RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

PM7350

S/UNI-DUPLEX ™

DSLAM

REFERENCE DESIGN

WAN CARD SOFTWARE

RELEASED

ISSUE 2: NOVEMBER 2000

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

CONTENTS

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

1   INTRODUCTION

1.1   Scope

This document introduces the reader to the software components of the DSLAM WAN card. It consists of two main sections: the Users' Guide and the Developers' Guide. The Users' Guide discusses setting up and running the software on the DSLAM WAN card. The Developers' Guide provides details of the software to enable modification and further development.

1.2   Target Audience

This document has been prepared for all users of PMC-Sierra reference design boards that need to understand the software/firmware running on the DSLAM WAN cards. To learn more about the WAN Card hardware, please refer to "DSLAM Reference Design: WAN Card [4]". For an overview of the DSLAM System, please refer to "DSLAM Reference Design: System Design [5]".

1.3   Objectives

- To provide an overview conceptual understanding of the DSLAM WAN card software components.

- To document the setup required to run the software.

- To document the user interface of the software.

- To document the software design thereby allowing it to be maintained.

- To direct the user to other task-specific documents.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

2   USERS' GUIDE

2.1   Overview

The following files are provided:

- The source code of the firmware. (See Developers' Guide section for complete list and description.)

- compile.bat – batch file for compiling and linking the source code.

- MAIN.OBJ – object file that can be loaded to a BDM (Background Debug Mode) emulator.

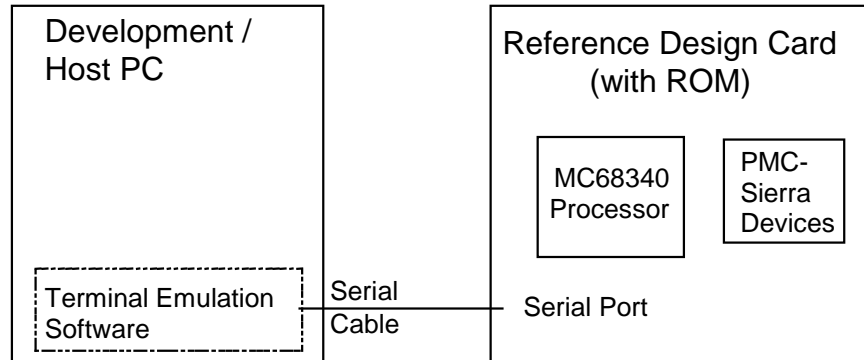- main.s19 – Motorola S-record file that can be burned into a ROM.

2.2   Setup

The DSLAM WAN card software is designed to run under two different setups: ROM (Read Only Memory) and BDM (Background Debug Mode). If modification of the source code is not required, ROM setup allows easy setup with minimal hardware. For further development, the BDM setup is recommended. Both setups are described in detail below.

2.2.1 ROM

If modification of source code is not required, the ROM setup is used. Figure 1 illustrates a typical configuration for running the firmware in ROM setup. In this case the software is burned into the ROM and the user cannot modify the software. Jumper J27 (ROM/BDM SELECT) should be open for ROM setup.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Figure 1: Typical User Configuration

The Motorola S-record file provided, "main.s19" can be programmed to the ROM directly and no compiling is required.

2.2.2 BDM

Figure 2 illustrates an example test-bench required to test and modify the firmware. Jumper J27 (ROM/BDM SELECT) should be shorted for BDM setup. The firmware may need modification to add extra functionality or implement new services required by the user and user specific application software. The development PC should contain the following development tools:

- Nohau In-Circuit Emulator (or other Background Debug Mode emulator)

- Introl-C Compiler/Linker (or other suitable C compiler/linker)

Figure 2: Development Testbed

RELEASED

APPLICATION NOTE

PMC-2000684

PMC *PMC-Sierra, Inc.*

PM7350 S/UNI-DUPLEX

ISSUE 2   *DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL*
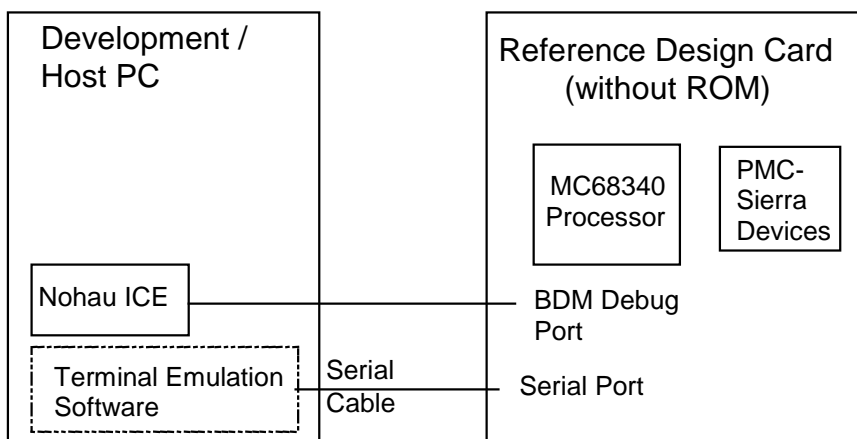
The next section describes how to compile the source code to load to the Nohau In-Circuit Emulator. Please refer to the Nohau documentation for details on how to load the compiled code.

2.2.3 Compilation

In order to compile the source code, Introl-C compiler/linker must be installed in the system and included in the search path. The batch file "compile.bat" is provided and executing it will produce the appropriate object files and Motorola S-record files for BDM or ROM setup.

The linker command file "c68.ld" provided is targeted for BDM setup. In order to generate a Motorola S-record file for ROM setup after development with BDM setup, the file "c68.ld" must be modified. The line that reads:

```
let __RAMstart = 0xE000; /* address at which to map RAM */
```

needs to be modified to:

```
let __RAMstart = 0x100000; /* address at which to map RAM */
```

2.2.4 Serial Port connection

The user interface for the DSLAM WAN Card software is provided by the serial port of the "DSLAM WAN Card" reference design card. Typically this port is attached to a terminal via a serial cable. The terminal provides a keyboard interface to enter commands directly, or a terminal emulation program that allows a file of commands to be sent to the serial port. The serial port should be set up to 9600bps, 8 data bits, no parity and 1 stop bit.

The serial cable connects the serial port of the host computer to the RJ-45 connector of the WAN Card and has the following pin out:

DB-9 pin #	signal name	RJ-45 pin #
1	Not Connected	
2	RXD	3
3	TXD	6
4	Not Connected	
5	GND	4
6	Not Connected	
7	RTS	8
8	CTS	1
9	Not Connected	

RELEASED

APPLICATION NOTE

PMC-2000684

PMC *PMC-Sierra, Inc.*

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

2.3   User Interface

The commands are text characters that can be created via a DOS or Windows editor such as notepad, or entered directly at the terminal. Each command must be delimited by the carriage return and line feed characters. These characters are inserted by typing the return key and are not visible within an editor.

Blank lines are valid and are ignored by the command parser.

Comments may be inserted into a script file by placing the asterisk character in the first column of each line that contains comments.

Fields of a command are delimited by an arbitrary number of blank characters (spaces) or tab characters. The fields of a command must be entered in the order specified and as described below:

- A field with no brackets implies that the text is to be typed exactly as shown.

- <> brackets around a field imply that the textual representation of a number or character string must be entered in the field.

- [ ] brackets around a field, or fields, imply that the field(s) is optional.

- | separates parameters that are mutually exclusive

2.3.1 Command Summary

The following list of commands are defined in the following sections:

- act

- clear_stat

- count <device>

- deact

- init [<device>]

- help

- loopback <device> <type> <channel> <enable|disable>

- read <device> <register>

RELEASED

APPLICATION NOTE

PMC-2000684

PMC *PMC-Sierra, Inc.*

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

- reset [<device>]

- stat

- write <device> <register> <data>

2.3.2 Detailed Command List

The following pages provide a detailed summary for each of the commands.

2.3.2.1 act

The activate command activates the S/UNI-DUPLEX device for normal operation. The S/UNI-DUPLEX device must be initialized by the "init" command before activation. [Note: The S/UNI-QJET device is automatically activated at initialization.]

Syntax:

act

2.3.2.2 clear_stat

The clear statistic command is issued to clear the accumulated history of interrupts and other statistics of the S/UNI-DUPLEX device

Syntax:

clear_stat

2.3.2.3 count

The display counters command is used to display the accumulated counters since the last time this command was issued or, if the command was never issued before, since the device was activated. The counters can be displayed for one specified device at a time.

Syntax:

count <device>

Parameter Description:

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

<device> selects the hardware device which is represented by typing one of the following character strings without the quotation marks.

Character String	Device
"duplex"	The S/UNI-DUPLEX chip onboard (S/UNI-DUPLEX device must be activated)
"qjet"	The S/UNI-QJET chip onboard

Output:

These are the counters that the command will display for the S/UNI-DUPLEX device:

- Serial Link 1 Cells Received Count

- Serial Link 2 Cells Received Count

- Serial Link 1 HCS Errors Count

- Serial Link 2 HCS Errors Count

- Serial Link Cells Transmitted Count

These are the counters that the command will display for the S/UNI-QJET device:

- Framing Bit Error Count

- LCV Count

- Rx Cell Count

- Tx Cell Count

Example:

```
* display accumulated counts for the qjet device

count qjet
```

2.3.2.4 deact

The deactivate command deactivates the S/UNI-DUPLEX from normal operation.

Syntax:

RELEASED

APPLICATION NOTE

PMC-2000684

PMC *PMC-Sierra, Inc.*

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

deact

2.3.2.5 init

The initialize command configures the devices. [Note: All devices are automatically initialized and activated upon startup or full reset.]

Syntax:

init [<device>]

Parameter Description:

<device> selects the hardware device(s) which is represented by typing one of the following character strings without the quotation marks. When this option is not specified all devices are selected.

Character String	Device
"duplex"	The S/UNI-DUPLEX chip onboard
"qjet"	The S/UNI-QJET chip onboard

Example:

```
* initialize all devices

init

* initialize the QJET device

init qjet
```

2.3.2.6 help

Displays help information. The command will list all the commands available on the interface and provide a brief description of what each command does.

Syntax:

help

2.3.2.7 loopback

The loopback command enables/disables loopbacks for S/UNI-DUPLEX and S/UNI-QJET devices. Please refer to the S/UNI-DUPLEX data sheet [2] and

S/UNI-QJET data sheet [1] for the functionality of the different modes of loopback.

Syntax:

loopback <device> <type> <channel> <enable|disable>

Parameter Description:

<device> selects the hardware device which is represented by typing one of the following character strings without the quotation marks.

Character String	Device
"duplex"	The S/UNI-DUPLEX chip onboard
"qjet"	The S/UNI-QJET chip onboard

<type> Selects the type of loopback which is represented by typing one of the following character strings without the quotation marks.
Note: QJET loopback modes are mutually exclusive. Setting the QJET to one loopback mode will disable the other modes.

Character String	Device
"line"	Line loopback
"diag"	Diagnostic loopback
"payload"	Payload loopback (valid for S/UNI-QJET only)
"disable"	Disable loopback (valid for S/UNI-QJET only)

<channel> Selects a channel of the device, which is represented by typing one of the following character strings without the quotation marks.

Character String	Device
"RxD1"	Receive Serial Channel 1 (Duplex)
"RxD2"	Receive Serial Channel 2 (Duplex)
"C1"	Channel 1 (QJET)
"C2"	Channel 2 (QJET)
"C3"	Channel 3 (QJET)
"C4"	Channel 4 (QJET)

<enable|disable> Enables or disable loopback for the specified serial link (required for S/UNI-DUPLEX; not valid for S/UNI-QJET).

Example:

```
* enable line loopback on serial link1 of the duplex device
```

RELEASED

APPLICATION NOTE

PMC-2000684   ISSUE 2   DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

PMC-Sierra, Inc.   **PM7350 S/UNI-DUPLEX**

```
loopback duplex line RxD1 enable

* disable diagnostic loopback on serial link2 of

*    duplex device

loopback duplex diag RxD2 disable

* enable payload loopback channel 1 of QJET

loopback qjet payload c1

* disable loopback on channel 2 of QJET

loopback qjet disable c2
```

2.3.2.8 read

The read command causes the MC68340 to issue a read to the external device register. The 8-bit value read from the external device is displayed.

Syntax:

read <device><register>

Parameter Description:

<device> selects one of the hardware devices which is represented by typing one of the following character strings without the quotation marks.

Character String	Device
"duplex"	The S/UNI-DUPLEX chip onboard
"qjet"	The S/UNI-QJET chip onboard
"abs"	No device. <register> provides the absolute address Note: The user must ensure that the address corresponds to RAM/ROM or a device.

<register> is a hex value representing the register offset.

Example:

```
* read register 0x1B of the QJET device

* and register 0x06 of the Duplex device
```

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

```
read qjet 1B

read duplex 6
```

2.3.2.9 reset

The reset command does a software reset on the selected device(s). When all devices are selected (by omitting the device parameter), the firmware is also restarted. A device will have to be initialized manually after a device specific reset before further use. [Note: All devices are automatically initialized and activated upon startup or full reset.]

Syntax:

reset [<device>]

Parameter Description:

<device> selects the hardware device(s) which is represented by typing one of the following character strings without the quotation marks. When this option is not specified all devices are selected.

Character String	Device
"duplex"	The S/UNI-DUPLEX chip onboard
"qjet"	The S/UNI-QJET chip onboard

Example:

```
* reset all devices and re-start software

reset

* reset the QJET device

reset qjet
```

2.3.2.10   stat

The "stat" command displays the accumulated statistical counts and interrupts since the last time it was cleared using the "clear_stat" command. The information displayed is related to the S/UNI-DUPLEX device only.

Syntax:

stat

RELEASED

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

APPLICATION NOTE

PMC-2000684

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Output:

The following statistics of the S/UNI-DUPLEX device will be displayed:

- Serial Link Transmit Overflow Count Status

- Serial Link Transmit Updated

- Receive Logical Channel FIFO Overflow

- Transmit Logical Channel FIFO Overflow

- Phy Input Cell Transferred

- Invalid SOC Sequence

- Phy Input Parity Error

- Phy Output Error

- Microprocessor Insert FIFO Ready

- Microprocessor Insert FIFO Full

- Extract Cell CRC Error

- Clock Lock Fail

- Receive Serial Channel Out of Delineation (for each Channel)

- Receive Serial Channel In Delineation (for each Channel)

- Receive Serial Channel FIFO Overrun (for each Channel)

- Receive Serial Channel HCS Error (for each Channel)

- Receive Serial Channel Out of Sync (for each Channel)

- Receive Serial Channel In Sync (for each Channel)

- Receive Serial Link Extract FIFO Overflow (for both links)

- Receive Serial Link Loss of Signal (for both links)

- Receive Serial Link Signal Detected (for both links)

- Receive Serial Link Out of Delineation (for both links)

- Receive Serial Link In Delineation (for both links)

- Receive Serial Link Active (for both links)

- Receive Serial Link Not Active (for both links)

- Receive Serial Link Out of Sync (for both links)

- Receive Serial Link In Sync (for both links)

- Receive Serial Link CRC8 Error (for both links)

- Receive Serial Link HCS Error (for both links)

- Receive Serial Link Counter Updated (for both links)

- Receive Serial Link Counter Overflow (for both links)

- Receive Serial Link Valid BOC (for both links)

- Extract FIFO ready

- Number of Interrupts occurred

2.3.2.11   write

The write command causes the MC68340 to write an 8-bit value to a register of an external device.

Syntax:

write <device> <register> <value>

Parameter Description:

<device> selects one of the hardware devices which is represented by typing one of the following character strings without the quotation marks.

Character String	Device
"duplex"	The S/UNI-DUPLEX chip onboard
"qjet"	The S/UNI-QJET chip onboard
"fpga"	The FPGA onboard
"abs"	No device. <register> provides the absolute address Note: The user must ensure that the address corresponds to RAM/ROM or a device.

<register> is the register offset within the device, a hex value.

<value> is the 8-bit hex value written to the external device.

Example:

```
* write 0xFF to register 0x08 of the duplex device

* and register 0x00 of qjet

write duplex 8 FF
```

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

```
write qjet 0 FF
```

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL
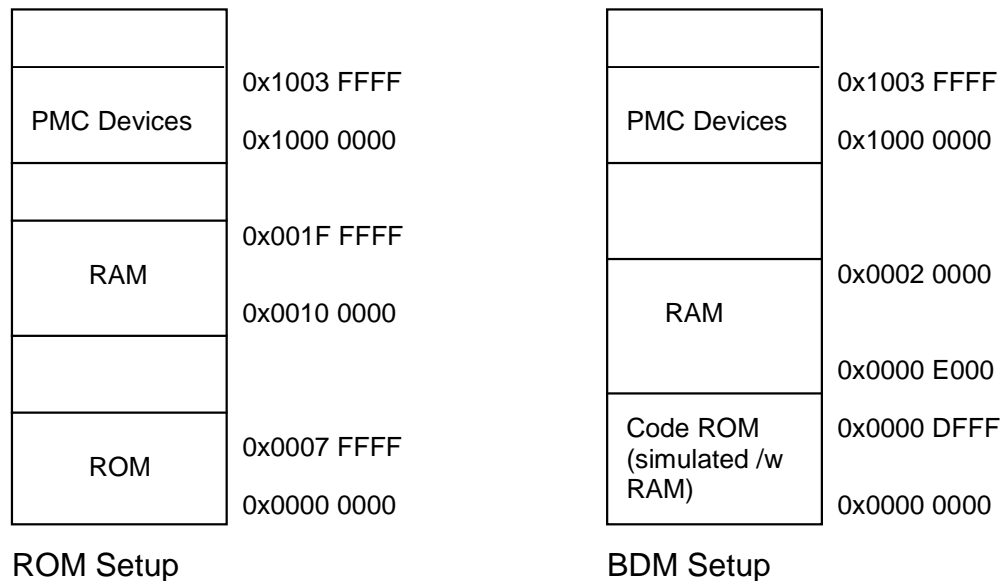
3   DEVELOPERS' GUIDE

3.1   Memory

Figure 3 depicts the memory map for both the ROM Setup and the BDM setup.

Figure 3: Memory Map

PMC Devices	0x1003 FFFF
	0x1000 0000
RAM	0x001F FFFF
	0x0010 0000
ROM	0x0007 FFFF
	0x0000 0000

ROM Setup

PMC Devices	0x1003 FFFF
	0x1000 0000
RAM	0x0002 0000
	0x0000 E000
Code ROM (simulated /w RAM)	0x0000 DFFF
	0x0000 0000

BDM Setup

The following are the base addresses of each of the devices in the PMC Devices region:

S/UNI-QJET   0x1001 0000
S/UNI-DUPLEX 0x1002 0000
FPGA   0x1003 0000

If new features are added, the RAM and ROM requirements of the new firmware might increase. If such requirements exceed the ranges provided, the new firmware will fail to execute. Please check the memory map file "main.m68" which is created at linking, to make sure code and data is mapped within the provided ranges.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

3.2   Source Code Description

3.2.1 Start Up Main Module

The file "__main.c" has only one function that simply receives a call from start.s and calls the main function in "main.c"

3.2.1.1 __main: Main Module Caller

Call the main module.

Prototype   `void __PASCAL __main(void)`

Input   None

Output   None

Return Codes   None

3.2.2 System Functions

The files "_system.c" and "_system.h" provide functions to initialize the system and manage interrupts.

3.2.2.1 DisableExternalInerrupts: Disable Interrupts

Disable interrupts so execution of the program will not be stopped.

Prototype   `void DisableExternalInterrupts(void)`

Input   None

Output   None

Return Codes   None

3.2.2.2 EnableExternalInerrupts: Enable Interrupts

Enable Interrupts.

Prototype   `void EnableExternalInterrupts(void)`

Input   None

Output   None

Return Codes   None

3.2.2.3 ReenableInerrupts: Re-Enable Interrupts

Re-enable interrupts if it has been disabled before.

Prototype   `void ReenableInterrupts(void)`

Input   None

Output   None

Return Codes   None

3.2.2.4 sysInitialize: Initialize System

Initialize the system by disabling timer and enabling interrupts.

Prototype   `void sysInitialize(void)`

Input   None

Output   None

Return Codes   None

3.2.3 Linker Input Command File

The file "c68.ld" configures the linker to generate the executable image. The files that need to be linked are listed in this file.

3.2.4 Command Parameter Handler

The files "com.c" and "com.h" handle the parameter of commands specific to the WAN card.

3.2.4.1 strtolower: Convert String to Lower Case

Convert any upper case characters in a string to lower case.

Prototype   `void strtolower(char *strTmp)`

RELEASED

APPLICATION NOTE

PMC-2000684

PMC *PMC-Sierra, Inc.*

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Input   `strTmp`: The string to be converted. This string is overwritten as it is used for output as well.

Output   `strTmp`: The string after conversion.

Return Codes None

3.2.4.2 findArg: Find Command Argument

Call the appropriate function based on the arguments passed in the command line.

Prototype `void findArg(char **szArg, ARG_ENTRY *aArgs)`

Input   `szArgs`: Array of arguments of the command.

      `aArgs`: List of acceptable arguments and the appropriate function to call.

Output   None

Return Codes None

3.2.4.3 notifyError: Argument Error Notification

Notifies the user that the arguments in the command have error(s). This is done by sending an error message through the serial port.

Prototype `void notifyError(void)`

Input   None

Output   None

Return Codes None

3.2.4.4 cmdActivate: Activate Command Handler

Handle the activate command by calling the Duplex driver API to activate the S/UNI-DUPLEX device.

Prototype `void cmdActivate(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Output   None

Return Codes None

3.2.4.5 hlpActivate: Activate Help Information

Display a short description of what the activate command does.

Prototype   `void hlpComment(void)`

Input   None

Output   None

Return Codes None

3.2.4.6 cmdClearStat: Clear Statistics Command Handler

Handle the clear statistics command by calling the Duplex driver API to reset all statistical counts of the S/UNI-DUPLEX device.

Prototype   `void cmdClearStat(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.7 hlpClearStat: Clear Statistics Help Information

Display a short description of what the clear statistics command does.

Prototype   `void hlpClearStat(void)`

Input   None

Output   None

Return Codes None

3.2.4.8 cmdCounters: Count Command Handler

Handle the count command.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Prototype   `void cmdCounters(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.9 hlpCounters: Count Help Information

Displays a short description of what the count command does.

Prototype   `void hlpCounters(void)`

Input   None

Output   None

Return Codes   None

3.2.4.10   CountersDuplex: Duplex Device Counters

Call the Duplex driver API to obtain counters stored in the S/UNI-DUPLEX device registers and display them.

Prototype   `void CountersDuplex(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.11   CountersQjet: QJET Device Counters

Display the counters in the QJET device.

Prototype   `void CountersQjet(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

3.2.4.12 cmdDeactivate: Deactivate Command Handler

Handle the deactivate command by calling the Duplex driver API to deactivate the S/UNI-DUPLEX device.

Prototype   `void cmdDeactivate(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.13 hlpDeactivate: Deactivate Help Information

Display a short description of what the deactivate command does.

Prototype   `void hlpDeactivate (void)`

Input   None

Output   None

Return Codes   None

3.2.4.14 cmdInit: Initialize Command Handler

Initialize command handler.

Prototype   `void cmdInit(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.15 hlpInit: Initialize Command Help Information

Display a short description of what the initialize command does.

Prototype   `void hlpInit(void)`

Input   None

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Output   None

Return Codes   None

3.2.4.16   InitAll: Initialize All Devices

Initialize all the S/UNI-DUPLEX and QJET devices.

Prototype   `void InitAll(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.17   InitDuplex: Initialize Duplex Device

Calls the Duplex driver API to initialize the S/UNI-DUPLEX device.

Prototype   `void InitDuplex(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.18   InitQjet: Initialize QJET Device

Call the QJET driver to initialize a specified QJET device.

Prototype   `void InitQjet(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.19   cmdLoopback: Loopback Command Handler

Handle the loopback command.

Prototype   `void cmdLoopback(char** szArgs)`

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.20   hlpLoopback: Loopback Command Help Information

Display a short description of what the loopback command does.

Prototype   `void hlpLoopback(void)`

Input   None

Output   None

Return Codes None

3.2.4.21   LoopbackDuplex: Duplex Device Loopback

Call the Duplex driver API to enable/disable loopback for the S/UNI-DUPLEX
device.

Prototype   `void LoopbackDuplex(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.22   LoopbackQjet: QJET Device Loopback

Call the QJET driver to enable/disable loopback for the QJET device.

Prototype   `void LoopbackQjet(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

3.2.4.23   cmdRead: Read Command Handler

Handle the read command.

Prototype   `void cmdRead(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.24   hlpRead: Read Command Help Information

Display a short description of what the read command does.

Prototype   `void hlpRead(void)`

Input   None

Output   None

Return Codes None

3.2.4.25   ReadDuplex: Duplex Device Read Register

Call the Duplex driver API to read a register from the S/UNI-DUPLEX device.

Prototype   `void ReadDuplex(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.26   ReadQjet: QJET Device Read Register

Call the QJET driver to read a register from the QJET device.

Prototype   `void ReadQjet(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Return Codes   None

3.2.4.27   ReadFPGA: FPGA Read Register

Display an error message that reading from FPGA is not supported in hardware.

Prototype   `void ReadFPGA(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.28   ReadAbsolute: Absolute Address Read

Read from an absolute address.

Prototype   `void ReadAbsolute(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.29   cmdReset: Reset Command Handler

Handle the reset command.

Prototype   `void cmdReset(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.30   hlpReset: Reset Command Help Information

Display a short description of what the reset command does.

Prototype   `void hlpReset(void)`

Input   None

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Output   None

Return Codes   None

3.2.4.31   ResetAll: Reset All Devices

Reset all the devices and re-start the firmware.

Prototype   `void ResetAll (char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.32   ResetDuplex: Reset Duplex Device

Call the Duplex driver API to do a software reset on the S/UNI-DUPLEX device.

Prototype   `void ResetDuplex(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.33   ResetQjet: Reset QJET Device

Call the QJET driver to reset all the QJET device.

Prototype   `void ResetQjet(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes   None

3.2.4.34   cmdStat: Statistics Command Handler

Handle the statistics command by calling the Duplex driver API to return all statistical counts of the S/UNI-DUPLEX device and displaying them.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Prototype   `void cmdStat(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.35   hlpStat: Statistics Help Information

Display a short description of what the statistics command does.

Prototype   `void hlpStat(void)`

Input   None

Output   None

Return Codes None

3.2.4.36   cmdWrite: Write Command Handler

Handle the write command.

Prototype   `void cmdWrite(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.37   hlpWrite: Write Command Help Information

Displays a short description of what the write command does.

Prototype   `void hlpWrite(void)`

Input   None

Output   None

Return Codes None

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

3.2.4.38   WriteDuplex: Duplex Device Write

Call the Duplex driver API to write to a register of the S/UNI-DUPLEX device.

Prototype   `void WriteDuplex(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.39   WriteQjet: QJET Device Write

Call the QJET driver to write to a register of the QJET device.

Prototype   `void WriteQjet(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.40   WriteFPGA: FPGA Write

Write to a register of the FPGA.

Prototype   `void WriteFPGA(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.4.41   WriteAbsolute: Absolute Address Write

Write to an absolute address.

Prototype   `void WriteAbsolute(char** szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Return Codes   None

3.2.5 QJET Drivers

The files "qjet.c" and "qjet.h" provide a simple driver for the QJET devices. The functions provided are discussed below.

3.2.5.1 qjetInit: Initializing QJET Device

This function initializes the QJET device.

Prototype   `int qjetInit(QJET pQjet)`

Input   `pQjet`: Base address of the QJET device to be initialized.

Output   None

Return Codes `QJET_SUCCESS` (Function was completed successfully)

`QJET_ERROR` (There was an error)

3.2.5.2 qjetLoopback: Setup/Disable Loopback Modes

This function enables a loopback mode or disables loopback for one channel based on the loopback type passed in.

Prototype   `int qjetLoopback(QJET pQjet, unsigned int uiChannel, unsigned char LoopbackType)`

Input   `pQjet`: Base address of the QJET device.

`uichannel`: Identifies the channel (1-4) that the loopback should be applied for.

`LoopbackType`: Identifies the mode of loopback, which can be one of:

- `QJET_LPBK_LINE`

- `QJET_LPBK_PAYLOAD`

- `QJET_LPBK_DIAG`

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

- `QJET_LPBK_DISABLE`

Output   None

Return Codes `QJET_SUCCESS` (Function was completed successfully)

                  `QJET_ERROR` (There was an error)

3.2.5.3 qjetRead: Reading from QJET Registers

This function reads the contents of a register of the QJET device.

Prototype   `int qjetRead(QJET pQjet, unsigned int uiRegId, unsigned char *value)`

Input   `pQjet`: Base address of the QJET device.

                  `uiRegId`: Address offset of the register

Output   `value`: Register value

Return Codes `QJET_SUCCESS` (Function was completed successfully)

                  `QJET_ERROR` (There was an error)

3.2.5.4 qjetReset: Resetting QJET Device

This function applies a software reset to the QJET device.

Prototype   `int qjetReset(QJET pQjet)`

Input   `pQjet`: Base address of the QJET device to be reset.

Output   None

Return Codes `QJET_SUCCESS` (Function was completed successfully)

                  `QJET_ERROR` (There was an error)

3.2.5.5 qjetWrite: Write to QJET Register

This function writes a value to a register of the QJET device.

Prototype   `int qjetWrite(QJET pQjet, unsigned int uiRegId, unsigned char value)`

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Input   `pQjet`: Base address of the QJET device to be initialized.

`uiRegId`: Address offset of the register

`value`: Register value

Output   None

Return Codes `QJET_SUCCESS` (Function was completed successfully)

`QJET_ERROR` (There was an error)

3.2.6 Duplex Drivers

The following files are part of the Duplex driver. The generic duplex drivers were ported to work under this specific environment. Please refer to the Duplex Device Driver Design Specification [3] for detail explanation.
dpx.c
dpx.h
dpx_api.c
dpx_api.h
dpx_err.h
dpx_hw.c
dpx_hw.h
dpx_rtos.c
dpx_rtos.h
dpx_test.c
dpx_test.h
sysPci.h

3.2.7 Event Manager Module

The files "events.c" and "events.h" implements the event manager module that is used to handle incoming commands.

3.2.7.1 evAddRecordToList: Add Event Record

Places an event record, whose fields have been appropriately filled in, at the tail of the queue of events to be processed.

Prototype   `int evAddRecordToList(EVENT_RECORD *pRecord)`

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Input   `pRecord`: Address of the event record to be queued.

Output   None

Return   Either 0 (Failure) or 1 (Success).

3.2.7.2 evInitialize: Event Module Initialization

Initializes event manager data structures. Must be called before any other event manager routine.

Prototype   `void evInitialize(void)`

Input   None

Output   None

Return   None

3.2.7.3 evGetFreeRecord: Get Next Free Event

Returns address of an empty event record and removes that record from the pool of free event records.

Prototype   `EVENT_RECORD *evGetFreeRecord(void)`

Input   None

Output   None

Return   address of next free event record

3.2.7.4 evGetNext: Get Next Event

Returns address of next event record to be processed.

Prototype   `EVENT_RECORD *evGetNext(void)`

Input   None

Output   None

Return   address of next event record to be processed

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

3.2.7.5 evReturnFreeRecord: Return Free Event Record

Returns an event record which is no longer required to the pool of free event records.

Prototype   `int evReturnFreeRecord(EVENT_RECORD *pRecord)`

Input   `pRecord`: Address of the event record to be freed.

Output   None

Return   Either 0 (Failure) or 1 (Success).

3.2.8 Global Variables and Definitions

The files "globals.c" and "globals.h" provides the global variables and definitions used by other modules. There is no function that can be called in this module.

3.2.9 WAN Card Initialization

The files "card.c" and "card.h" provides functions related to the setup of the WAN card.

3.2.9.1 cardDuplexInit: Setup Duplex Drivers

Sets up the Duplex driver. Also, initializes and activates all devices.

Prototype   `int cardDuplexInit(void)`

Input   None

Output   None

Return Codes `DPX_SUCCESS` (0) or an error value.

3.2.10 Main Module

The file "main.c" is the main module of the firmware.

3.2.10.1   main: Main module

This is the main module of the firmware. It calls the required initialization routines and starts main event loop of the command parser.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC *PMC-Sierra, Inc.*

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Prototype   `main (int argc, char **argv)`

Input   `argc`: This parameter is ignored.

                  `argv`: This parameter is ignored.

Output   None

Return Codes None

3.2.11 Serial Port Interface

The files "serial.c" and "serial.h" handles serial communication with the host system.

3.2.11.1   sioInit: Serial Port Initialization

Initializes the serial port and related variables.

Prototype   `int sioInit(int baud)`

Inputs   `baud`: baud rate (bits per second) at which to activate the serial port

Output   None

Return Codes Either 0 (Success) or 1 (Failure).

3.2.11.2   sioEchoControl: Terminal Echo Control

Enables/disables echoing of incoming characters (duplexing).

Prototype   `int sioEchoControl (int iFlag)`

Inputs   `iFlag`: 1 to turn echoing on; 0 to turn echoing off

Output   None

Return Codes Either 0 (Success) or 1 (Failure).

3.2.11.3   sioSendChars: Send Characters

Send a specified number of characters through the serial port.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Prototype   `int sioSendChars(char *pBuffer, int iSize)`

Inputs   `szBuffer`: pointer to characters to be send

　　　　`iSize`: number of characters to send

Output   None

Return Codes   Number of characters which were sent through the serial port.

3.2.11.4   sioSendString: Send String

Send a string through the serial port. This is done by calling `sioSendChars` with the correct parameters.

Prototype   `int sioSendString(char* szBuffer)`

Inputs   `szBuffer`: pointer to characters to be send

Output   None

Return Codes   Number of characters which were sent through the serial port.

3.2.12 Command Parser

The files "sioMain.c" and "sioMain.h" provides functions to parse commands and handle generic commands not specific to the WAN card.

3.2.12.1   sioMain: Main Event Loop

Infinite loop that waits for commands and calls the user command handler to handle them.

Prototype   `void sioMain (void)`

Input   None

Output   None

Return Codes None

3.2.12.2   UserCommand: User Command Handler

This function receives an event (command) and executes the appropriate function that corresponds to that command.

Prototype   `void UserCommand(EVENT_RECORD *pRec)`

Input   `pRec`: Event record containing the command

Output   None

Return Codes None

3.2.12.3   cmdComment: Comment Command Handler

An empty function used to ignore all comment lines.

Prototype   `void cmdComment(char **szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.12.4   cmdHelp: Help Command Handler

Handles the help command.

Prototype   `void cmdHelp(char **szArgs)`

Input   `szArgs`: Array of arguments of the command.

Output   None

Return Codes None

3.2.12.5   hlpComment: Comment Help Information

Displays a short description of how to write comments.

Prototype   `void hlpComment(void)`

Input   None

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

ISSUE 2

PM7350 S/UNI-DUPLEX

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

Output   None

Return Codes   None

3.2.12.6   hlpHelp: Help Command Help Information

Displays a short description of the help command.

Prototype   `void hlpHelp(void)`

Input   None

Output   None

Return Codes   None

3.2.13 Microprocessor Initialization

The assembly file "start.s" contains the list of interrupt vectors. It also initializes the microprocessor (Motorola 68340) and sets up chip selects. This code is given control when the 68340 resets.

3.2.14 ANSI Library Update

The file "strtoul.c" replaces the `strtoul` function provided by the standard library. Early versions of the IntrolC compiler came with a very old implementation of the ANSI library which provides less functionality so this file is used instead. The `strtoul` function provided is the same as the current ANSI implementation.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC-Sierra, Inc.

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

4   REFERENCES

1. PMC-Sierra Inc., PMC-1960835, "S/UNI-QJET Data Sheet", July 1999, Issue 6.

2. PMC-Sierra Inc., PMC-1980581, "S/UNI-DUPLEX Dual Serial Link PHY Multiplexer Data Sheet", April 2000, Issue 5.

3. PMC-Sierra Inc., PMC-1990799, "S/UNI-DUPLEX Device Driver Manual", July 1999, Issue 1.

4. PMC-Sierra Inc., PMC-1990832, "DSLAM Reference Design: System Design", September 2000, Issue 3.

5. PMC-Sierra Inc., PMC-1990474, "DSLAM Reference Design: WAN Card", September 2000, Issue 3.

RELEASED

APPLICATION NOTE

PMC-2000684

PMC *PMC-Sierra, Inc.*

PM7350 S/UNI-DUPLEX

ISSUE 2

DSLAM REFERENCE DESIGN: WAN CARD SOFTWARE MANUAL

CONTACTING PMC-SIERRA, INC.

PMC-Sierra, Inc.
105-8555 Baxter Place Burnaby, BC
Canada V5A 4V7

Tel: (604) 415-6000

Fax: (604) 415-6200

Document Information: document@pmc-sierra.com
Corporate Information: info@pmc-sierra.com
Application Information: apps@pmc-sierra.com
                             (604) 415-4533
Web Site: http://www.pmc-sierra.com

PMC-2000684 (R2)   Issue date: November 2000