaded into TX_FIFO. Fig. 8 shows the more detailed operation flow.

IV. THE TRANSMITTER UNIT DESIGN

The MAC control unit consists of two major parts: the transmitter unit and the receiver unit. This section will demonstrate the FPGA implementation of transmission for MAC control based on the description in Section II, and the receiver unit will be demonstrated in the next section. The first level of the hierarchy functional blocks of the Transmitter Unit mainly consists of four blocks, i.e., the fragment, the timer, the backoff, and the transmission state machine (Txsm) as shown in Fig. 9. The transmit state machine (Txsm) is the core control part of this unit, which has to realize the wireless transmission procedure as defined in Section II, and to call the related function blocks to fulfill any particular task (such as to request the fragment block for packet formation). The fragment function block in Fig. 9 is shown further in details in Fig. 10, where it includes five blocks: MSDUFIFO, PrepareHeader, MPDUFIFO, LoadCRC, and LoadMPDU_ind. The principal function of fragment is to set up the fragment threshold according to the frame length confinement specified in the PLCP, to segment the MAC service data unit (MSDU) frame into the MAC protocol data unit (MPDU) frame in case that the length of MSDU to be transmitted is over the threshold, to load the MPDU frame into MPDU FIFO, and to prepare the MAC header and loading, and CRC generation and loading. Informed for transmission is signaled by MAC layer through the Transmit Enable (TE) pin. The 802.11 standard suggests that a CRC-32 be suitable for error detection. In practice, a CRC-16 (16 bits) may be enough for detecting the frame error because the maximum length of an MPDU frame is 2312 bits. The design of the CRC-32 generator and decoder is similar to that of CRC-16, which has been described in [14].

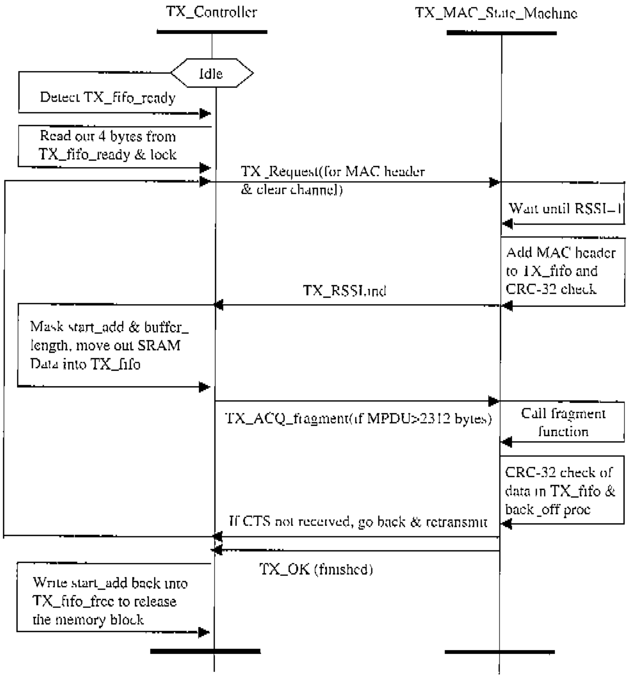


Fig. 8. Core processor transmission operation flow.

A unique feature of 802.11 is the backoff procedure. Due to the difficulty in collision detection, 802.11 adopts the CSMA/CA scheme. The collision avoidance is implemented through the backoff procedure. The timer is to generate the necessarily periodic time slots at a 20 ms period basis and some delays specified in both the distributed coordination function (DCF) and point distributed function (PCF). The delays, for example, include short inter-frame space (SIFS-10ms), PCF inter-frame space (PIFS-30ms), and DCF inter-frame space (DIFS-50ms). The backoff procedure is set up by the timer and the backoff functional blocks in Fig. 9 when it is triggered by the BACKOFF signal. The detailed FPGA circuits can be found from [2]–[9]. To verify such design, the AHDL simulator (provided by Altera) can be used to confirm our implementation. With appropriate input signals, we are able to observe and verify correct operation within the logic circuits. When the station has data to transmit, the upper layer will download the MSDU data into the MSDU FIFO buffer via the DATA_IN[7...0] data bus and waits for fragmentation. The first task of fragmentation is to generate the MAC header. After segmenting the MSDU into MPDU and put behind the MAC header, the CRC code is also generated and appended to the MPDU. This will complete the fragmentation process. The data in each step can be verified through simulation but is omitted in this paper due to limited space. Fig. 11 shows the timing diagram of the MPDU transmission, where we can see the intact MPDU packet formation.

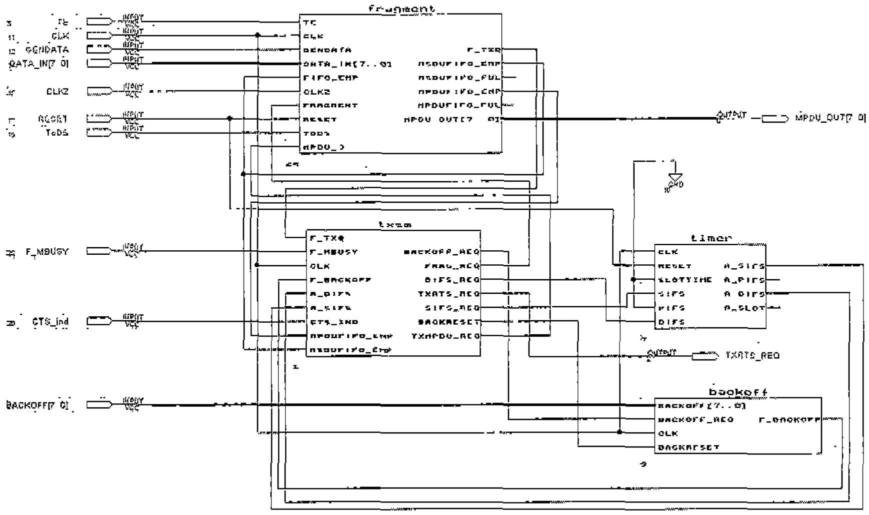


Fig. 9. The first hierarchy of the transmitter unit.

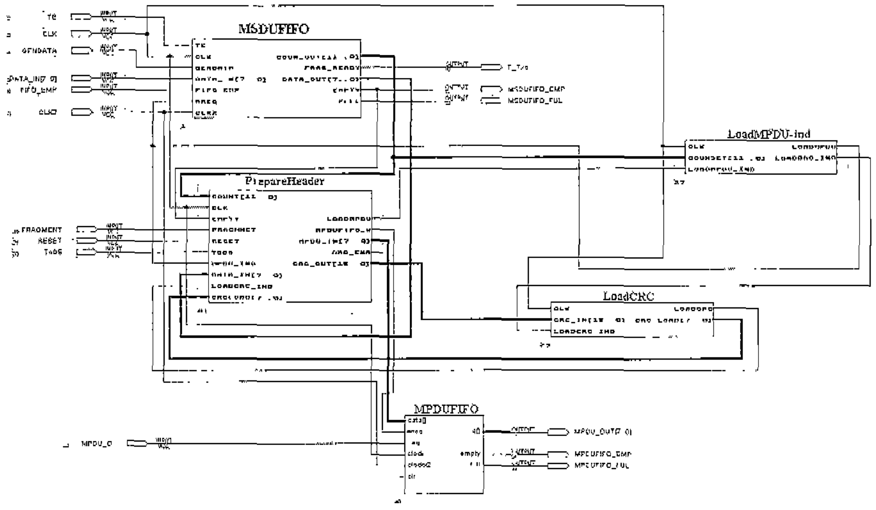
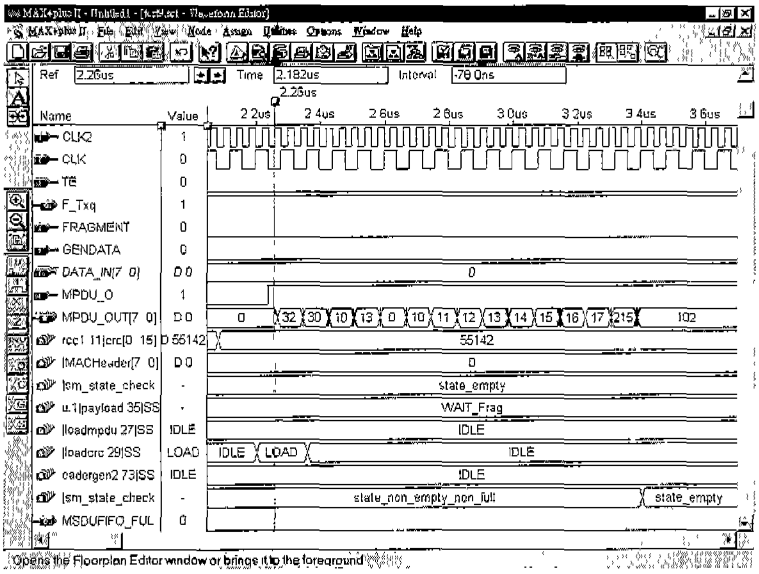


Fig. 10. The fragment function implementation.


MAX+plus II - Untitled1 - [tx.scf - Waveform Editor]
MAX+plus II File Edit View Node Assign Utilities Options Window Help
2.2us 2.4us 2.6us 2.8us 3.0us 3.2us 3.4us 3.6us
0
0 32 30 10 13 0 10 11 12 13 14 15 16 17 215 102
55142
0
state_empty
WAIT_Frag
IDLE
IDLE LOAD IDLE
IDLE
state_non_empty_non_full state_empty

Ref 2.26us Time 2.182us Interval -78.0ns

2.26us

Name	Value
CLK2	1
CLK	0
TE	0
F_Txq	1
FRAGMENT	0
GENDATA	0
DATA_IN[7 0]	D0
MPDU_O	1
MPDU_OUT[7 0]	D0
rcc1 11|crc[0 15]	D 55142
|MACHeader[7 0]	D0
|sm_state_check	-
u.1|payload 35|SS	-
|loadmpdu 27|SS	IDLE
|loadcrc 29|SS	LOAD
cadergen2 73|SS	IDLE
|sm_state_check	-
MSDUFIFO_FUL	0

Opens the Floorplan Editor window or brings it to the foreground

Fig. 11. The timing diagram of transmitting the MPDU frame.

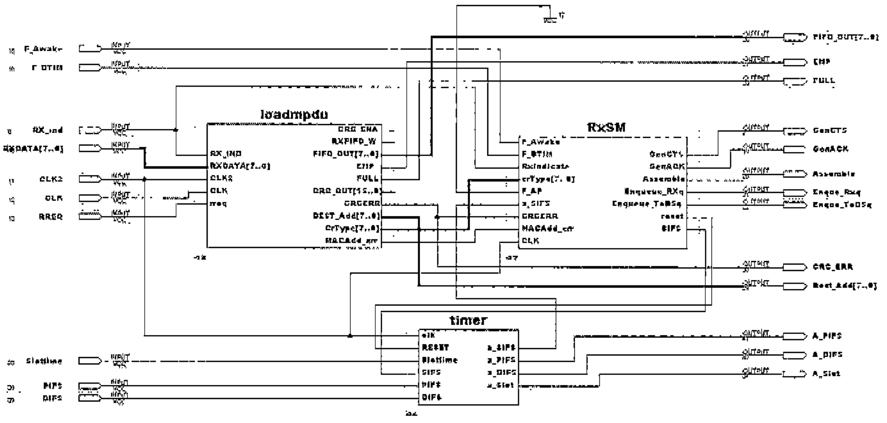


Fig. 12. The functional block of the receiver unit.

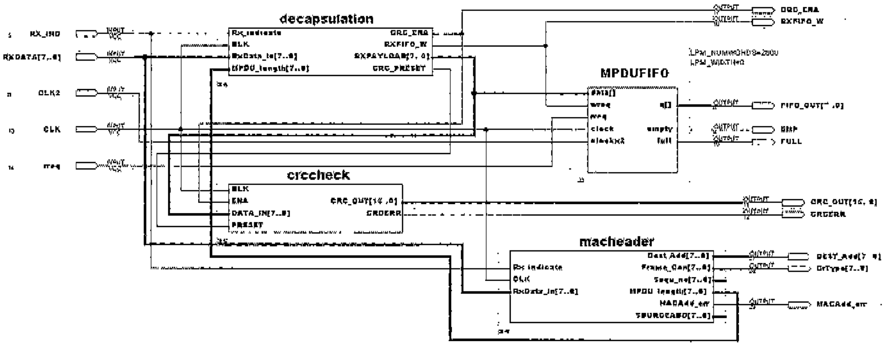


Fig. 13. The loadmpdu functional block.

V. THE RECEIVER UNIT DESIGN

As shown in Fig. 12, there are three core parts in the receiver unit: the [loadmpdu], the [timer], and the [receiver state machine (RxSM)]. The RxSM has to operate according to the 802.11 standard. For example, when receiving the data frame correctly, it will respond with an acknowledgment. Or if handshake of request-to-send (RTS) and clear-to-send (CTS) signals is desired (optional in 802.11), then after receiving an RTS when the station is idle for receiving, it has to command the transmission control state machine (txsm) to issue the CTS short frame. On the other hand, it may receive the CTS signal after sending the RTS. Then it will inform the txsm to transmit the first MPDU frame. Another example is when it receives the acknowledgment of the previously transmitted MPDU, it will inform the Txsm to continue sending the MPDU process. When a packet frame is received, the MAC header and the data field will be extracted separately. Then it will analyze the MAC header and check its destination address. If it matches this station's address, the normal receiving procedure is then activated, including analyzing the frame control field, deciding the frame data type, and other related process for the RxSM. At the same time, CRC check will continue to the end of reception, or until it detects an error. In case of an error, the CRCERR pin will be set to 1 to indicate the error and stop the receiving process. The loadmpdu function in Fig. 12 is shown in more details in Fig. 13, where it includes the decapsulation of the MPDU packet, CRC check, the MAC header analysis, and the loading into the MPDU FIFO.

In case that the MAC address does not correspond to its own address, the ToDS field (1 bit) in the frame will be checked. If the ToDS is 1, then it is destined to some other station in the distribution system through this station (access point). In this case, packet filtering and conversion has to be performed and is described in the following Section and in more details in [2]–[15]. While if the ToDS field is 0, then the MACAdd_err pin will be set to 1 to indicate wrong MAC address and the frame must hence be discarded. To verify our design, we note that when the physical layer receives data frame, it will inform the MAC layer via the RX_ind pin. The MAC layer will take out the data via the RXDATA[7...0] data bus and extract the MAC header and payload for further processing. During the receiving process, the CRC will check the frame continuously. If an error is detected, the CRCERR will be active. Otherwise, it will be kept in an inactive state. The timing diagram of MPDU frame reception is verified in Fig. 14, where it shows that the MPDU transmitted in Fig. 11 is correctly received and the CRC check also comes out correct subsequently.

VI. THE DS INTERFACE AND DEVICE DRIVERS

A. The DS Interface

In the case that the station is also an access point, the DS interface part shown in Fig. 1 has to be implemented based on which kind of DS is chosen. According to the standard, the DS may be

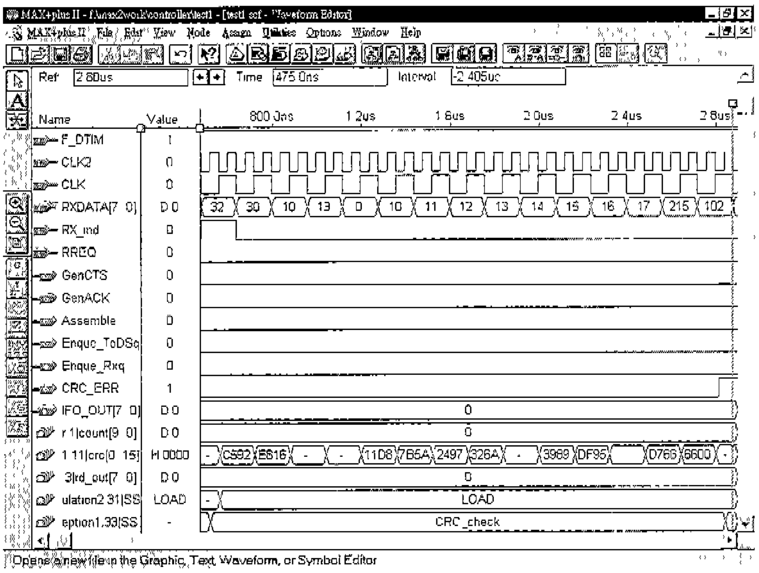


Fig. 14. The timing diagram of receiving the MPDU frame.

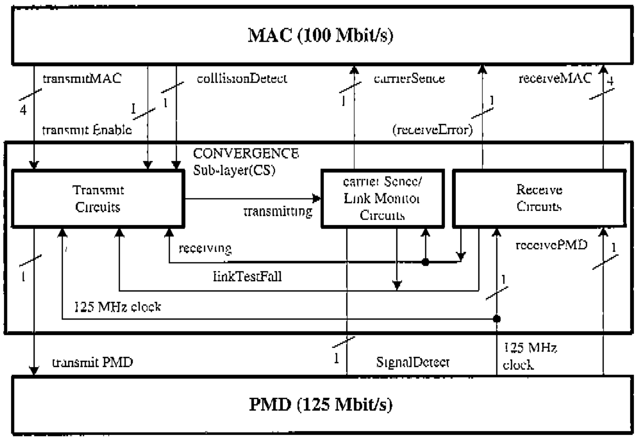


Fig. 15. The Fast Ethernet layered data flow.

another wireless medium, an 802.X, or another wired network. The port station that connects the 802.11 LAN and other wireless LAN or 802.X is sometimes called the access point, while if it is connected to other wired networks such as X.25 or FDDI, it is then called a portal. The distributed system can thus extend the wireless LAN to other types of networks. A natural thought of application is to consider the 100BaseX Fast Ethernet as the distributed system and the bridging station is called an access point. The Fast Ethernet has improved the transmission speed of 10BaseT from 10 Mbps to 100Mbps while preserving the same MAC protocol and the frame format. The data flow of Fast Ethernet is shown in Fig. 15, where the convergence sublayer (CS) is not defined in 10BaseT. The convergence sublayer is the primitive to interface the CSMA/CD MAC sublayer and the physical media dependent (PMD) sublayer. Also, clock encoding, such as Manchester code or differential Manchester code, is not suitable due to the fact that high-speed transmission may often cause transmission error. Instead, it utilizes the bit encoding method to guarantee a synchronous receiving because every symbol after encoding has enough signal conversion. The bit encoding used in 100BaseX Fast Ethernet is the so-called 4B/5B encoding in which the (J,K) symbols and (T,R) symbols are used to indicate the frame start space and frame stop space respectively.

As shown in Fig. 15, the top layer is the Ethernet MAC controller that takes charge of transmission and receiving process specified in the Fast Ethernet protocol. The lower part is the PMD layer which is intended to support different physical transmission media such as fiber optic, unshielded twisted pair, and shielded twisted pair. The middle layer is the convergence sublayer (CS). It consists of several digital logic circuits and is responsible for frame fragmentation, including MAC header and tail formation and extraction, clock encoding/decoding, and frame check sequence (FCS). The CS is the core part of Fast Ethernet and its FPGA implementation has been shown in [9]. Here we will emphasize on the packet filtering between 802.11 and the Fast Ethernet.

A.1 Packet Filtering and Protocol Conversion

The frame formats between 802.11 and other distributed systems are usually different. This section will describe the necessary steps for the frame format conversion. The 802.11 MAC data frame format, including the detailed frame control field and address field, and the Fast Ethernet MAC frame format are shown in Table 1. Two bits of (ToDs, FromDS) are used to indicate the traffic flow of the packet. To take a more general case for instance, we assume four address fields to identify the MAC address. The WDA and WSA are the wireless destination address and source address respectively. The BSSID is used to verify the original or destined basic service set. The DSDA and DSSA are the destination address and source address in the distributed system respectively. The RA is the next immediate receiver's address in the DS and the TA is the transmitter's address that is transmitting this frame. We assume that the access point has set up a (wireless) MAC address table for association management. There are five cases in terms of traffic flow: (1) flow

Table 1. The packet format comparison.

802.11 MAC DATA FRAME FORMAT						
Frame Control	Duration/ ID	Address Field	Length Byte	Sequence Number	Data Field	CRC-32 CRC-32

THE ADDRESS FIELD IN 802.11					
Tods	FromDS	Address 1	Address 2	Address 3	Address 4
0	0	WDA	WSA	BSSIDDA	BSSIDSA
0	1	WDA	BSSIDDA	DSSA	N/A
1	0	BSSIDSA	WSA	DSDA	N/A
1	1	RA	TA	DSDA	DSSA

THE FAST ETHERNET MAC DATA FRAME FORMAT						
Preamble	SFD SFD	Destination Address	Source Address	Length Field	Data Field	FCS FCS(CRC-32)

within the same BSS, (2) flow from one BSS to another BSS, (3) flow within the DS, (4) flow from one BSS to the DS, (5) flow from the DS to some BSS. Only cases (4) and (5) involve the packet filtering process, which we describe as follows.

A.2 Wireless MAC Format to the DS MAC Format

When the wireless receiver receives a data frame, it first extracts the MAC header to check both the address field and the frame control field. If both bits in (ToDS, FromDS) are zero and the destination address matches , then it is destined to the wireless medium. After CRC checking, it informs the host for appropriate actions. If the value of ToDS is 1, then this packet is sent out from one of the wireless stations and is destined to some station within the DS. The packet filtering process will then be performed as follows.

(1) Separate the MAC header and payload in the wireless packet.
(2) Convert the wireless MAC header into the Fast Ethernet MAC header.
(3) Put the payload behind the transformed MAC header, then generate the FCS and append to the payload.
(4) Download to the data buffer of DS and inform DS control state machine for transmission.

A.3 DS MAC Format to the Wireless MAC Format

The procedure is similar to above description. In case that the destination address in the Fast Ethernet MAC header is not destined to this AP, the AP must check its own wireless MAC address table. If it matches one of the addresses in the address table, then it will perform the packet filtering process as follows:

(1) Separate the MAC header and payload in the Fast Ethernet MAC packet.
(2) Convert the Fast Ethernet MAC header into the 802.11 wireless MAC header.
(3) Put the payload behind the transformed MAC header, then generate the new CRC code and append it to the payload.
(4) Download it to the Tx-FIFO buffer of the wireless transmitter and inform wireless transmission control state machine for transmission.

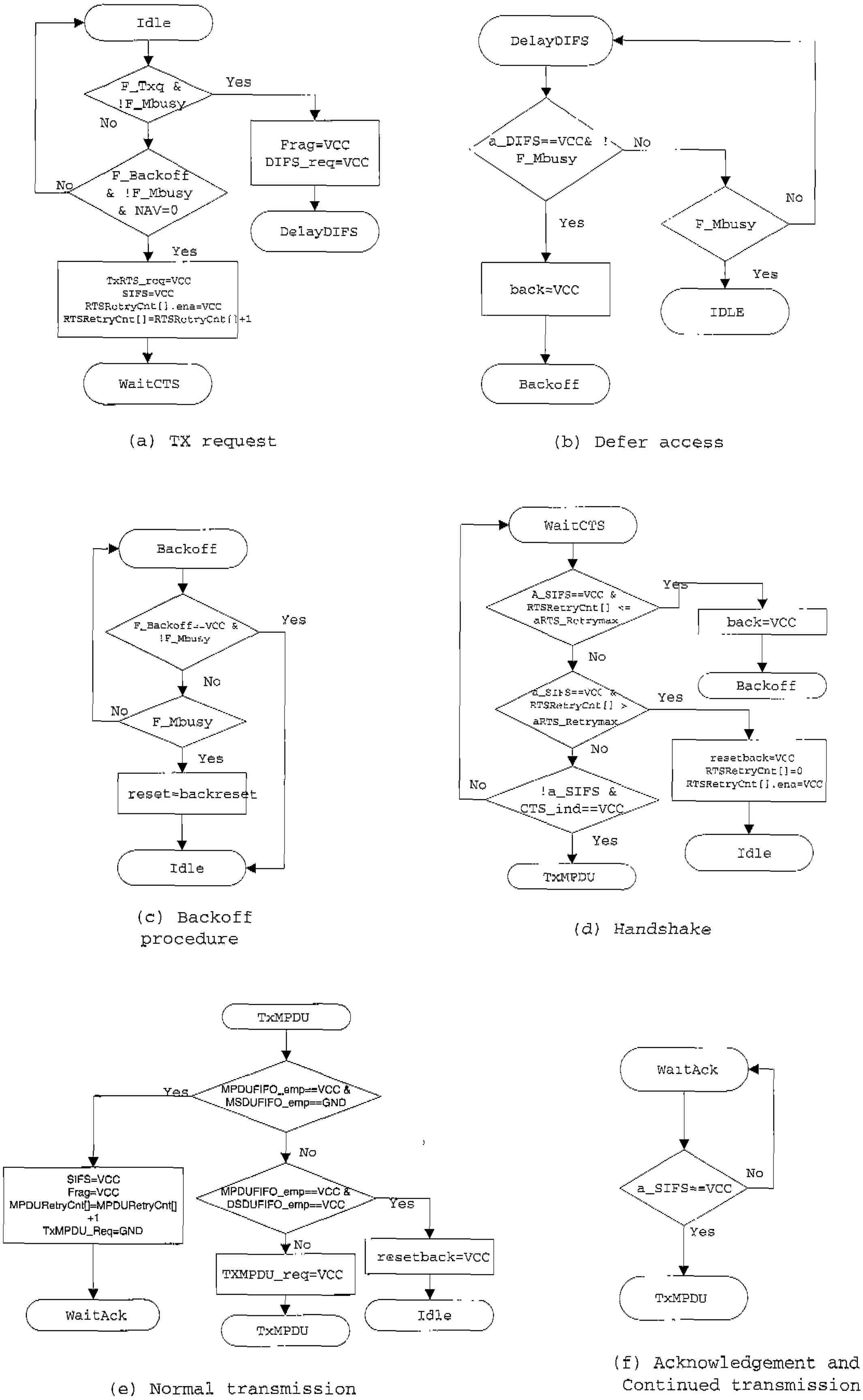
B. *The Device Drivers*

The device drivers can be programmed through the core microprocessor based on the MAC control state machine. As usual, microprocessors such as 8051 or 80196 are low-cost choices. However, considering that a station may also serve as an access point, one may also consider an asynchronous chip such as Motorola 680XX series that may better handle both the wireless MAC and DS MAC. Fig. 16 shows the flow chart of the MAC transmit control that can be fulfilled by our receiver unit. Fig. 16(a) shows that a station wants to send a packet, where it must sense the channel first. If the channel is idle, it will defer access by a DIFS time. After the DIFS time, it starts the backoff procedure. Once the backoff timer decreases to zero and the NAV is also clear (optional), it then sends the RTS signal and waits for CTS signal. Fig. 16(b) is the control for deferring access and initiating the backoff timer. Fig. 16(c) is the counting state of backoff timer, including exponential reset of the backoff timer. Fig. 16(d) shows the handshake procedure before transmission. Fig. 16(e) is the state in normal transmission, where when both the MPDU FIFO and MSDU FIFO are empty, it will wait for the acknowledgement frame. Fig. 16(f) is the action taken whether the acknowledgement is received or not.

As for the receive control, Fig. 17 shows its flow control. Fig. 17(a) shows the wake-up call from the inactive state (power saving). Fig. 17(b) is the state to confirm with the incoming packet by checking the delivery traffic indication message (DTIM) and the transmitter control state machine. If it is confirmed, then the CRC check will be initiated. Fig. 17(c) is the operation of CRC check. When CRC error is detected, it will trigger the negative acknowledgement, discard the packet and goes back to the idle state. If no error is detected, it then goes to the receiving process. Fig. 17(d) is the receiving process involved with our Receiver Unit that is specified in the MAC control state machine. Note that the PCF operation can also be implemented similarly as shown in [9].

VII. CONCLUSION

While the medium access control protocol of the 802.11 wireless LAN is complicated, many steps of the process can be rapidly realized through FPGA implementation as shown in this paper. From our experience, the cost-effective hardware circuits



(a) TX request

(b) Defer access

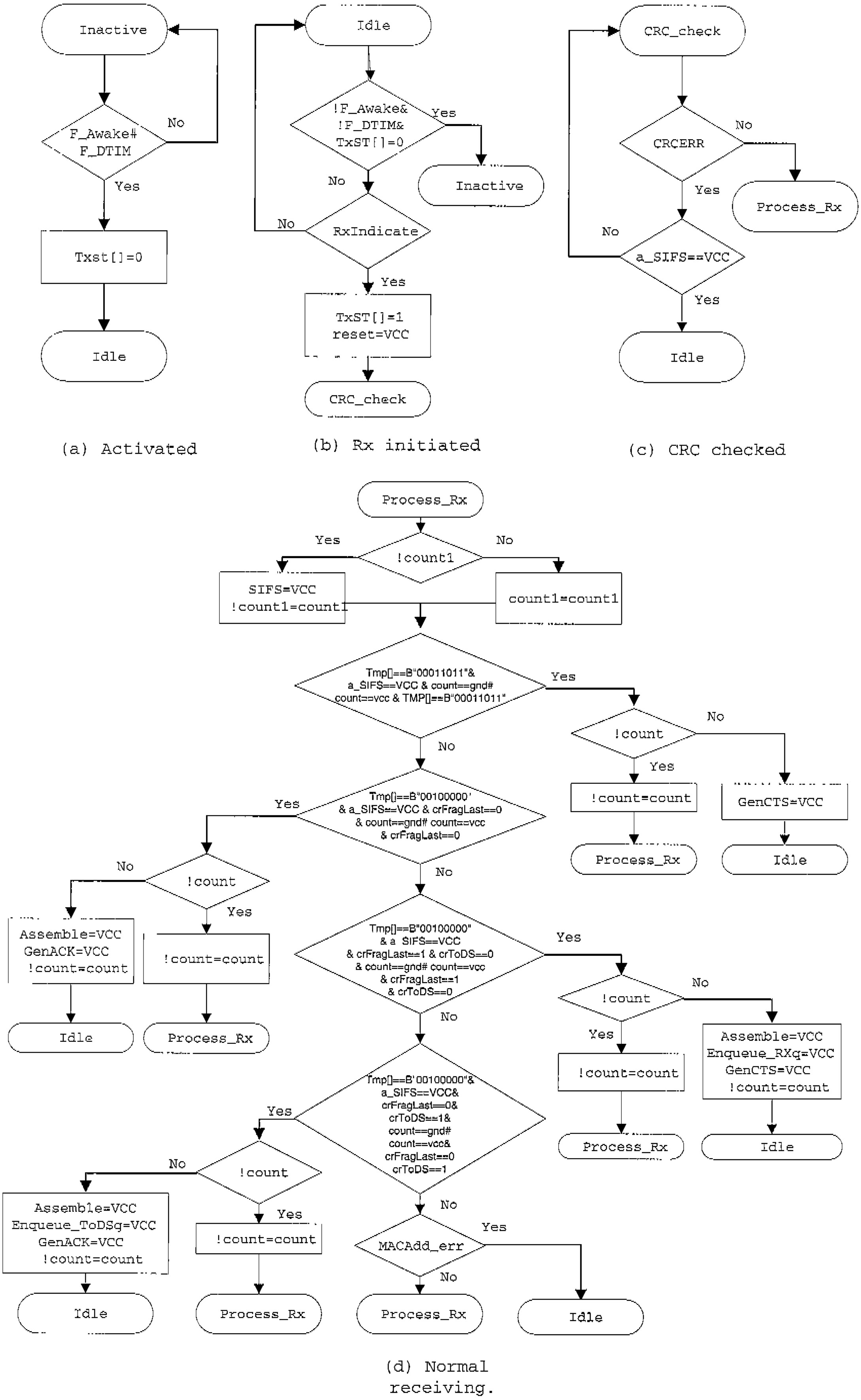
(c) Backoff procedure

(d) Handshake

(e) Normal transmission

(f) Acknowledgement and Continued transmission

Fig. 16. The flow chart of MAC transmit control.



(a) Activated

(b) Rx initiated

(c) CRC checked

(d) Normal receiving.

Fig. 17. The flow chart of MAC receive control.

can ease the realization of MAC control function and is also a trend in future products. We also note that integrating wireless LANs with other wired networks will be favorable in terms of commercial needs [10], [14]–[16]. The portal design for interfacing to the ATM networks will be our future goal.

ACKNOWLEDGEMENTS

The author would like to thank Taiwan National Science Council for financing this research project, Mr. Michael C. Wu, CEO of Telephony & Networking Communication Equipment, Inc., Taiwan for the discussion and support, and Professor C. Chen and Professor L. Lin in the Department of Electronics Engineering, National Yunlin University of Science and Technology for their unselfish comments

REFERENCES

[1] IEEE Standards Dept., Draft Std. IEEE 802.11, "Wireless LANs (P802.11 D6)," 1997.

[2] S. Jiang, "Study and design of a CDMA wireless LAN: Transceiver, interface, and protocol," *NSC Project Report*, NSC86-2212-E224-009, Taiwan, R.O.C., Oct. 1997.

[3] C.-C. Chang, *Research and Implementation of the Wireless LAN Protocol*, MS Thesis, National Yunlin University of Science and Technology, June 1997.

[4] S. Jiang and M.-T. Hsiao, "Performance evaluation of a receiver-based handshake protocol for CDMA networks," *IEEE Trans. Commun.*, vol. 43, no. 6, pp. 2127–2138, June 1995.

[5] M. White, M. Waller, G. Dunnett, and P. Lister, "FPGA design with Mentor and Altera CAD Software," *IEE Colloquium on Software Support and CAD techniques for FPGAs*, London, UK, 1994, pp. 1/1–1/5,

[6] M. Chang, "From VHDL to CPLD — a synthesizable journey," one-day tutorial workshop, in *Proc. of IEEE International ASIC Conference*, Austin, Texas, Sept. 21, 1995.

[7] Altera Target Applications, "Implementing CRCCs in altera devices." Ver. 1, July 1995.

[8] Altera MAX+PLUS II— User's manual.

[9] G. Shieh, *Design of an Access Point for Wireless Local Area Network*, MS Thesis, National Yunlin University of Science and Technology, June 1998.

[10] S. Jiang and G. Shieh, "Rapid prototyping of network interface for access points in WLANs," *Chunghwa Telecom Technical Quarterly*, vol. 2, no. 4, pp. 343–359, ISSN 0258-0284, Nov. 1998.

[11] S. Jiang and W.-D. Weng, "An evaluation model for integrated services on wireless broadband CDMA networks," *Wireless Personal Commun., an International J., Special issue on Wireless Broadband Commun.*, Kluwer Academic Publisher, vol. 10, no. 1, pp. 137-153, June 1999.

[12] D. Banks and M. Prudence, "A high-performance network architecture for a PA-RISC workstation," *IEEE Trans. Commun.*, vol. 11, no. 2, pp. 191-201, Feb. 1993.

[13] V. Jacobson, "Efficient protocol implementation," in *Proc. of ACM SIGCOMM'90*, Sept. 1990.

[14] S. Jiang and G. Shieh, "Design of a wireless access interface card for fieldbus," in *Proc. of the 5th International Conference on Automation Technology (AT'98)*, Taipei, Taiwan, R.O.C., July 20, 1998, D1-4.

[15] S. Jiang and S. Wang, "A novel wireless access protocol for factory automation networks," in *Proc. of the 5th International Conference on Automation Technology (AT'98)*, Taipei, Taiwan, R.O.C., July 20, 1998, B1-4.

[16] S. Jiang, "Wireless communications and a priority access protocol for multiple terminals in factory automation," *IEEE Trans. Robotics and Automation*, vol. 14, no. 1, pp. 123–136. Feb. 1998.

[17] S. Jiang and C. C. Chang, " Performance evaluation of an improved MAC protocol for 801.11 WLANs," *J. Technology*, National Taiwan University of Science and Technology (copyrighted by the R.O.C. Educational Department), Taiwan, vol. 14, no. 2, pp. 123–136, June 1999.

Songchar Jiang was born in Chaiyi, Taiwan, R.O.C. He received the B.S.E.E. degree from Tamkang University, Taipei, Taiwan, in 1982, the M.S. degree from the state University of New York at Buffalo in 1987, and the Ph.D. degree from the School of Electrical Engineering of Purdue University, West Lafayette, IN, in 1991. Between 1982 and 1984, he served with the China Air Force as a Second Lieutenant specializing in military communications. From 1984 to 1985, he was with the R&D Department of Sampo Electronics, Taiwan. Since 1992 he has been an Associate Professor at the National Yunlin University of Science and Technology (formerly National Yunlin Institute of Technology). He has led a research team conducting several research projects in wireless access technologies since 1994. His current research topics include wireless access systems and protocols, mobile computing, factory automation, HFC networks, and performance evaluation of computer and telecommunication Networks. Dr. Jiang is a member of Eta Kappa Nu.